
Università degli Studi di Firenze
Facoltà di Ingegneria - Dipartimento di Sistemi e Informatica

Tesi di Laurea Magistrale in Ingegneria Informatica

# Riconoscimento di eventi in video mediante l'utilizzo di string-kernel

*Candidato*
Filippo Amendola

*Relatori*
Prof. Alberto Del Bimbo

Ing. Marco Bertini

*Correlatori*
Ing. Lamberto Ballan

Ing. Giuseppe Serra

Anno Accademico 2008-2009

*Un tratto di strada finisce,*
*un piccolo grande incrocio attende.*
*Guardi l'orizzonte, cerchi la direzione,*
*rimani a riflettere sulla scelta migliore.*
*Poi smetti di pensare, fai per sederti,*
*e ti accorgi che i tuoi piedi han già mosso tre passi.*
***F.A.***

# Ringraziamenti

*Ringrazio il Professore A. Del Bimbo e M. Bertini, tutti i ricercatori e gli assistenti del MICC che mi hanno assistito e sopportato durante questo lungo lavoro di tesi.*

*Ringrazio i miei genitori per avermi dato la possibilità di intraprendere questo cammino e per aver avuto fiducia in me.*

*Ringrazio tutti i colleghi universitari per avermi sostenuto, incoraggiato e aiutato nel proseguimento degli studi e con cui ho trascorso questi anni della mia vita. Un ringraziamento particolare ad Ambra che ha condiviso da vicino tutto il tragitto percorso.*

*Ringrazio tutti i miei amici, vecchi e nuovi, belli e brutti, simpatici e antipatici, per l'aiuto morale che mi hanno sempre fornito. Un ringraziamento specialmente ad Alessio e Serena, presenti in particolar modo.*

*Ringrazio me stesso per aver saputo sfruttare questa occasione e per essermi impegnato almeno quanto basta.*

# Indice

# Elenco delle figure

# Sommario

L'informatica applicata al multimediale oramai ha invaso le nostre vite in maniera del tutto trasparente, inserendosi nei più svariati settori, come la musica, sempre più integrata con la presenza dei lettori MP3, la fotografia digitale, i film, le trasmissioni televisive, i video amatoriali e i servizi di distribuzione dei contenuti multimediali su internet. Considerando tutte queste fonti di materiale digitale, la quantità di informazioni che ognuno di noi sta accumulando comincia ad avere un peso rilevante, portando alla creazione di veri e propri archivi multimediali sempre più variegati ed ampi.

Questo fatto genera la necessità di introdurre e adottare delle tecniche di annotazione e indicizzazione di tali contenuti che facilitino il recupero e la ricerca di specifiche informazioni, soprattutto a livello semantico. Se in precedenza l'annotazione era eseguita manualmente, con il crescente impiego dei dispositivi di registrazione e memorizzazione digitale, tale procedimento non è più possibile, e la ricerca si è orientata alla realizzazione di sistemi in grado di individuare automaticamente all'interno di archivi multimediali i contenuti che soddisfano la richiesta dell'utente. Per quanto riguarda i file video digitali, l'obiettivo si focalizza sulla *Concept Detection*, cioè il rilevamento di concetti sia statici, oggetti e scene, che dinamici come eventi ed azioni. Se per i primi vi è già uno grosso lavoro svolto negli anni passati, i secondi, intrinsecamente più complessi, sono attualmente in fase di studio.

Il lavoro realizzato in questa tesi si focalizza sul riconoscimento di eventi e azioni all'interno di generiche sequenze video. L'idea che si trova alla base di questo studio è quella di estendere e analizzare il comportamento statico delle tecniche utilizzate per l'*object recognition*, sfruttando tecniche di *Information Retrieval* (IR) attualmente applicate in ambito testuale. Negli ultimi anni, per la descrizione delle caratteristiche locali delle immagini, trova largo impiego l'utilizzo di descrittori locali di punti di interesse, come i SIFT (Scale Invariant Features Trasform). Questo tipo di descrittore ha avuto un grande successo perché risulta particolarmente robusto rispetto alle principali alterazioni che le immagini posso subire, e, in aggiunta a queste proprietà, si caratterizza anche per la sua semplicità di individuazione e computazione, che ne giustifica la larga diffusione che ha ottenuto recentemente.

Il lavoro realizzato parte da queste considerazioni e ha l'obiettivo di utilizzare queste features anche per il riconoscimento di eventi e azioni all'interno di sequenze video. Basandosi sull'assunzione che, per la descrizione di tali concetti, l'aspetto temporale sia di fondamentale importanza, il punto cruciale sta nell'individuare quale sia il modo migliore di integrare tale aspetto nella procedura di riconoscimento. Dal momento che le sequenze video non sono altro che una successione di fotogrammi, le tecniche attuali inseriscono nel descrittore di ogni singola immagine, o nel descrittore dell'intera sequenza, informazioni di carattere temporale in modo da inglobare tale aspetto direttamente nella rappresentazione del video. L'idea che invece è stata sviluppata in questo lavoro, è quella di paragonare la sequenza di fotogrammi, rappresentata con le tecniche già consolidate nel *object recognition*, ad una sequenza di parole di un semplice testo, e sfruttare questo paragone per utilizzare, con degli accorgimenti particolari, le tecniche di analisi testuale nel contesto visuale. L'obiettivo principale è di verificare se un approccio di

questo tipo, in cui l'aspetto temporale viene considerato a posteriori nell'evolversi della sequenza stessa dei descrittori statici, può essere una strada promettente da prendere in considerazione per gli sviluppi futuri.

La tesi presenta quindi dei risultati sperimentali ricavati a partire da due tipi di database: il primo di video sportivi (calcio), di dimensioni ridotte su cui è stata eseguita la verifica dei parametri, ed il secondo di notiziari televisivi, cioè TRECVID 2005, un dataset che ormai è divenuto lo stato di fatto nella valutazione delle tecniche di *Concept Recognition*.

# Capitolo 1

# Introduzione

## 1.1 Il contesto e lo Stato dell'Arte

Il recupero e la classificazione di contenuti multimediali secondo la loro semantica, è attualmente una delle sfide più difficili nella Computer Vision, specialmente considerando l'enorme variabilità dei contenuti presenti all'interno di ogni singola categoria di concetti e le molteplici problematiche che si possono verificare nella rappresentazione di questi concetti, come le occlusioni, la confusione introdotta dallo sfondo, i cambiamenti di illuminazione, di posa e così via. Il campo dell'indicizzazione, cioè la classificazione delle risorse multimediali in base ai loro contenuti, ha assistito ad una rapida crescita negli ultimi anni, alimentato da una sempre crescente digitalizzazione, dall'aumento dei supporti di memorizzazione e dall'incremento della capacità di trasmissione. La nascita di siti come YouTube [1] e la sezione video di Google [2] hanno contribuito ad aumentare la quantità di materiale video fruibile liberamente dagli utenti, promuovendone la condivisione. Questi archivi digitali

[1] http://www.youtube.com
[2] http://video.google.com

sono sfruttati nei contesti più svariati: gruppi musicali ne fanno uso per auto-promuoversi, politici ed attivisti per esporre il proprio pensiero e molti utenti vi si appoggiano per realizzare veri e propri video-diari personali. In questo contesto l'uso di sistemi in grado di indicizzare automaticamente, ordinare e recuperare dati multimediali dal punto di vista semantico, analizzandone il contenuto, è divenuta una necessità stringente. Se in origine l'indicizzazione veniva demandata ad operatori che lo svolgevano in maniera del tutto manuale, con i conseguenti problemi di scalabilità e di errori di etichettatura, la recente esplosione della quantità di contenuti fruibili pubblicamente ha reso questa operazione del tutto improponibile. Spinti da questa incalzante domanda, sono emersi studi e ricerche per ottenere tecniche di analisi efficaci ed efficienti sia nella creazione dei contenuti, che nella loro ricerca.

Tra le varie categorie di contenuti multimediali, il campo che al momento sta impegnando gran parte dei ricercatori riguarda lo studio e l'analisi delle sequenze video. Gli studi sulla classificazione di immagini e sull'analisi audio,

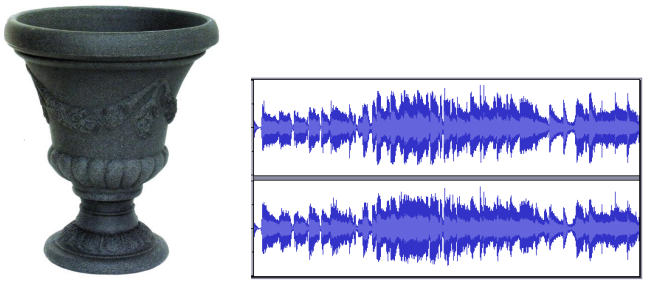

**Figura 1.1:** *Esempi di contenuti multimediali*

infatti, hanno già avuto il loro momento di estrema attenzione e, anche se i settori di ricerca non sono stati abbandonati, le performance ottenute sono già molto elevate.

Nel campo dei video digitali, le attuali tecniche di descrizione del contenu-

to multimediale hanno avuto un'evoluzione e un miglioramento consistente se analizzate in un contesto statico, cioè di riconoscimento delle scene e di oggetti. Se però estendiamo l'analisi a concetti più complessi, quali azioni ed eventi, che possiedono, oltre a caratteristiche puramente di apparenza, come per gli oggetti e le scene, anche informazioni più astratte, come l'aspetto temporale, le tecniche precedenti risultano ancora deboli e poco robuste.

Fornendo una breve panoramica sulle tecniche oggi conosciute, possiamo classificare gli approcci in due grandi rami:

- approcci olistici;
- approcci basati sulle parti.

I primi utilizzano l'immagine nella sua totalità e si basano sull'analisi complessa della stessa, cercando di ricavare informazioni sintetiche da caratteristiche come la posa, il contorno o il moto. I secondi utilizzano solo parte dell'immagine e una caratteristica su cui eseguire l'analisi desiderata, eliminando i dati superflui. La caratteristica scelta quindi ha un ruolo fondamentale e può essere considerata o come modello strutturale specifico o astratta in modelli detti *bag-of-features*. Anche la scelta di tale caratteristica può avere due approcci differenti: globali e locali. I primi catturano da ogni singola immagine delle informazioni di carattere generale, cioè informazioni che non dipendono dal dettaglio delle forme o di particolari elementi degli oggetti all'interno, ma cercano di individuare l'andamento generale dell'intera scena, come ad esempio l'analisi del colore. Semplicemente da questa definizione si nota che tale approccio risulta essere adatto a descrivere la scena o gli oggetti, ma è più difficile intravedere un legame con azioni o eventi all'interno dell'immagine. Al contrario, la rappresentazione locale cerca di individuare all'interno dell'immagine dei punti che possiedono delle proprietà

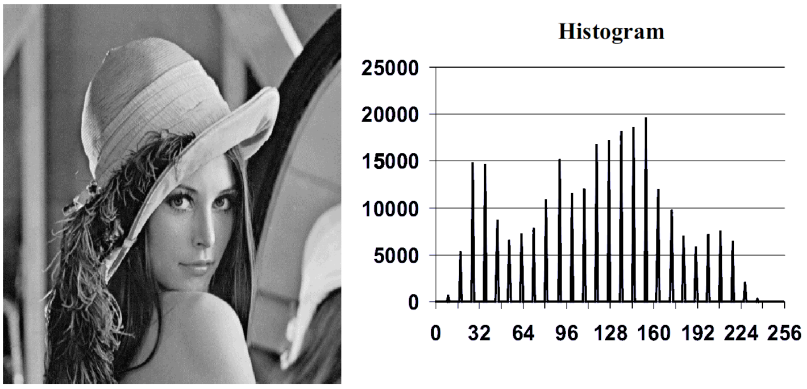


**Figura 1.2:** *Istogramma dei livelli di grigio*

di robustezza e invarianza a particolari trasformazioni, e quindi possono essere impiegati in maniera più efficace per l'analisi semantica del concetto da cercare.

Gli studi svolti in precedenza, elemento di partenza per le successive considerazioni, prendono in esame immagini statiche su cui effettuare analisi, riconoscimento e classificazione di scene e oggetti. Vi sono numerosi articoli e lavori che verificano svariate tecniche e modelli partendo, per esempio, dai risultati ottenuti dall'analisi del testo. Ngo et al. [1] utilizzando il modello *Bag-of-Words* (BoW) applicato a delle features locali, prendono in prestito alcune tecniche di analisi del testo, cioè la "pesature dei termini", la rimozione delle "stop word" e la "selezione delle features", per la classificazione di scene. Nel lavoro di Lazebnik et al. [2] vengono unite informazioni spaziali, ricavate dalla suddivisione dell'immagini in sotto regioni, tecnica ripresa da [3], con l'estrazione di descrittori locali di punti di interesse, cioè i SIFT (Scale Invariant Features Transform). Questa tecnica lavora partizionando in maniera incrementale l'immagine in sotto regioni di calcolo sempre più piccole ed estraendo gli istogrammi locali delle features. La "piramide spaziale" risultante è una semplice, e computazionalmente efficiente, estensione della

rappresentazione non ordinata delle *Bag-of-Words*, e mostra un significativo miglioramento delle prestazioni nell'obiettivo di *Object Recognition*.

La ricerca si è poi orientata verso la rilevazione e il riconoscimento di concetti più complessi, quai azioni e scene, evolvendo le tecniche precedenti. Niebles et al. [4], per riconoscere le azioni umane, usano una tecnica non supervisionata di apprendimento (pLSA), derivata dal dominio testuale, sfruttando le features estratte dal rilevatore proposto da Dollàr [5]. Zhou et al. [6], utilizzando i SIFT per la descrizione del contenuto dei fotogrammi, descrivono la sequenza video attraverso dei modelli statistici a mistura di gaussiane (Gaussian Mixture Models - GMM). Ngo et al. [7], estraendo le stesse caratteristiche di Niebles e inserendole nel modello BoW, aggiungono ulteriori informazioni riguardanti sia il moto relativo intra-fotogramma, attraverso il calcolo dei vettori di moto, sia informazioni sulla correlazione tra i contenuti di fotogrammi differenti, attraverso la creazione di una particolare ontologia ricavata dalle caratteristiche estratte. Xu et al. [8] basandosi su features completamente differenti, cioè istogrammi i cui elementi rappresentano la percentuale di affinità con un determinato concetto, chiamati *concept score* (CS), effettuano un'analisi sul partizionamento del video in sotto parti, e il successivo allineamento multi scala attraverso l'utilizzo della metrica EMD (Earth Mover's Distance), vedi Figura 1.3.

Nella letteratura scientifica più recente sono stati proposti una grande quantità di rilevatori di punti di interesse spazio-temporali, principalmente estendendo operatori nati per le immagini. I lavori di Laptev e Dollàr, [9] [5], hanno avuto il maggior successo e su di essi sono basati diversi lavori successivi. Scovanner et al. [10] modificano il descrittore SIFT di Lowe [11] estendendo il calcolo dell'orientazione di ogni punto da uno spazio bidimensionale ad uno tri-dimensionale, costruendolo in funzione delle imma-

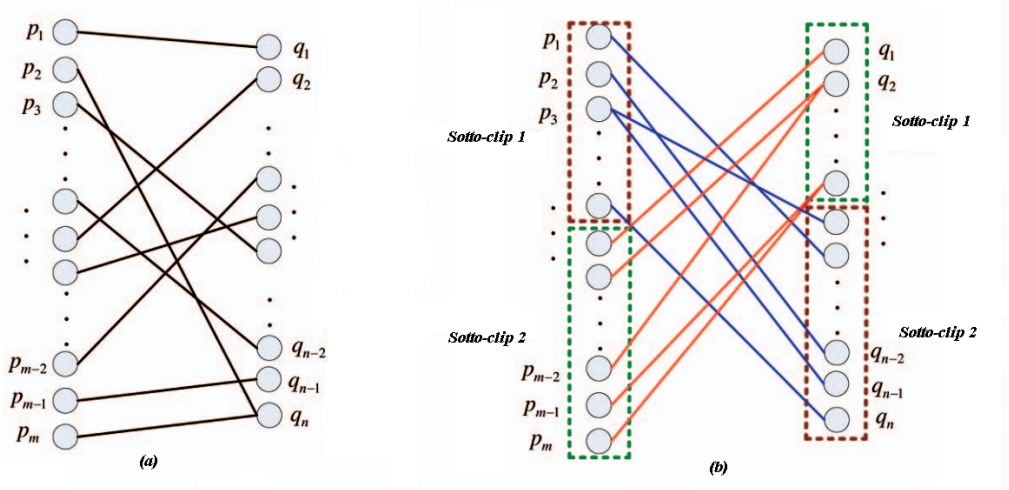


**Figura 1.3:** *Allineamento delle sequenze video effettuato da Xu et al. Con $p_i$ e $q_j$ si identificano i fotogrammi delle sequenze. In (a) l'allineamento viene eseguito tra i fotogrammi di tutta la sequenza. In (b), partizionando le sequenze in sotto-clip, viene effettuato l'allineamento solamente all'interno delle sotto-clip.*

gini precedenti e successive. Wong et al. [12] studiano come selezionare le parti dell'immagine che contengono le informazioni più rilevanti, tramite l'analisi di informazioni globali, per poi estrarre da queste aree le informazioni locali ottenute dai lavori di Laptev e Dollàr.

Uno dei maggiori problemi rilevati da molti di questi lavori e, più in generale, dalle applicazioni del descrittore SIFT a problemi di recupero in grandi collezioni di features, è sicuramente l'alta dimensionalità dei descrittori stessi. L'elevato numero di features localizzate in ciascuna immagine rende infatti molto dispendiose le operazioni di confronto su larga scala; per far fronte al problema sono stati proposti diversi metodi in cui si cerca di ridurre la dimensionalità dello spazio dei features vector, ricorrendo a tecniche come la PCA (Principal Component Analysis).

## 1.2 Obiettivi

Essendo l'*Event Recognition* in ambito multimediale ancora un campo relativamente giovane, dove i risultati non soddisfano le aspettative e le tecniche utilizzate, derivanti dall'ambito del riconoscimento di oggetti, non sembrano adattarsi a questa nuova categoria di ricerca, l'obiettivo principale di questa tesi è quello di adattare ed esaminare le tecniche già sperimentate in altri ambienti per fornire ulteriori basi di sviluppo.

Nonostante siano già stati proposti in letteratura alcuni studi riguardanti le tecniche di *Text Retrieval* applicate a questo contesto, crediamo che fornire un ulteriore approfondimento permetta di ricavare informazioni importati per il futuro. L'intrinseca complessità del problema porta la ricerca verso lo studio di nuove tecniche di astrazione dei concetti in esame, ampliando e favorendo la nascita di nuovi metodi sempre più complessi ed elaborati, ma trascurando leggermente l'analisi dei precedenti lavori.

Nel lavoro svolto, sono utilizzate le tecniche di analisi del testo, come l'approccio detto *bag-of-features*, per cercare di modellare l'aspetto temporale che riteniamo essere uno dei fattori principali per descrivere i concetti astratti che vogliamo recuperare, ovvero azioni e d eventi. Come verrà illustrata in dettaglio nel Capitolo 3, l'idea è stata quella di sfruttare l'analogia tra un video, cioè una sequenza di immagini, e un testo, cioè una sequenza di parole. Una volta individuato un dizionario di riferimento, il primo passo è stato associare un descrittore, detto "frase", alla sequenza, il quale è caratterizzato dal possedere tanti sotto elementi quanti sono i frame del video, vedi Figura 1.4. In questo modo, per ogni video, riusciamo ad estrarre sia informazioni sul contenuto statico, ovvero gli oggetti che rappresentano la scena, sia informazioni temporali, cioè andando ad analizzare l'andamento di questi descrittori lungo la sequenza.

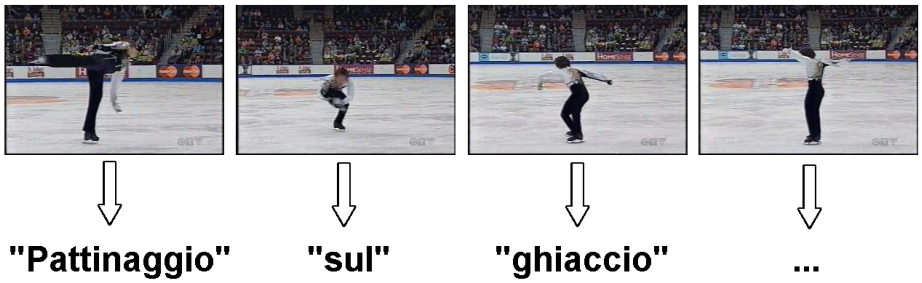


**Figura 1.4:** *Analogia tra una sequenza video e un testo.*

Completate quest'associazioni, il passo successivo per il riconoscimento delle sequenze ha coinvolto l'analisi delle "frasi" cercando di sfruttare l'analogia con il mondo testuale. A tale scopo, sono state sperimentate delle metriche di edit-distance e di analisi delle sotto stringhe per ricavare un parametro di distanza tra ogni coppia di video, così da poterne effettuare la classificazione.

## 1.3 Organizzazione della tesi

Viene fornita ora una breve descrizione dell'organizzazione di questo documento.

Il capitolo appena concluso fornisce il contesto generale in cui inserire questa tesi, illustrando quelle che sono le linee guida del momento e discutendo le varie problematiche, riportando le soluzioni precedentemente proposte in letteratura e introducendo l'approccio da noi proposto.

Nel **Capitolo 2** vengono approfondite e descritte in dettaglio le tecniche e le metodologie su cui si è focalizzata l'attenzione per questo lavoro di tesi, concludendo con l'estensione di tali processi all'ambiente da noi studiato.

Nel **Capitolo 3** verrà illustrata la soluzione proposta per il riconosci-

mento di eventi e azioni in video digitali. In particolare questo capitolo è strutturato in modo da affrontare e analizzare ogni singolo passo della procedura proposta, sottolineando le varie scelte da noi adottate.

Infine, nel **Capitolo 4**, verranno presentati in dettaglio gli esperimenti ed i risultati ottenuti, partendo proprio dalla descrizione del dataset utilizzato e fornendo una panoramica dei parametri che sono stati presi in considerazione negli esperimenti eseguiti.

Le ultime osservazioni sono illustrate nel **Capitolo 5**, fornendo suggerimenti per gli sviluppi futuri del nostro approccio e discutendo eventuali alternative possibili.

# Capitolo 2

# Rappresentazione di concetti in immagini e video

I primi obiettivi nell'ambito della *Concept Detection* sono stati quelli di individuare oggetti specifici all'interno di immagini. Con il passare degli anni e il progredire della ricerca, l'obiettivo si è spostato verso il riconoscimento di categorie di oggetti, cercando di individuare quelli che sono dei pattern comuni che possano generalizzare dei concetti di più alto livello. Gli approcci utilizzati possono essere suddivisi in approcci olistici o basati sulle parti, ma, come precedentemente accennato, la ricerca si è orientata verso l'utilizzo di approcci basati sulle parti, perché risolvono alcuni problemi, come l'occlusione, e catturano in maniera più semplice i concetti di interesse. Un ulteriore sviluppo, che ha segnato il proliferare di quest'ultimo approccio, riguarda le tipologie di caratteristiche che sono individuabili all'interno di un'immagine. Recentemente le features di tipo globale, come per esempio l'istogramma di colore, sono passate in secondo piano, preferendo utilizzare features locali capaci di sintetizzare meglio i concetti presenti all'interno di un'immagine. Le features globali, fornendo comunque informazioni molto

rilevanti, vengono utilizzate come integrazione alle features locali.

In questo capitolo daremo una definizione di features locali e descriveremo il funzionamento e le caratteristiche di questi operatori. Successivamente verranno analizzati in dettaglio i meccanismi che stanno dietro agli approcci basati sulle parti, orientandosi sui modelli non strutturati e descrivendo le tecniche già sperimentate nel riconoscimento di oggetti e scene. Infine, verranno illustrate le problematiche che devono essere affrontate nel caso esteso del riconoscimento di azioni/eventi.

## 2.1 Punti di interesse e descrittori locali

La classificazione di immagini o video in categorie semanticamente distinte è un problema di grande interesse nell'ambito della ricerca. Inizialmente, ponendosi come obiettivo quello di riconoscere oggetti specifici, come una mela rossa o una FIAT 500, le tecniche sviluppate focalizzavano l'attenzione sull'individuazione di features altamente specifiche. In questo modo, ogni tecnica sperimentata, risultava altamente discriminante nel contesto scelto, ma molto inefficiente se applicata a contesti differenti. Generalizzando questi oggetti in classi, per esempio frutta e macchine, le informazioni soggettive degli oggetti vengono perse, concentrandosi sullo studio di quegli aspetti che sintetizzano l'intere categorie. Questo obiettivo viene raggiunto effettuando uno studio delle caratteristiche all'interno delle immagini, che possano discriminare contenuti distinti; inizialmente questo tipo di classificazione si basava sulla descrizione del colore, della tessitura o di altre proprietà visuali delle immagini, cioè approcci di tipo globali. Features come l'istogramma di colore, comprese tutte le sue derivazioni, e i filtri di Gabor appartengono a questa categoria. Si possono aggiungere a queste features tecniche partico-

**Figura 2.1:** *Esempio di oggetti specifici.*

lari per aumentare le prestazioni, come l'inclusione di informazioni spaziali tramite la suddivisione dell'immagine in regioni rettangolari, o segmentata in base agli oggetti in primo piano e quelli in secondo piano. Le caratteristiche calcolate in queste regioni sono concatenate a formare un singolo descrittore vettoriale dell'immagine.

Recentemente l'attenzione si è spostata su features diverse per la descrizione delle immagini, focalizzandosi su punti particolari, chiamati *keypoints* o *local interest points*, con un approccio di tipo locale. Questo tipo di rappresentazione cattura proprio l'aspetto disordinato e sparso dei concetti contenuti, senza imporre vincoli specifici di ordinamento o concatenazione. Un insieme di keypoints può essere visto come una patch saliente dell'immagine, caratterizzata da un alto contenuto informativo locale della stessa.

Ci sono due passi fondamentali che bisogna distinguere:

- la rilevazione;
- la descrizione.

Per quanto concerne la rilevazione o individuazione di questi punti chiave, ci sono vari approcci:

- estrazione casuale;

- griglia regolare (campionamento denso): l'immagine viene segmentata da linee orizzontali e verticali che individuano i punti da estrarre. A dispetto della semplicità, questo metodo fornisce dei buoni risultati per la rilevazione di textures, scenari naturali e tutte quelle situazioni in cui l'oggetto o la persona da rilevare è messa in evidenza da uno sfondo uniforme. Questo è dovuto essenzialmente al fatto che la griglia è capace di catturare informazioni globali, cioè distribuite uniformemente nell'immagine. Lo svantaggio è che tralascia la maggior parte dell'informazione saliente degli concetti in esame;

- punti di interesse (campionamento sparso): caratteristiche particolari sono rilevate con appositi detectors che sono in grado di selezionare zone precise (come egdes, corners, blobs) dell'immagine. Vi sono svariate tecniche di estrazione:

  – Harris corner detector;

  – Difference of Gaussian (DoG) [13]; viene impiegato all'interno del SIFT (Lowe [11]);

  – Affine covariant patches.

  Il vantaggio di usare il campionamento sparso risiede nel fatto che cattura le informazioni più rilevanti dell'immagine, ma di contro l'individuazione di tali punti è più complessa e dipendente dal detector utilizzato.

Selezionati i punti salienti, il passo successivo consiste nel descriverli in maniera robusta. Diversi tipi di informazioni possono essere prese in considerazione, quali i bordi o il gradiente dell'immagine. Una tecnica comune è quella di suddividere l'intorno del punto in celle, ed estrarre da ogni cella

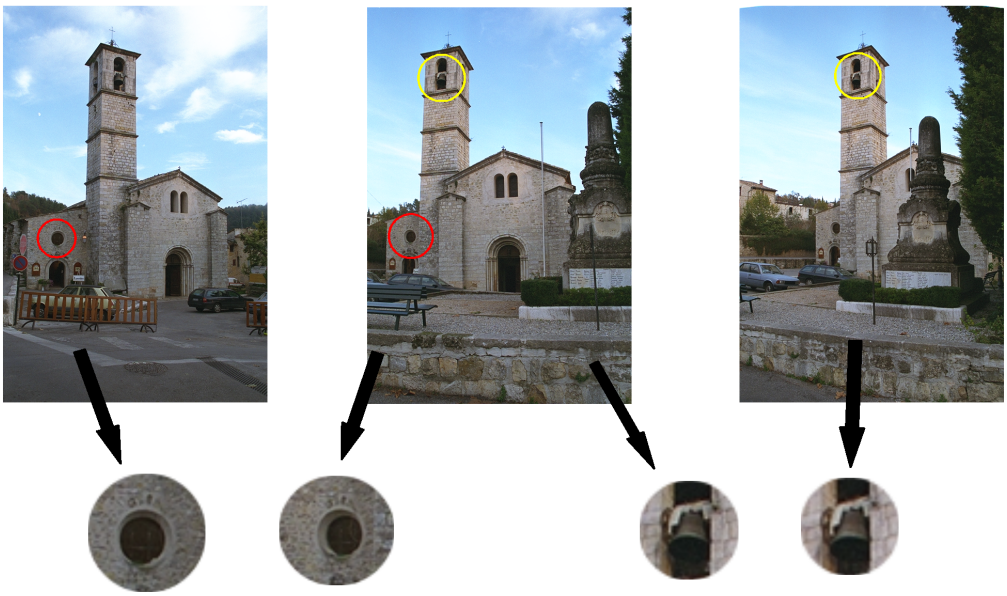

**Figura 2.2:** *Il cambio di punto di vista altera globalmente la facciata della chiesa. Prendendo in esame solo piccole porzioni di scena, come il rosone o la finestra, le alterazioni sono molto meno evidenti.*

un istogramma quantizzato delle orientazioni dei gradienti dei singoli pixel, che in pratica è la versione grossolana del descrittore SIFT analizzato successivamente.

Il vantaggio che si ottiene da una rappresentazione di questo tipo, è innanzitutto la possibilità di localizzare un oggetto in un'immagine anche se occluso, a patto ovviamente che la regione visibile contenga un numero sufficiente di punti di interesse. Sempre allo scopo del riconoscimento di oggetti, l'uso di una descrizione locale consente una maggiore robustezza a trasformazioni prospettiche e distorsioni. Infatti, se globalmente l'immagine subisce una trasformazione prospettica (ad esempio per il cambio di punto di vista, vedi Figura 2.2), localmente le regioni sono meno deformate e questa deformazione può essere modellata come una trasformazione affine. Infine l'uso di

una rappresentazione sparsa per scene ed oggetti si è dimostrata vantaggiosa anche nell'ambito della classificazione. Ovvero la difficoltà di creare modelli (appresi statisticamente) per oggetti rigidi (sedie, moto, aerei ... etc.), le cui categorie sono altamente variabili al loro interno, può essere superata rappresentando ciascun oggetto come collezione di regioni locali.

### 2.1.1 SIFT: Scale Invariant Features Transform

Il SIFT è un operatore, proposto da Lowe [11], che consente l'estrazione di features locali da un'immagine, in modo da garantire buone performance di matching da differenti viste di uno stesso oggetto o di una stessa scena. Le features estratte sono locali, invarianti a cambiamenti di scala (dovute ad esempio ad un'operazione di zoom) ed a rotazioni. Allo stesso tempo sono particolarmente robuste a cambi di illuminazione, rumore, trasformazioni geometriche affini ed a variazioni del punto di vista nello spazio 3D.

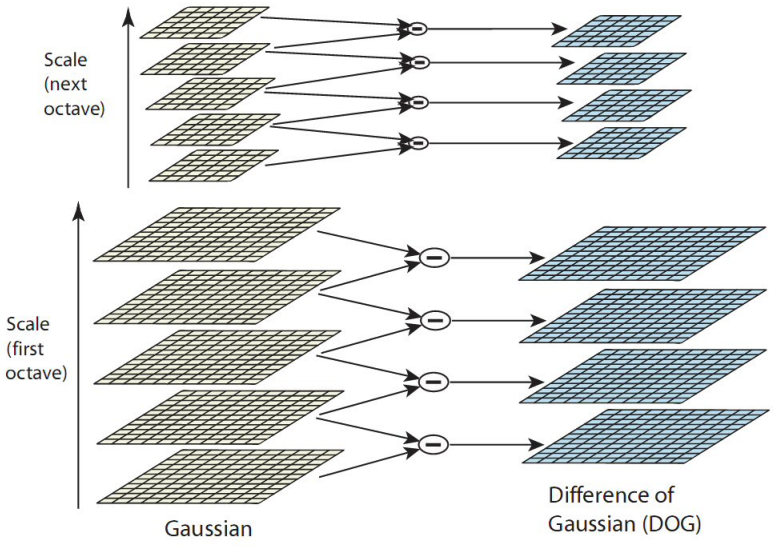


**Figura 2.3:** *Calcolo efficiente dell'operatore Difference of Gaussians(DoG) sfruttando le piramidi di immagini.*

L'individuazione di questi punti può essere sinteticamente riassunto nella seguente successione di passi:

1. individuazione degli estremi locali nello scale-space: si effettua filtrando ripetutamente l'immagine originale con kernel gaussiani di diversa varianza, ottenendo due piramidi di immagini (Figura 2.3): la prima costituita dalle immagini ripetutamente convolute con filtri Gaussiani $G(x, y, s)$, la seconda dalle diverse DoG, $D(x, y, s)$. Gli estremi locali saranno quindi ricercati ad ogni livello della piramide di DoG;

2. localizzazione dei keypoints, cioè dei massimi e dei minimi locali della $D(x, y, s)$: viene effettuata comparando ciascun campione con gli otto adiacenti del livello corrente e con i nove delle due scale immediatamente superiore ed inferiore. Viene poi eseguite una scrematura dei punti scelti;

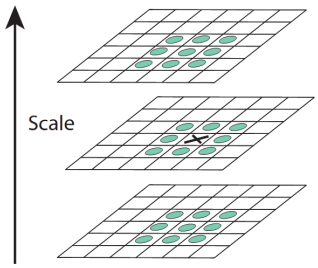


**Figura 2.4:** *Gli estremi locali della DoG sono individuati confrontando il pixel (contrassegnato dalla X) con i 26 adiacenti in una regione 3x3 alla scala corrente, ed alle due scale adiacenti.*

3. assegnazione di una (o più) orientazioni canoniche: una volta individuate le coordinate e la scala del keypoint, si assegna una orientazione che garantisce una buona robustezza a rotazioni;

4. generazione dei descrittori locali, sfruttando i dati pre-calcolati nella fase precedente.

I primi due passi riguardano il rilevamento dei punti, mentre gli ultimi due permettono di ottenere un descrittore efficace e robusto.

Tutto il procedimento meriterebbe un'analisi approfondita per la comprensione dei dettagli implementativi, ma rimandiamo direttamente alla lettura dell'articolo in questione [11] e di alcuni lavori che ne fanno uso, [14] e [15].

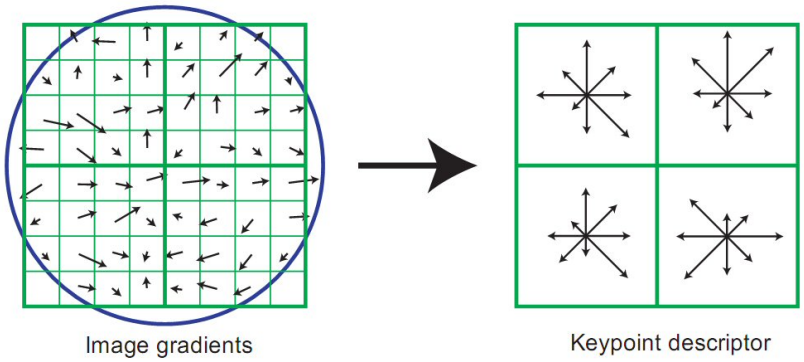


**Figura 2.5:** *Generazione dei descrittori locali. La figura mostra la creazione di un descrittore di dimensioni 2x2, costruito a partire da un set di 8x8 campioni dell'immagine L.*

Riassumendo brevemente è possibile dire che il maggiore contributo dell'operatore SIFT è sicuramente il suo descrittore. Il descrittore SIFT è considerato di fatto il miglior descrittore in quanto robusto e resistente a deformazioni locali poiché:

- sfrutta il gradiente come misurazione della regione locale rilevata dall'operatore DoG, vedi Figura 2.3: questo dà robustezza rispetto ai cambi di illuminazione;

- la regione considerata viene suddivisa in 16 sotto regioni e per ognuna di esse viene calcolato un istogramma delle orientazioni del gradiente (vedi Figura 2.5);

- viene assegnata a ciascun punto un'orientazione principale per ottenere invarianza alla rotazione. L'invarianza alla rotazione è ottenuta calcolando le suddette orientazioni relativamente all'orientazione principale.

## 2.2 Modello Bag-of-Words (BoW)

Le features locali permettono di estrarre delle caratteristiche specifiche di un'immagine e il primo modo di utilizzarle è in sostituzione alle features globali, cioè utilizzare queste caratteristiche direttamente per descrivere il contenuto delle immagini. Il passo immediatamente successivo è quello di svincolarsi dalla singole features locali che sono una parte discriminante dell'immagine, e di andare a generalizzarle raggruppandole secondo qualche criterio.

Un modello che esprime questo concetto nasce nell'*Information Retrieval* in ambito testuale. Un documento, cioè un insieme di parole appartenenti ad un dizionario, viene rappresentato come un vettore le cui componenti descrivono la frequenza di ogni parola del dizionario all'interno del testo stesso. Questo concetto può essere raffinato in svariati modi, a partire da una pre-elaborazione delle parole, per esempio lo *stemming*, cioè il processo di riduzione della forma flessa di una parola alla sua forma radice (e.g. Danzatrice e Danzatore verranno entrambi ridotti a Danz), oppure tramite l'eliminazione delle *stop words*, cioè parole, come gli articoli, che non hanno rilevanza hai fini dei concetti espressi nel testo. Un'altra tecnica che viene

spesso utilizzata in questo ambito è la *feature selection*, che ha lo scopo di individuare quali dati hanno un elevato contenuto informativo, associando ad ogni termine un punteggio in base ad una statistica relativa a tale termine e alle categorie del problema di classificazione. Alcuni esempi sono la statistica Chi-Quadro, $\chi^2(t, c)$, o la mutua informazione:

$$MI(t, c) = \sum_{t \in (0,1]} \sum_{c \in (0,1]} P(t, c) \log \frac{P(t, c)}{P(t)P(c)}; \tag{2.1}$$

per entrambi questi test statistici un valore elevato significa una dipendenza rilevante tra le due variabili (termine e categoria). Vi sono numerosi studi che illustrano le tecniche o i parametri migliori da utilizzare in determinati contesti.

Questo tipo di rappresentazione non tiene conto dell'ordine delle parole, né della punteggiatura, né di eventuali riferimenti tra un documento ed altri (e.g. collegamenti tra documenti html presenti nel web); un modello di questo tipo è chiamato **Bag-of-Words (BoW)**, ovvero un insieme non ordinato di termini. Un testo quindi è rappresentato da un vettore o istogramma, di dimensione uguale al dizionario, i cui elementi indicano la frequenza dei termini nel documento in esame.

L'idea è di applicare questo modello anche nell'ambito visuale, cioè trattare un'immagine allo stesso modo di un documento, ed individuare delle *visual words* appartenenti ad un dizionario visuale, da utilizzare per creare gli istogrammi di frequenza. L'obiettivo è quello di trasformare quindi il modello *Bag-of-Words* testuale nel modello **Bag-of-Visual-Words (BoVW)** in ambito visuale.

Riprendendo le features locali descritte nel capitolo precedente, per descrivere il contenuto di un'immagine, il primo passo consiste nell'individuare

un *dizionario visuale*. Il problema fondamentale sta nell'ottenere un vocabolario finito a partire da features locali appartenenti a spazi di dimensioni piuttosto elevate.

Il passo chiave consiste nel cercare di raggruppare insieme keypoints simili attraverso un algoritmo di clustering, ottenendo così una rappresentazione compatta dello spazio delle features. Il più utilizzato, proprio per la sua facilità di implementazione, è l'algoritmo *K-Means* che andremo ad descrivere dettagliatamente nella sezione successiva. I dizionari possono essere generati

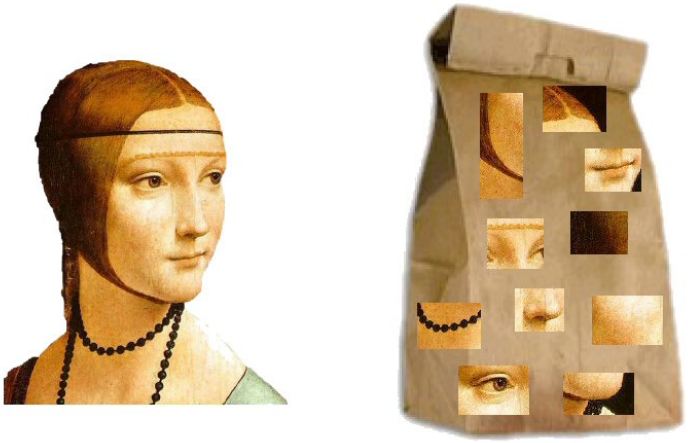

**Figura 2.6:** *Bag-of-Visual-Words.*

discretizzando i dati tramite una annotazione semantica delle feature [16], o guidando il processo di quantizzazione in maniera supervisionata [17], [18]. Il problema di ottenere dizionari visuali efficaci è stato affrontato in passato da Jurie e Triggs [19].

Sivic e Zisserman [20] estendono per la prima volta il modello Bag-of-Words al campo multimediale, cercando di fondere le tecniche di IR nate e sviluppatesi per il testo, con i metodi di descrizione locale di immagini. A questo scopo definiscono per la prima volta il concetto di dizionario visuale.

**Figura 2.7:** *Elementi base per la creazione del dizionario.*

Trattando ogni elemento del dizionario come *visual words*, possiamo pensare di avere un vocabolario capace di descrivere tutti i possibili patterns. Un'immagine in questo modo può essere rappresentata come *Bag-of-Visual-Words*, Figura 2.8, ovvero come un vettore che indica quali parole del vocabolario descrivono i contenuti in esame.

Nasce il problema di come assegnare i punti estratti dall'immagine al dizionario di riferimento per generare la parola. Se nell'ambito testuale questo problema non si pone, o semplicemente è un problema di importanza minore, nel nostro ambito invece la questione è più delicata, in quanto le operazioni coinvolte agiscono su elementi particolari, di cui non se ne conosce il significato semantico. Quindi, in funzione della dimensione del dizionario, l'assegnazione di un punto semplicemente al cluster più vicino in termini di distanza euclidea, può risultare troppo generico. Possono essere impiegate tecniche di assegnamento più complesse, che tengono conto di altri fattori o semplicemente che distribuiscono il punto in esame su più cluster proporzionalmente alla sua distanza da essi.

Associando questa *word* all'immagine viene generata un'impronta sintetica del contenuto statico, che può quindi essere utilizzata per la sua successiva analisi e classificazione. Questa procedura ha come passo chiave la scelta di features altamente informative che caratterizzino il contenuto dell'immagine

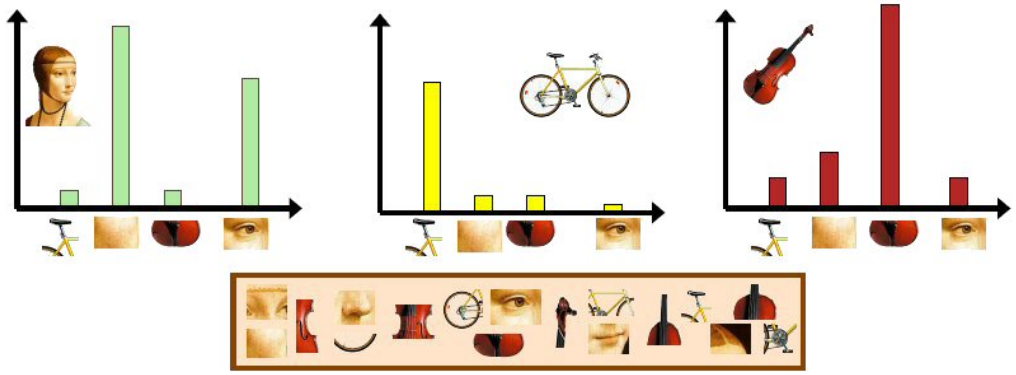

**Figura 2.8:** *Istogramma delle frequenza.*

e che possano risultare efficienti per la classificazione. Ricordando quanto detto nei paragrafi precedenti, i *keypoints* sono dei punti ad alto contenuto informativo locale di un'immagine, e possono essere rappresentati da descrittori semplici o complessi quali ad esempio i SIFT (sezione 2.1.1). Tecniche di descrizione locale, ed in particolare SIFT, sono state impiegate recentemente in letteratura in maniera massiccia per il riconoscimento di oggetti; sono stati presentati inoltre studi approfonditi di analisi e di comparazione delle performance delle diverse tecniche di descrizione locale, valutandone le potenzialità proprio in merito al problema della Object Recognition. Ngo et al. [1] riprendono proprio le tecniche di *Information Retrieval* (IR) e le applicano all'ambito visuale, analizzando la dimensione del dizionario, vari schemi di pesi e la loro normalizzazione (*Term Frequency - tf, Inverse Document Frequency - idf, tf-idf*), l'eliminazione delle stop-word e la selezione delle features per limitare la grandezza del vocabolario (*document frequency (DF), Chi-Quadro, mutual information (MI)*). Il loro obiettivo è di fornire la base per gli sviluppi futuri a partire dai lavori precedenti sviluppati nell'ambito testuale.

Bisogna anche tenere presente che la combinazione di varie tecniche può

dare risultati migliori, per esempio la combinazione del SIFT, che lavorano su immagini a livelli di grigio, con informazioni riguardanti il colore, come per esempio istogrammi di colori o momenti. Nel lavoro di Lazebnik et al. [2] vengono unite informazioni spaziali, ricavate dalla suddivisione dell'immagini in sotto regioni, tecnica ripresa da [3], con l'estrazione di features locali, cioè i SIFT. Questa tecnica lavora partizionando in maniera incre-

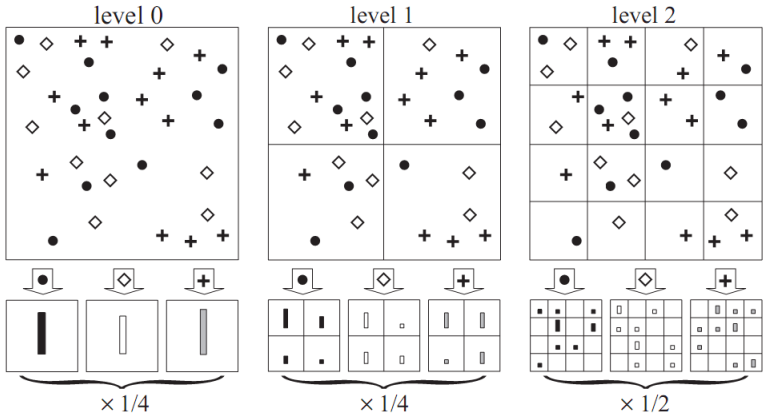


**Figura 2.9:** *Toys example di costruzione della piramide spaziale a 3 livelli.*

mentale l'immagine in sotto regioni di calcolo sempre più piccole e estraendo gli istogrammi locali delle features. La "piramide spaziale" risultante, vedi Figura 2.9, è una semplice, e computazionalmente efficiente, estensione della rappresentazione non ordinata delle *Bag-of-Words*, e mostra un significativo miglioramento delle prestazioni nell'obiettivo di *Object Recognition*.

La creazione di un dizionario visuale permette di ridurre la complessità dello spazio dei descrittori visuali, riducendolo ad un numero finito di prototipi: questa tecnica ci consente di trattare il dato visuale con le tecniche provenienti dall'IR. Un altro dei vantaggi dell'uso di parole visuali è l'implicita robustezza di un sistema di questo tipo: se il dizionario è creato in maniera efficace, i descrittori dei punti di interesse vengono aggregati in modo da rap-

presentare la stessa parte di un oggetto o di una scena dando vita appunto a delle parole visuali. Il principio alla base del meccanismo del modello bag of words è l'esistenza di un dizionario condiviso in cui le parole compaiono in più documenti sia della stessa categoria che di categorie differenti. Questo tipo di modellazione è quindi in grado di catturare la semantica (seppure in maniera rozza) presente nei documenti di testo. Questa tecnica realizza implicitamente un meccanismo di corrispondenza robusta tra punti chiave: oggetti dello stesso tipo (ad es. delle moto) possono avere un aspetto molto variabile; tuttavia alcune delle loro parti (ad es. le ruote, i fari) avranno una forte somiglianza. Destrutturando così la rappresentazione dell'immagine e rappresentando l'aspetto di ciascuna regione locale con un prototipo è possibile creare modelli statistici appresi per oggetti, scene o, come nel nostro caso, azioni o comportamenti umani.

### 2.2.1 K-Means

Per ottenere il *dizionario visuale* da utilizzare come elemento di riferimento, una possibilità è di partizionare lo spazio delle features attraverso un algoritmo di clustering. Uno dei più diffusi e semplici algoritmi di clustering è K-Means, il quale permette di suddividere gruppi di oggetti in $K$ partizioni sulla base dei loro attributi. È una variante dell'Algoritmo di aspettazione-massimizzazione il cui obiettivo è determinare i $K$ gruppi di dati generati da distribuzioni gaussiane. Si assume che gli attributi degli oggetti possano essere rappresentati come vettori, e che quindi formino uno spazio vettoriale. L'obiettivo che l'algoritmo si prepone è di minimizzare la varianza totale intra-cluster. Ogni cluster viene identificato mediante un centroide o punto medio. L'algoritmo segue una procedura iterativa. Inizialmente crea $K$ partizioni e assegna ad ogni partizione i punti d'ingresso o casualmente o usando

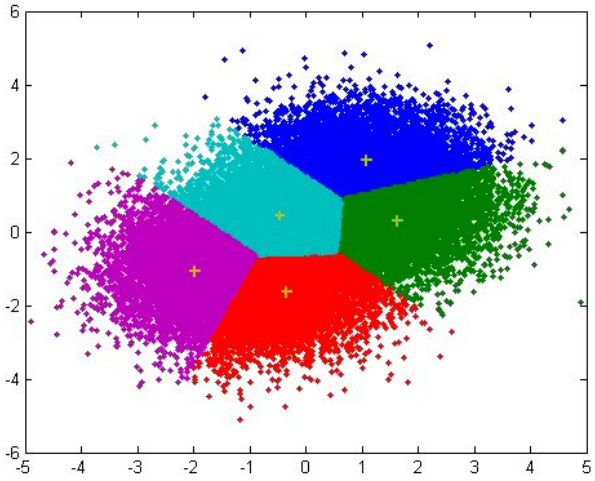


**Figura 2.10:** *Suddivisione di uno spazio 2D utilizzando l'algoritmo di clustering K-Means*

alcune informazioni euristiche. Quindi calcola il centroide di ogni gruppo. Costruisce quindi una nuova partizione associando ogni punto d'ingresso al cluster il cui centroide è più vicino ad esso. Quindi vengono ricalcolati i centroidi per i nuovi cluster e così via, finché l'algoritmo non converge.

Ne diamo di seguito una definizione formale. Dati m vettori in $\mathbb{R}^N$, definiamo $X = X_1, X_2, ..., X_m$ come insieme degli oggetti. Ricordiamo che si definisce partizione degli oggetti il gruppo di insiemi $P = P_1, P_2, ..., P_K$ che soddisfano le seguenti proprietà:

- $\bigcup_1^K P_i = X$: tutti gli oggetti devono appartenere ad almeno un cluster;
- $\bigcap_1^K P_i = 0$: ogni oggetto può appartenere ad un solo cluster;
- $0 \subset P_i \subset X$: almeno un oggetto deve appartenere ad un cluster e nessun cluster può contenere tutti gli oggetti.

Ovviamente deve valere anche che $1 < K < N$; non avrebbe infatti senso né cercare un solo cluster né avere un numero di cluster pari al numero di oggetti. Una partizione viene rappresentata mediante una matrice $U \in \mathbb{N}^{K \times N}$, il cui generico elemento $u_{ij} = 0, 1$ indica l'appartenenza dell'oggetto $x_j$ al cluster $x_i$. Indichiamo quindi con $C = C1, C2, ..., CK$ l'insieme dei K centroidi. A questo punto definiamo la funzione obiettivo come:

$$V(U, C) = \sum_{i=1}^{K} \sum_{X_j \in P_i} \|X_j - C_i\|^2 \tag{2.2}$$

e di questa calcoliamo il minimo seguendo la procedura iterativa:

1. Genera $U_v$, e $C_v$, casuali;
2. Calcola $U_n$, che minimizza $V(U, C_v)$;
3. Calcola $C_n$, che minimizza $V(U_v, C)$;
4. Se l'algoritmo converge ci si ferma, altrimenti $U_v = U_n$, $C_v = C_n$, e torna al passo 2.

Tipici criteri di convergenza sono o il cambiamento nullo nella matrice $U$, o la differenza fra i valori della funzione obiettivo in due iterazioni successive non deve superare una soglia prefissata.

La più popolare implementazione di questo algoritmo è il cosiddetto metodo di Lloyd (1956) nel quale sono implementate le seguenti euristiche per i passi 2 e 3. Al passo 2 viene associato ciascun punto al centro a lui più vicino; al passo 3 viene ricalcolato ogni centro come la media dei punti assegnati a quel cluster. La popolarità di questo algoritmo deriva dalla sua velocità di convergenza e semplicità di implementazione. Un dizionario visuale si può creare anche tramite altri strumenti, ognuno dei quali cerca di ovviare ai problemi dell'algoritmo suddetto. Ad esempio Mikolajczyk et al. [21] utilizzano

un algoritmo agglomerativo per ovviare all'inizializzazione casuale dell'algoritmo K-Means. L'algoritmo K-Means, non garantendo la convergenza ad un minimo globale della funzione obiettivo a causa dell'inizializzazione casuale, genera ad ogni ripetizione un dizionario differente.

Un altro svantaggio dell'algoritmo è che esso richiede di scegliere il numero $K$ di cluster da trovare. Inoltre, se i dati non sono naturalmente partizionati si ottengono risultati strani e in particolare, l'algoritmo funziona bene solo quando sono individuabili cluster sferici nei dati.

In ultima istanza l'algoritmo K-Means non è robusto nei confronti degli outliers; alcuni centri, anche di cluster ben definiti, possono essere attratti verso regioni "vuote" di spazio a causa anche di pochi punti molto distanti dal "vero" centro, ma non assegnabili ad altre partizioni. In un dominio conosciuto, e a maggior ragione nel caso si lavori in un dominio ancora sconosciuto, è utile effettuare un'analisi dei campioni per cercare di eliminare gli outliers prima di eseguire l'algoritmo di clustering, può generare risultati nettamente migliori.

## 2.3 Estensione al riconoscimento di azioni/eventi

L'approccio *Bag-of-Visual-Words* può essere applicato anche nel caso di riconoscimento di sequenze video, andando quindi ad elaborare dei contenuti multimediali che contengono informazioni di carattere temporale. In questo caso subentrano delle nuove informazioni che devono essere opportunamente rappresentate: affinché avvenga un corretto riconoscimento dei filmati devono essere perciò utilizzate delle features efficienti per la descrizione di scene dinamiche.

Una prima analisi possibile è quella di riprendere esattamente le tecniche applicate al contesto statico e di applicarle in versione estesa, cioè creando un nuovo descrittore concatenando i singoli descrittori estratti da ogni frame della sequenza, senza cioè aggiungere nessuna informazione temporale se non quella derivante dall'unione di più descrittori. L'informazione estratta del video però è di carattere puramente visuale, senza prendere in esame aspetti di correlazione tra frame o di evoluzione temporale delle features. In questa maniera viene catturato solo informazioni sul "cosa" è coinvolto in un evento e non come si evolve temporalmente.

In una sequenza video, un evento può essere descritto da due aspetti:

- *cosa* è rappresentato, ad esempio persone, oggetti, edifici, ecc;
- *come* l'evento evolve nel dominio temporale, vale a dire lo svolgimento dell'azione.

Il primo è costituito da informazioni statiche e risponde alla domanda che cosa. Le risposte a queste domande possono essere ottenute fondamentalmente da immagini statiche. Le caratteristiche per descrivere questo aspetto sono state studiate intensamente, includendo quelle globali (momenti di colore, wavelet texture, istogramma di colore e di contorno), locali (Bag-of-Visual-Words), e semantiche (concept score). Il secondo aspetto, invece, contiene le informazioni dinamiche dell'evento e risponde essenzialmente alla domanda come, ad esempio, il movimento di oggetti e l'interazione fra persone diverse. Queste informazioni possono essere raccolte solo da tutta la sequenza. Le informazioni di movimento, come prima intuizione, possono essere un importante elemento per descrivere l'evoluzione dell'evento.

Recentemente, diverse features del movimento sono state sviluppate per acquisire informazioni temporali, ad esempio l'istogramma di movimento, e

la mappa di moto. Tuttavia, le attuali tecniche sono ancora deboli in caso *Event-based Concept* perché:

- la maggior parte dei descrittori considerano solo uno dei due aspetti, vale a dire sfruttare separatamente *cosa* o *come* non può descrivere un evento;
- sono utilizzate solo le informazione sulla distribuzione del moto, il quale è stato dimostrato essere affetto da rumore in caso di video non vincolati;
- il moto osservato nel video è distorto dai movimenti della camera e non può rappresentare correttamente la reale attività e interazione degli oggetti coinvolti nell'evento.

Una possibilità per catturare le informazioni caratteristiche delle azioni o eventi, sta nel fare una pre-analisi delle sequenze video, andando ad individuare quei frame che possono essere identificati come rappresentanti dell'evento scelto, cioè utilizzare un approccio *key-frame based*. L'altra strada consiste nell'analizzare più frames della sequenza contemporaneamente e cercare di inglobare all'interno del descrittore le informazioni temporali caraatterizzanti tali concetti. Diamo nelle sezioni successive una descrizione di questi approcci.

### 2.3.1 Approccio key-frame based

In molti scenari, le informazioni rilevanti sono contenute in pochi frames della sequenza. Questi frames risultano significativi a causa di alcuni cambiamenti nei dati come la direzione, la velocità o la variazione del comportamento dall'andamento standard. Volendo fare un esempio, si consideri la traiettoria

tracciata da una mano quando si apre la porta: la forma della traiettoria dipende dalla persona che apre la porta, la posizione iniziale della mano, la direzione di visualizzazione della fotocamera, ecc. Modellare queste variazioni non è né facile né rilevante per l'attività di apertura della porta: di fatto l'azione di apertura si verifica nel giro di pochi fotogrammi quando la mano è a contatto con la porta. La sequenza dei fotogrammi chiave (estendere la mano, afferrare la maniglia e aprire la porta) è sufficiente per la rappresentazione dell'evento. Allo stesso modo, possiamo dire che camminare è una sequenza di eventi o posizioni chiave che comprende la posizione di riposo, cioè quando i piedi sono più vicini gli uni agli altri, e la posizione di oscillazione, cioè quando i piedi sono alla massima distanza. Il jogging può essere rappresentato con una serie simile di frames, ma le modifiche da frame a frame sono diverse da quelle di camminare.

La prima intuizione quindi, consiste nell'individuare all'interno della sequenza un frame rappresentativo che identifichi l'azione in questione. La scelta può essere fatta seguendo vari approcci, a partire dalla scelta casuale del frame all'interno della sequenza. Può essere sfruttato il dominio di conoscenza in maniera top-down, ma richiede un inteso lavoro di modellazione a priori che risulta essere lungo e tedioso. Se invece si concentra l'attenzione sulla variazione dei dati per identificare il frame principale, si va incontro a problemi di eliminazione del rumore. Infatti ipotizzando che le caratteristiche importanti dell'evento siano presenti in maniera persistente nella maggioranza dei frames della sequenza, può risultare difficile distinguere i micro cambiamenti dal rumore stesso della sequenza. Cuntoor in [22] presentano proprio un approccio di questo tipo, basando la rilevazione dei keyframes proprio sui cambiamenti dei dati, definendo un operatore di trasformazione che, ad ogni istante, lega lo stato passato con lo stato futuro.

Individuato il keyframe rappresentativo della sequenza, le tecniche applicate sono le stesse dell'*Object Recognition*, quindi: estrazione, analisi delle features e successiva classificazione.

### 2.3.2 Spatio-Temporal-Interest-Points (STIP)

Nell'approccio key-frame based il grosso dell'analisi viene fatto a monte, cioè individuando il key-frame più rappresentativo della sequenza e successivamente impiegarlo per l'analisi statica. Il secondo approccio consiste nell'utilizzare tutti i frame della sequenza ed estrarre un descrittore che inglobi informazioni temporali al suo interno. Si tratta quindi di modificare lo spazio su cui operano i descrittori di un frame ed ampliarlo considerando anche i frame adiacenti, cioè passare da uno spazio 2D ad uno 3D. In maniera simile agli operatori che lavorano in uno spazio bi-dimensionale, anche gli operatori 3D applicano banalmente un filtro al segnale video per ricavare i punti di interesse. Il video viene modellato tramite una funzione $I(x, y, t) : \mathbb{R}^3 \rightarrow \mathbb{R}$ la cui immagine è schematizzabile con un volume costituito dalla sequenza dei fotogrammi del video. Gli intorni dei massimi locali della risposta del filtro suddetto vengono poi estratti e ne viene creata una descrizione. Il clip viene così descritto da una collezione di volumi dimensionati in base alla scala del detector.

Negli ultimi anni sono stati realizzati svariati lavor che tipicamente sono estensioni di operatori già usati con successo nel caso 2D. Laptev et al. [9], ad esempio estendono l'operatore di Harris per la rilevazione degli angoli al caso spazio-temporale. Quest operatore presenta alte risposte in presenza di angoli spazio-temporali, dove: punti del volume $I(x, y, t)$ hanno ampie variazioni di intensità in direzioni ortogonali nello spazio e nel tempo. Questo tipo di operatore rileva pattern di moto relativi ad angoli spaziali che invertono

il proprio movimento, ad esempio la punta del piede agli estremi temporali dell'azione della corsa. Il sistema sviluppato da Laptev et al. è in grado di ottenere la scala caratteristica, estendendo il concetto di scale-space al tempo, e di adattare le regioni rilevate alla velocità del moto della telecamera. Tuttavia, è evidenza sperimentale, che la rappresentazione ottenuta è eccessivamente sparsa e che gli angoli spazio-temporali sono rari. Ke et al. [23] estendono il rilevatore di facce Viola&Jones al caso volumetrico definendo, per analogia al lavoro precedente, l'integral video e le features volumetriche. Il lavoro è orientato ad un riconoscimento real time delle azioni e ottiene una performance inferiore al lavoro di Schuldt et al. [24] basato invece sul detector e le feature di Laptev. Oikonomopoulos et al. [25] propongono un estensione del rilevatore di regioni salienti: questo lavoro, come quello di Laptev, ha il problema dell'eccessiva sparsità delle features rilevate.

La robustezza di una descrizione basata sull'orientazione del gradiente in due dimensioni è il principale motivo di successo del descrittore SIFT; la distintività di questo descrittore è data sia dall'uso delle orientazioni del gradiente pesate dal suo modulo, che dalla località degli istogrammi. Usare il gradiente fornisce robustezza per ciò che riguarda le variazioni di illuminazione; nel caso del descrittore SIFT, inoltre, la rappresentazione del punto tramite orientazioni permette di assegnare ad ognuno di essi una o più orientazioni e di utilizzarle come sistema di riferimento ottenendo così invarianza alle rotazioni. Cercando quindi di estendere questo operatore al 3D, l'obiettivo è calcolare il gradiente della funzione $I(x, y, t) : \mathbb{R}^3 \rightarrow \mathbb{R}$, in modo da codificare la variazione dell'aspetto locale nel tempo.

Il problema di rappresentare l'intorno del punto di interesse come istogramma delle orientazioni è stato affrontato in [10] [9]. Una metodica per la quantizzazione dell'angolo solido è proposta da Scovanner et al. [10]: nel

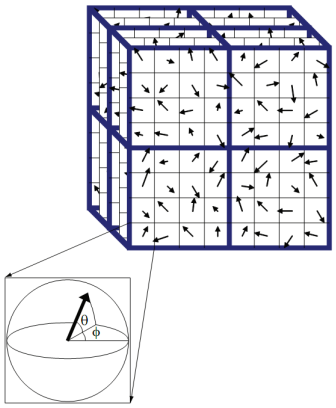


**Figura 2.11:** *Descrittore 3D SIFT.*

loro lavoro viene effettuata direttamente applicando un peso variabile in base all'intervallo. Così valori più vicini ai "poli" saranno pesati di più di valori più vicini all' "equatore" di modo da compensare la distorsione data dalla rappresentazione in coordinate sferiche.

Infine, come accennavamo nell'introduzione al capitolo, le features globali vengono utilizzate come elemento integrante delle features locali: in tal senso è interessante citare il lavoro di Wong et al. [12] perché utilizza tali informazioni per individuare le parti in movimento su cui estrarre le features tri-dimensionali.

# Capitolo 3

# Approccio proposto

Questo capitolo è dedicato alla descrizione delle tecniche implementate in questo lavoro di tesi. Per effettuare riconoscimento di concetti dinamici, cioè riconoscimento di azioni ed eventi, la maggior parte degli studi più recenti introducono l'aspetto temporale direttamente dentro il descrittore della sequenza video, analizzando più frame contemporaneamente e modificando i precedenti rilevatori di caratteristiche salienti. Un secondo metodo per modellare l'aspetto temporale è quello di analizzare a posteriori le tecniche già implementate per il riconoscimento di oggetti e scene, evitando così di modificare a basso livello i meccanismi di estrazione delle features, ma estendendo tali tecniche a più frame della sequenza e valutandone l'andamento. A questa seconda categoria appartengono le tecniche recentemente presentate in [7] [6] e [8], una volta recuperati i descrittori del video, viene effettuata un'elaborazione su tali caratteristiche lavorando sulla correlazione che vi è tra i frame.

L'obiettivo di questa tesi è quello di analizzare il descrittore di una sequenza comparandola all'analisi di un testo, identificando lettere, parole e il vocabolario di riferimento. Questo approccio rende possibile ricondursi a

situazioni su cui poter applicare le tecniche già sperimentate sull'analisi del testo, adattandole alle nuove circostanze visuali.

Nel dettaglio il lavoro svolto si articola come segue:

- estrazione delle features, cioè la scelta di quali descrittori utilizzare per descrivere il contenuto della scena;
- creazione del dizionario, che consiste nel applicare l'algoritmo K-Means per ottenere una descrizione sintetica dello spazio delle features;
- rappresentazione di un video, cioè la rappresentazione globale di una sequenza su cui poter analizzare le caratteristiche temporali;
- confronto tra frasi. Questa parte risulta essere quella di maggior interesse, in cui si descrive come sfruttare le tecniche di analisi testuale per ottenere un valore di confronto tra due sequenze distinte;
- classificazione. Ultima fase che utilizza le Support Vector Machine con kernel pre-calcolato.

## 3.1 Estrazione delle features

Il primo punto consiste nel cercare di trovare delle features che siano in grado di descrivere in modo efficiente una scena video: abbiamo pensato di utilizzare i SIFT, che sono i descrittori attualmente più utilizzati per la rappresentazione di immagini, e di cui sono forniti maggiori dettagli nella sezione 2.1.1. Nei sistemi che implementano il modello BoW, [1] [7] [6] [26] , ormai l'utilizzo di questi descrittori è divenuto lo standard proprio per la loro robustezza.

Poiché una sequenza video è composta da frame, ed ognuno di questi non è altro che un'immagine statica, la prima fase del lavoro coinvolge un'elaborazione di basso livello che ci permette di estrarre queste features dai frames di un video. Poiché i SIFT hanno la proprietà di essere invarianti rispetto alla dimensione e alla rotazione dell'immagine, e sono robusti a variazione di illuminazione, rumore, occlusione e cambiamenti di punti di vista, la scelta si è orientata in questa direzione.

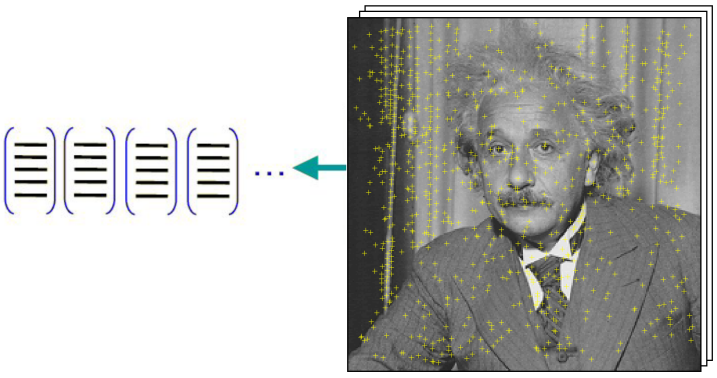

**Figura 3.1:** *Estrazione dei SIFT dai frame di un video*

A partire da un programma che estrae i SIFT, il passo successivo è stato quello di utilizzare delle librerie di OpenCv per l'implementazione di una funzione che ci permettesse la manipolazione dei frame, in modo poi da richiamare l'estrazione dei punti su ognuno di essi. Al termine di questa elaborazione, quello che si ottiene è un numeroso insieme di punti ricavati dall'intera sequenza video, opportunamente rappresentati da descrittori nel relativo spazio di features. Per evitare di calcolare ad ogni esperimento questa enorme mole di dati in funzione del dataset scelto, con significativa perdita di tempo, si è scelto di utilizzare delle librerie di serializzazione che ci hanno permesso di salvare, e soprattutto recuperare, le strutture calcolate per ogni video.

È necessario fare una considerazione su quello che abbiamo chiamato spazio delle features: i SIFT hanno dei descrittori che appartengono ad uno spazio a 128 dimensioni, per cui la mole dei dati che deve essere elaborata è notevole. Questa proprietà delle informazioni estratte non è da trascurare, in quanto risulta essere un punto cruciale nella realizzazione del codice, dal momento che si è resa necessaria l'ottimizzazione dell'occupazione di memoria per l'elaborazione dei dati. Questo spazio risulta influenzare la dimensione del dizionario, rendendo ingestibile l'elaborazione nel momento in cui si scelgono dizionari troppo grandi: per questo motivo sono state adottate delle tecniche particolari, che illustreremo nel seguito.

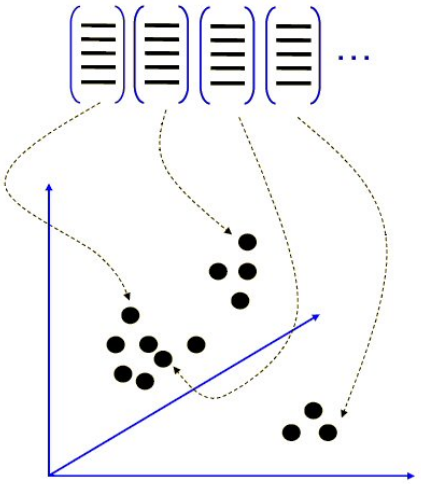

**Figura 3.2:** *Spazio delle features*

## 3.2 Creazione del dizionario

Dopo aver trovato una rappresentazione che descriva in modo opportuno una sequenza video, la fase successiva coinvolge la realizzazione e l'esecuzione di un algoritmo di clustering. Algoritmi di questo tipo, applicati a punti in

n-dimensioni, non fanno altro che creare dei *bin* e distribuire i punti all'interno di essi in base ad un criterio di "distanza": nel nostro caso, abbiamo utilizzato il noto algoritmo *K-Means*, calcolando la distanza di un punto da un altro attraverso la nozione di distanza euclidea (vedi Figura 3.2 eFigura 3.3).

L'algoritmo viene utilizzato nella fase iniziale del programma, in cui vengono presi in ingresso i punti di scene video precedentemente etichettate e selezionate, e calcola i centroidi dei vari cluster in base alla distribuzione di questi punti (il numero di cluster è un parametro scelto e/o noto a priori). I centroidi calcolati vengono salvati su disco e costituiscono il *dizionario di base*.

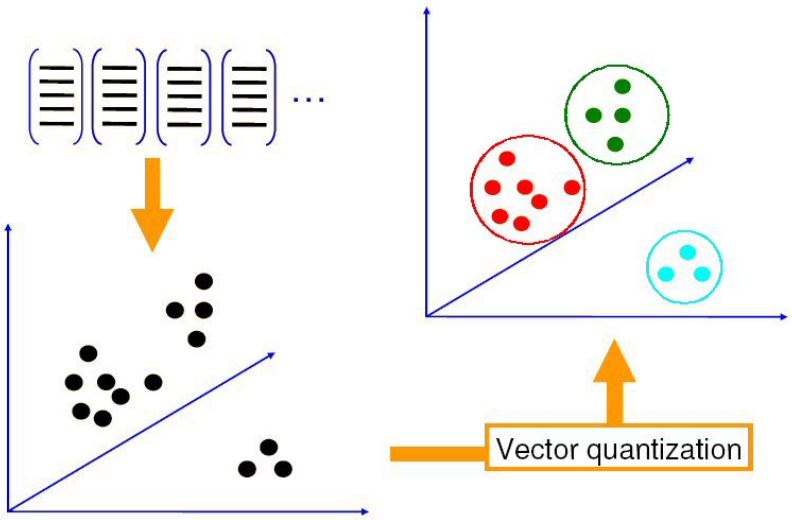


**Figura 3.3:** *Calcolo dei cluster*

## 3.3 Rappresentazione di un video

Il passo successivo, utilizzando il dizionario di base e i punti estratti da sequenze video sconosciute, è di assegnare ad ogni punto un' "etichetta"

che descriva in quale degli $n$ cluster ricade. Ricollegandosi alla descrizione dell'approccio di Bag-of-Words del capitolo 2.2, i due passi corrispondono rispettivamente alla creazione del dizionario e alla verifica di quali "parole" sono contenute nel "documento" da riconoscere, con la differenza che le parole del dizionario sono i cluster (e quindi maggiore è il numero di cluster e maggiore è la dimensione del dizionario), e che i documenti da analizzare sono le azioni e gli eventi. Vengono quindi creati degli istogrammi di frequenza per ogni frame che quantificano il numero di punti per ogni elemento del dizionario. Quindi ad ogni sequenza video verranno associati tanti istogrammi quanti sono i frame della sequenza, generando quella che abbiamo chiamato *frase*. Per capire come questo algoritmo sia utilizzato per creare una descrizione di video, cerchiamo di fare un'astrazione: immaginiamo un video come una sequenza di immagini, che rappresentano i vari frame. Ogni frame è caratterizzato da un insieme di SIFT, quindi se il video ha $n$ frame, per ciascuno di essi abbiamo individuato una serie di punti. L'algoritmo di cluster non fa altro che studiare come varia la distribuzione di questi punti frame per frame: supponiamo di avere note le coordinate dei centroidi, in questo modo ognuno di questi punti verrà collocato nel cluster più vicino in termini di distanza. Usiamo un "toy example" per cercare di far capire meglio quello che è l'obiettivo del nostro progetto. Immaginiamo i nostri frame uno accanto all'altro: l'insieme delle frasi possiamo pensarle come dei "tubi", uno per visual word, che si disegnano nel tempo, come schematizzato in Figura 3.4.

Le curve che questi "tubi" modellano nel tempo indicano come i punti vengono distribuiti dall'algoritmo: in questo modo dovrebbe chiarirsi se la rappresentazione di scene video attraverso i SIFT risulta essere ottimale o meno. L'idea potrebbe essere, per esempio, quella di verificare se tutti i punti

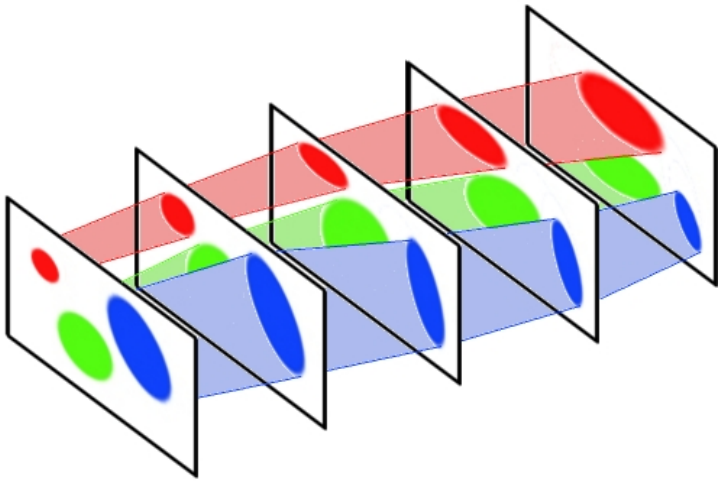

**Figura 3.4:** *Variazione della distribuzione dei SIFT nel tempo.*

che appartengono al volto di un giocatore rimangono nello stesso cluster per tutta la sequenza video.

Riportando l'attenzione sulla fase iniziale, quello che otteniamo è la rappresentazione di una sequenza video attraverso l'utilizzo di un dizionario: un video può essere visto come una *frase* composta da più *parole*, esattamente una per frame. Una parola non è altro che un vettore di frequenza che descrive la distribuzione dei SIFT estratti dal relativo frame sul dizionario preso in considerazione. Una frase risulta quindi essere un insieme di parole, tante quante sono i frame della sequenza video in esame.

### 3.3.1 Soft-Weighting

Come abbiamo accennato precedentemente, la dimensione del dizionario è un parametro cruciale, per cui, se troppo piccolo ingloba troppe features e perde di significatività, se troppo grande, invece, si specializza troppo sulle features di apprendimento e non generalizza i concetti in esame. Un problema riscontrato è che aumentando il dizionario si incrementa il tempo di

calcolo in maniera consistente e, considerando che l'elaborazione già ha un grosso dispendio di risorse e di tempo di calcolo, effettuare dei test con dizionari molto elevati, ma anche adeguati allo spazio delle features, risulta essere improponibile. Per questo motivo è stata ripresa una tecnica di assegnamento delle features all'interno del dizionario leggermente più elaborata, proposta da Ngo et al. [26]. In questo articolo viene valutato l'impatto di vari fattori sulle performance dell'approccio BoW. Questi fattori sono:

- keypoint detector. Sono stati esaminati 6 keypoint detectors: LoG, DoG, Harris Laplace, Hessian Laplace, Harris Affine, Hessian Affine. Originariamente è stata condotta una valutazione utilizzando come descrittore della regione del keypoint quello utilizzato dai SIFT a 128 dimensioni. Dalle varie considerazioni emerge che DoG risulta essere il miglior detector.

- dimensione del dizionario. La dimensione del vocabolario è critica e sconosciuta. Una dimensione eccessivamente piccola raggruppa troppi keypoints insieme generalizzando troppo il concetto, una dimensione grande rischia di particolareggiare troppo. La conclusione in riportata in [26] è che con l'introduzione dei Soft-weighting, descritti in seguito, la dimensione del vocabolario risulta essere mitigata, permettendo di utilizzare dizionari di grandezza inferiore e con il conseguente vantaggio di tempo di calcolo e di elaborazione.

- schema dei pesi delle visual words. L'assegnamento di ogni keypoints ad un elemento del vocabolario è un aspetto che viene approfondito e la soluzione proposta è quella di lasciare un margine di scelta, assegnando un vettore (dimensione impostata sperimentalmente a 4) che rappresenta i clusters più vicini. Mantenendo questa informazione si

cerca di svincolarci dagli schemi classici adottati fino ad ora, che erano stati direttamente recuperati dal dominio del text retrieval.

Vengono anche utilizzate diverse funzioni di kernel nella fase di classificazione con SVM. Infine viene esaminata la fusione tra BoW con features globali, quali momenti di colore e wavelet texture. L'unione viene fatta attraverso la "late fusion", ovvero fondendo gli output dei due detectors. Dai risultati ottenuti, Ngo et al. in [26] deducono che le tecniche di BoW con features locali combinate con features globali forniscono un incremento significativo delle performance che si attesta intorno al 50%. Questo porta a concludere che le tecniche di BoW non siano efficaci in se, ma complementari alle ben note features globali adottate nel content-based retrieval.

Sfruttando quindi il lavoro di Ngo et al., per ogni keypoint dell'immagine, invece di cercare solo l'elemento del dizionario più vicino, vengono selezionati $N$ parole del dizionario su cui il punto viene distribuito in maniera pesata. Supponendo di avere un vocabolario visuale di $K$ visual words, usiamo un vettore di dimensione $K$, $T = [t_1, ...t_K]$, dove ogni componente $t_k$ rappresenta il peso della visual word $k$ in un'immagine tale che:

$$t_k = \sum_{i=1}^{N} \sum_{j=1}^{M_i} \frac{1}{2^{i-1}} sim(j,k) \tag{3.1}$$

dove $M_i$ rappresenta il numero di keypoints classificati a distanza $i$th dalla parola $k$. La misura $sim(j,k)$ rappresenta la similarità tra il keypoint $j$ e la parola visuale $k$, cioè semplicemente la distanza euclidea. Notare che nell'equazione 3.1 il contributo di ogni punto è dipendente dalla similarità con l'elemento del dizionario pesato da $\frac{1}{2^{i-1}}$, cioè da un valore che indica a quale posto si è classificato il punto in base alla distanza dalle parole del dizionario.

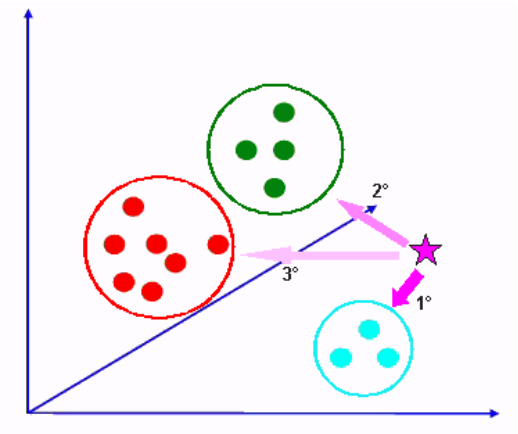


**Figura 3.5:** *Assegnazione di un keypoint a più parole visuali del vocabolario*

Usando lo schema proposto per cercare di mitigare uno svantaggio importante derivante direttamente dal dominio del text retrieval, Ngo et al. [26], hanno ottenuto un incremento delle prestazioni significativo ed è per questo motivo che è stato preso in considerazione per questo lavoro di tesi.

La scelta del parametro $N$ è stata presa sulla base dei risultati del lavoro di Ngo et al., valutando se effettivamente vi era un miglioramento nelle prestazioni o se questo accorgimento, applicato al nostro contesto, sarebbe risultato inutile.

### 3.3.2 Normalizzazione delle parole

Bisogna fare una considerazione sulla creazione di ogni singola "word" della frase associata ad un video. Dal momento che il numero di punti caratteristici estratti da un frame non è noto a priori e la varianza di tale quantità può essere anche molto elevata, in dipendenza dal dataset scelto, si rende necessario effettuare un passo di normalizzazione. A tale scopo sono state sperimentate le due normalizzazioni più comuni:

$$\sum_i x_i \tag{3.2}$$

$$\sqrt{\sum_i x_i^2} \tag{3.3}$$

I risultati ottenuti tendono a premiare la normalizzazione 3.3, ma, non essendo miglioramenti significativamente importanti, abbiamo optato per mantenere una versione meno complessa. Inoltre, utilizzando 3.2, durante il confronto tra le frasi, descritto successivamente, otteniamo dei vantaggi in termine di selezione dei parametri. Questo guadagno verrà discusso nel capitolo 4 durante la descrizione dei parametri, sezione 4.2.

## 3.4 Confronto tra frasi

Il passo immediatamente successivo riguarda la scelta delle metriche che determinano la *distanza* tra due video. Dato un dizionario di riferimento e dopo aver eseguito i passi descritti precedentemente, cioè aver estratto i SIFT e calcolato la frase equivalente per ogni video, occorre valutare se due sequenze rappresentano eventi simili o diversi. Nel nostro caso, data una coppia di video che rappresentano la stessa azione, occorre determinare una metrica che ci permetta di avere come risultato una distanza nulla o comunque la minima possibile, e, viceversa, data una coppia di eventi diversi dobbiamo ottenere una distanza sufficientemente grande da capire che i due video trattano azioni distinte. Questo è il punto in cui risultano utili le tecniche di *Text Retrieval*. Infatti, come abbiamo evidenziato, abbiamo chiamato il descrittore associato ad un video *frase* e ogni elemento del dizionario *parola*. Questo è stato fatto per richiamare l'attenzione sulle analogie che ci sono tra i precedenti campi di applicazione, cioè quello testuale, e l'ambiti di questa

ricerca. Se invece di utilizzare l'alfabeto, latino o di un'altra lingua, utilizzassimo un altro alfabeto creato specificatamente per un certo ambiente, le tecniche sviluppate per l'analisi testuale daranno gli stessi risultati? Ponendoci proprio questa domanda, siamo andati ad esplorare il mondo testuale cercando di recuperare tali tecniche e di riproporle nel nostro settore di ricerca, cioè quello multimediale. Tra i molteplici lavori, citiamo solo quelli che sono stati presi di riferimento e che poi andremo ad descrivere più in dettaglio, M. Neauhaus & H. Bunke [27] e H. Lodhi et al. [28].

### 3.4.1 Edit Distance (ED)

In [27] viene introdotto un metodo alternativo per la classificazione di testi usando le funzioni kernel basate sull'*Edit Distance* (ED). L'*ED* tra due stringhe di caratteri è il numero di modifiche elementari richieste per trasformare una stringa nell'altra e viceversa. Più formalmente:

**Definizione 1.** *Dato un alfabeto di riferimento $V$ di simboli. Una stringa $t$ è definita come una sequenza di simboli $\in V$ di lunghezza finita, tale che:*

$$t = t_1...t_n \in V^* = \bigcup_{i=0}^{\infty} V^i \tag{3.4}$$

dove

$$V^0 = \{\epsilon\}, n \geq 0 \tag{3.5}$$

identificando in $\epsilon$ la stringa vuota, $V^i$ è l'insieme di stringhe di lunghezza $i$ su $V$ e $V^*$ denota l'insieme di tutte le sequenze finite di simboli in $V$.

Una sequenza di modifiche elementari che trasformano la stringa $t$ in $t'$ è chiamato *percorso* da $t$ a $t'$ e $e(t, t')$ identifica l'insieme di tutti i possibili percorsi da $t$ a $t'$. Per misurare l'impatto di una modifica elementare, viene

definita una funzione $c$ che assegna un costo non negativo $c(w) \in \mathbb{R}^+ \cup \{0\}$ ad ogni modifica elementare $w$. L'idea chiave è che l'importanza delle operazioni è direttamente mappata al loro costo, in modo tale che modifiche irrilevanti avranno costi bassi e modifiche di rilievo, invece, avranno costi elevati. Il costo di un percorso può quindi essere calcolato come somma delle modifiche elementari:

$$d(t, t') = \min_{(w_1 \dots w_k) \in e(t,t')} \sum_{i=1}^{k} c(w_i) \tag{3.6}$$

Piccoli valori di $d$ indicano che sono necessarie poche modifiche elementari per trasformare due stringhe, quindi le due stringhe sono simili, mentre un alto valore di $d$ significa che, la quantità di modifiche necessarie per trasformare $t$ in $t'$, è molto elevata e quindi le due stringhe sono molto differenti. Questo valore viene utilizzato per definire la funzione kernel (descritta nella sezione 3.5) in fase di classificazione.

Come è possibile intuire, vi sono svariati modi di confrontare due stringhe e le vari tecniche si differenziano anche solo per i parametri assegnati all'interno della trasformazione adottata. Nello svolgimento del lavoro abbiamo implementato due metriche: la *Distanza di Levenshtein* e la *Distanza di Needleman-Wunch*. Queste due metriche sono state scelte perché permettono di confrontare stringhe di lunghezza differente, che nel nostro caso si traduce in video di durata diversa (quindi un numero di frame differenti), e risultano essere le due metriche principali della categoria. Di seguito le affrontiamo nel dettaglio fornendo pro e contro ed esempi esplicativi.

### Levenshtein Distance

Nella teoria dell'informazione e nella teoria dei linguaggi, la distanza di Levenshtein, o comunemente edit distance, è una misura per determinare quan-

to due stringhe siano simili. La distanza di Levenshtein tra due stringhe A e B è il numero minimo di modifiche elementari che consentono di trasformare A in B. Per modifica elementare si intende:

- la cancellazione di un carattere;
- la sostituzione di un carattere con un altro;
- l'inserimento di un carattere.

L'algoritmo procede iterativamente, considerando che i calcoli coinvolgono una matrice di dimensione elevate, e utilizza un vettore monodimensionale che viene aggiornato ad ogni passo. Riportiamo un esempio di calcolo di questa distanza tra due stringhe.

| | | s | a | m | | c | h | a | p | m | a | n |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| s | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| a | 2 | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| m | 3 | 2 | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | 4 | 3 | 2 | 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| j | 5 | 4 | 3 | 2 | 2 | 2 | 3 | 4 | 5 | 6 | 7 |
| o | 6 | 5 | 4 | 3 | 3 | 2 | 3 | 4 | 5 | 6 | 7 |
| h | 7 | 6 | 5 | 4 | 4 | 3 | 3 | 4 | 5 | 6 | 6 |
| n | 8 | 7 | 6 | 5 | 5 | 4 | 4 | 4 | 5 | 6 | 7 |
| | 9 | 8 | 7 | 6 | 5 | 5 | 5 | 5 | 5 | 6 | 7 |
| c | 10 | 9 | 8 | 7 | 6 | 5 | 6 | 6 | 6 | 6 | 7 |
| h | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 6 | 7 | 6 | 7 |
| a | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 6 | 7 | 7 |
| p | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 6 | 7 |
| m | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 6 |
| a | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 |
| n | | | | | | | | | | | |

**Figura 3.6:** *Esempio di calcolo della distanza di Levenshtein*

La distanza calcolata tra la stringa 'sam chapman' e 'sam john chapman', riportata in Figura 3.6, ha come valore quello riportato nell'ultima cella a

destra: questo indica che sono necessarie 5 modifiche elementari per fare il match completo delle due stringhe.

Benché questa distanza rappresenti una buona metrica, semplice ed intuitiva, presenta alcuni limiti sul valore calcolato, che possono essere riassunti nei seguenti punti:

- risulta essere almeno la differenza tra le lunghezze delle due strighe;
- è 0 se e solo se le due stringhe sono identiche;
- se le lunghezze delle due stringhe sono uguali, la distanza di Levenshtein non supera la distanza di Hamming, cioè è pari alla lunghezza delle strighe;
- il limite superiore è pari alla lunghezza della stringa più lunga.

#### Needleman-Wunch Distance

Questa seconda metrica di valutazione della distanza tra due stringhe note A e B, è comunemente utilizzata in bioinformatica per l'allineamento di sequenze di amminoacidi di due proteine. L'algoritmo prevede un confronto tra A e B che utilizza la stessa tecnica proposta dalla distanza di Levenshtein, con la differenza che in questo caso viene associato un peso diverso a seconda che venga effettuato un inserimento, una cancellazione o una sostituzione. Mostriamo con un'immagine un esempio di calcolo di questa distanza in cui inserimento e cancellazione pesano il doppio della sostituzione. Anche in questo caso il valore della distanza è quello che si trova nell'ultima cella in basso a destra, Figura 3.7.

Notiamo che il valore è esattamente il doppio di quello calcolato con la Levenshtein, e questo giustifica il peso diverso associato alle operazioni da eseguire per raggiungere il match delle due stringhe.

| | s | a | m | | c | h | a | p | m | a | n |
|---|---|---|---|---|---|---|---|---|---|---|---|
| s | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
| a | 2 | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |
| m | 4 | 2 | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 |
| | 6 | 4 | 2 | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 |
| j | 8 | 6 | 4 | 2 | 1 | 3 | 5 | 7 | 9 | 11 | 13 |
| o | 10 | 8 | 6 | 4 | 3 | 2 | 4 | 6 | 8 | 10 | 12 |
| h | 12 | 10 | 8 | 6 | 5 | 3 | 3 | 5 | 7 | 9 | 11 |
| n | 14 | 12 | 10 | 8 | 7 | 5 | 4 | 4 | 6 | 8 | 9 |
| | 16 | 14 | 12 | 10 | 9 | 7 | 6 | 5 | 5 | 7 | 9 |
| c | 18 | 16 | 14 | 12 | 10 | 9 | 8 | 7 | 6 | 6 | 8 |
| h | 20 | 18 | 16 | 14 | 12 | 10 | 10 | 9 | 8 | 7 | 7 |
| a | 22 | 20 | 18 | 16 | 14 | 12 | 10 | 11 | 10 | 8 | 8 |
| p | 24 | 22 | 20 | 18 | 16 | 14 | 12 | 10 | 12 | 10 | 9 |
| m | 26 | 24 | 22 | 20 | 18 | 16 | 14 | 12 | 10 | 12 | 11 |
| a | 28 | 26 | 24 | 22 | 20 | 18 | 16 | 14 | 12 | 10 | 12 |
| n | 30 | 28 | 26 | 24 | 22 | 20 | 18 | 16 | 14 | 12 | 10 |

**Figura 3.7:** *Esempio di calcolo della distanza di Needleman-Wunch*

Nel lavoro svolto, è stato fondamentale cercare di ottenere dei valori di distanza che fossero il più possibile scorrelati dalla lunghezza dei video in esame (quindi dal numero di frame). Come evidenziato nella sezione precedente, questa distanza è altamente condizionata dalla lunghezza delle stringhe da confrontare, e questo per noi rappresenta un limite perché due sequenze video possono rappresentare la stessa azione, a prescindere dalla loro lunghezza. Per tale motivo abbiamo deciso di implementare la metrica descritta, andando a valutare diversamente inserimento e cancellazione: queste pesano entrambe la metà rispetto alla sostituzione. In questo modo il valore di distanza ottenuto risulta essere maggiormente indipendente dal numero di frame esaminati.

### Passo di normalizzazione

Al termine del calcolo dell'ED è stato aggiunto un ulteriore passo, che possiamo definire come passo di normalizzazione della distanza calcolata: poiché il calcolo coinvolge una coppia di video, per scorrelare maggiormente questo valore dalla dimensione degli stessi, il passo di normalizzazione inserito prevede:

$$d(S_s, S_p) = \frac{NW(S_s, S_p)}{min(length(S_s), length(S_p))} \tag{3.7}$$

dove al numeratore troviamo la distanza di Needleman-Wunch tra le due stringhe e al denominatore il parametro di normalizzazione non è altro che il numero di frame del video di lunghezza minore.

### Confronto tra caratteri

Nelle metriche appena descritte, facendo riferimento a caratteri testuali, definire se due caratteri sono uguali o meno risulta semplice e ben definito. Queste nascono come metriche di confronto tra stringhe, e quindi caratteri appartenenti ad un alfabeto finito, per cui la comparazione di due lettere può dare come esito un valore positivo o negativo. Nel nostro caso il confronto avviene tra due valori numerici, e non possiamo permetterci di dire che la comparazione dia esito positivo solo nel caso di valori esattamente identici. In questo senso, è stato concepito l'uso di tecniche per eseguire un confronto e determinare il risultato in base ad una soglia di riferimento, al di sotto della quale si considerano le due parole uguali, altrimenti diverse. Considerando l'ambito visuale su cui vogliamo riportare tali procedure, i caratteri da considerare sono degli istogrammi di frequenza di dimensione uguale a quella del dizionario. Nasce quindi il problema di come confrontare due istogrammi.

Sono stati eseguiti diversi test per individuare la tecnica migliore, di cui sotto riportiamo un breve riassunto:

- **Chi-Quadro**, $d(H_i, H_j) = \sum_l (\frac{(H_i(l)-H_j(l))^2}{H_i(l)+H_j(l)}$, dove $H_i$ e $H_j$ sono istogrammi;

- **Mahalanobis**, $dist(x, A) = (x - \mu) \sum^{-1} (x - \mu)$, con $x$ coordinate del punto, $A$ insieme di punti, $\sum$ matrice di covarianza dell'insieme e $\mu$ valore medio dell'insieme. La distanza calcola quanto è probabile che il punto $x$ appartenga a tale distribuzione tenendo conto della covarianza dell'insieme di punti (e quindi della sua distribuzione statistica);

- **Intersection**, per ogni coppia di bin corrispondenti si considera quello a valore minimo (che dunque corrisponde all'intersezione tra i due). Poi si prende la somma di tali valori. $d(H_i, H_j) = \sum_l \min(H_i(l), H_j(l))$;

- **Correlazione**, è una misura di indipendenza lineare tra due oggetti. $d(H_i, H_j) = \frac{\sum_l (H_i(l) * H_j(l))}{\sqrt{\sum_l (H_i^{'}l)^2 * \sum_l H_j^{'}l)^2)}}$, con $H_k(l) = H_k(l) - \frac{1}{N} * \sum_w H_k(w)$ e $N$ numero di bin dell'istogramma;

- **Bhattacharyya**, $d(H_i, H_j) = \sqrt{(1 - \sum_l (\sqrt{(H_i(l) * H_j(l))}))}$, anch'essa una misura di correlazione;

- **Test di KolmogorovSmirnov**, nel caso in cui non sia possibile assumere alcuna distribuzione sui campioni, bisogna ricorrere a dei test non parametrici, come questo test, che confronta due distribuzioni cumulative.

I risultati dei test preliminari su queste tecniche di confronto hanno evidenziato che il **Chi-Quadro** fornisce un comportamento migliore degli altri:

per questo motivo è stato adottato come metodo di confronto, senza eseguire successivi test.

Ultima considerazione riguarda il fatto che queste tecniche restituiscono comunque un valore numerico che determina quanto due istogrammi siano simili o meno, ma non risolve il problema iniziale del confronto tra due caratteri. Praticamente semplificano il problema, richiedendo l'introduzione di una soglia di confronto che ci accompagnerà per tutto il lavoro e che determina il livello di uguaglianza che vogliamo associare a due istogrammi.

### 3.4.2 SubString kernel (SSK)

L'altro approccio che abbiamo sperimentato è quello che viene illustrato in [28]. In questo articolo viene proposto un nuovo metodo per classificare documenti di testo che fa uso di un kernel particolare. La funzione kernel (definita nella sezione successiva) è un prodotto interno nello spazio delle features generato da tutte le sottosequenze di lunghezza $k$. Una sotto sequenza è una qualsiasi sequenza ordinata di $k$ caratteri presenti nel testo non necessariamente contigui. Le sotto sequenze sono pesate in modo esponenziale da un fattore di decadimento $\lambda$ in funzione della loro distanza all'interno del testo, in modo da enfatizzare e dare maggior rilievo alle sequenze che nel testo sono contigue. Il calcolo diretto di questa funzione potrebbe comportare un tempo di elaborazione proibitivo anche per valori di $k$ relativamente bassi, in quanto la dimensione dello spazio delle features cresce esponenzialmente con $k$. In [28] vengono messe a punto delle tecniche di programmazione dinamica, atte a ridurre e rendere efficiente il calcolo di questa funzione. Purtroppo l'uso questo approccio nell'ambito multimediale, anche sfruttando tutti gli accorgimenti di ottimizzazione, se applicato a dataset di grandi dimensioni per ottenere risultati statisticamente corretti, richiede un tempo di calcolo

troppo elevato; questo ci ha costretto a ridurre il dataset a dimensioni molto limitate.

L'idea che sta alla base di questo approccio è di comparare due documenti testuali tramite le sotto stringhe che contengono: più sotto stringhe hanno in comune, più i due testi sono simili. Un aspetto importante è rappresentato dal fatto che le sotto stringhe non hanno bisogno di essere contigue e, il grado di contiguità di una sotto stringa nel documento, determina quanto sarà rilevante nella comparazione. Facciamo un esempio: la sotto stringa "**c-a-r**" è presente in entrambe le parole "**car**d" che "**c**ust**ar**d", ma con differente rilevanza proprio perché nella prima parola la sotto stringa è contigua, mentre nella seconda le lettere sono molto distanti le une dalle altre. Per ogni sotto stringa c'è una dimensione dello spazio delle features e il valore di ogni coordinata dipende da quanto frequentemente e quanto compattamente tale stringa è presente nel testo. Per tenere presente il grado di non contiguità della sotto stringa, è necessario introdurre un fattore di decadimento $\lambda \in (0, 1)$ che può essere usato per pesare la presenza delle features nel testo.

Facciamo ancora un esempio per spiegare meglio il concetto. Consideriamo le parole *cat, car, bat, bar*. Se prendiamo $k = 2$, otteniamo uno spazio delle features a 8 dimensioni dove le parole sono mappate come illustrato nella Figura 3.8.

| | c-a | c-t | a-t | b-a | b-t | c-r | a-r | b-r |
|---|---|---|---|---|---|---|---|---|
| $\phi$(cat) | $\lambda^2$ | $\lambda^3$ | $\lambda^2$ | 0 | 0 | 0 | 0 | 0 |
| $\phi$(car) | $\lambda^2$ | 0 | 0 | 0 | 0 | $\lambda^3$ | $\lambda^2$ | 0 |
| $\phi$(bat) | 0 | 0 | $\lambda^2$ | $\lambda^2$ | $\lambda^3$ | 0 | 0 | 0 |
| $\phi$(bar) | 0 | 0 | 0 | $\lambda^2$ | 0 | 0 | $\lambda^2$ | $\lambda^3$ |

**Figura 3.8:** *Esempio dello spazio generato considerando le parole cat, car, bat, bar e $k = 2$*

Così un kernel non normalizzato tra la parola *car* e *cat* è $K(car, cat) =$

$\lambda^4$, mentre la versione normalizzata è ottenuta dividendo per $K(car, car) = K(cat, cat) = 2\lambda^4 + \lambda^6$ ed ottenendo $K(car, cat) = \frac{1}{(2+\lambda^2)}$.

In questo caso non importa provare che il kernel soddisfa le condizioni di Mercer (matrice simmetrica e definita positiva) poiché deriva direttamente dalla definizione di prodotto interno.

Come accennavamo prima, il calcolo diretto del kernel, utilizzando sotto stringhe di lunghezza maggiore di 4, risulta essere impraticabile anche nel campo testuale senza l'adozione di particolari tecniche di ottimizzazione.

**Definizione 2.** (String Subsequence Kernel - SSK) *Definiamo $\sum$ essere un alfabeto finito. Una stringa è una sequenza finita di caratteri in $\sum$, incluso la sequenza vuota. Per le stringhe $s, t$, definiamo con $|s|$ la lunghezza della stringa $s = s_1...s_{|s|}$ e st la stringa ottenuta dalla concatenazione delle stringhe s e t. La stringa $s[i : j]$ è la sotto stringa $s_i...s_j$ di s. Diciamo che u è una sotto stringa di s se esistono gli indici $\boldsymbol{i} = (i_1, ..., i_{|u|})$, con $1 \leq i_1 < ... < i_{|u|} \leq |s|$, tale che $u_j = s_{i_j}$, per $j = 1, ..., |u|$ o abbreviando $u = s[\boldsymbol{i}] >$. La lunghezza $l(\boldsymbol{i})$ di una sotto sequenza s è $i_{|u|} - i_1 + 1$. Definiamo $\sum^n$ l'insieme di tutte le stringhe finite di lunghezza n e $\sum^*$ l'insieme di tutte le stringhe*

$$\sum^* = \bigcup_{n=0}^{\infty} \sum^n \tag{3.8}$$

*Definiamo ora lo spazio $F_n = \mathbb{R}^{\sum^n}$. La funzione di mapping $\phi$ per una stringa s è data dalla definizione di u coordinata $\phi_u(s)$ per ogni $u \in \sum^n$. Definiamo*

$$\phi_u(s) = \sum_{\mathbf{i}:u=s[i]} \lambda^{l(i)} \tag{3.9}$$

*per qualche $\lambda \leq 1$. Queste features misurano il numero di occorrenze della sotto stringa nella stringa s pesata in accordo con la sua lunghezza. Così, il prodotto interno dei vettori delle due stringhe s e t da una somma su tutte le*

*sotto sequenze comuni pesate in accordo con la loro frequenza di occorrenza e lunghezza:*

$$\begin{aligned} K_n(s,t) &= \sum_{u \in \sum^n} \langle \phi_u(s) \cdot \phi_u(t) \rangle = \sum_{u \in \sum^n} \sum_{\mathbf{i}: u=s[\mathbf{i}]} \lambda^{l(i)} \sum_{\mathbf{j}: u=t[\mathbf{j}]} \lambda^{l(\mathbf{j})} \\ &= \sum_{u \in \sum^n} \sum_{\mathbf{i}: u=s[\mathbf{i}]} \sum_{\mathbf{j}: u=t[\mathbf{j}]} \lambda^{l(\mathbf{i})+l(\mathbf{j})} \end{aligned} \tag{3.10}$$

Il calcolo diretto di queste features impiegherebbe $O(|\sum|^n)$ di tempo e di spazio, poiché questo è il numero di features coinvolte. Si è reso necessario adottare un meccanismo di calcolo ricorsivo, utilizzando una funzione di appoggio che, a partire dalla stringa nulla, aumenta progressivamente gli elementi considerati. Considerando gli accorgimenti di ottimizzazione adottati, la complessità del calcolo del kernel si riduce a $O(n|t||s|)$ per quanto riguarda il tempo di elaborazione. Si rimanda a [28] e [29] per maggiori dettagli.

Una volta che è stato creato il kernel di base, è necessario normalizzarlo per rimuovere qualsiasi rumore dipendente dalla lunghezza del documento. Viene così introdotta la funzione $\hat{\phi}(s) = \frac{\phi(s)}{\|\phi(s)\|}$. La funzione kernel risultante è:

$$\begin{aligned} \hat{K}(s,t) &= \left\langle \hat{\phi}(s) \cdot \hat{\phi}(t) \right\rangle = \left\langle \frac{\phi(s)}{\|\phi(s)\|} \cdot \frac{\phi(t)}{\|\phi(t)\|} \right\rangle = \frac{1}{\|\phi(s)\| \, \|\phi(s)\|} \langle \phi(s) \cdot \phi(t) \rangle \\ &= \frac{K(s,t)}{\sqrt{K(s,s)K(t,t)}} \end{aligned} \tag{3.11}$$

## 3.5 Classificazione

Sono stati fatti esperimenti utilizzando il classificatore Support Vector Machine (SVM), un insieme di metodi di apprendimento supervisionato per la

regressione e la classificazione di pattern, sviluppati negli anni '90 da Vladimir Vapnik, ed il suo team, presso i laboratori Bell AT&T. Appartengono alla famiglia dei classificatori lineari generalizzati e sono anche noti come classificatori a massimo margine, poiché allo stesso tempo minimizzano l'errore empirico di classificazione e massimizzano il margine geometrico.

Dato un insieme di istanze da classificare $x_i$ e le rispettive etichette $y_i \in [-1, 1]$ con $i = 1..l$, per ottenere l'iperpiano di separazione ottimale occorre risolvere il seguente problema di ottimizzazione:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=0}^{l} \xi_i \tag{3.12}$$

con il vincolo

$$y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i,$$

$$\xi > 0.$$

In questa formulazione le istanze $x_i$ sono mappate in uno spazio a più elevata dimensionalità (potenzialmente infinita) tramite la funzione $\phi$. La mappatura non deve essere esplicita; riformulando infatti il problema di minimizzazione come il duale del 3.12 le istanze $x_i$ appaiono solo in prodotti scalari (nello spazio delle feature):

$$\min_{\alpha} \frac{1}{2} \alpha w^T Q \alpha - e^T \alpha \tag{3.13}$$

con il vincolo

$$y^T \alpha = 0,$$

$$0 \leq \xi \leq C, i = 1, ..., l.$$

dove $e^T$ è il vettore le cui entrate sono tutte 1, $C > 0$ e $Q = y_i y_j K(x_i, x_j)$ è una matrice $l \times l$ semi-definita positiva. La funzione $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ calcola il prodotto scalare tra due vettori di feature direttamente nello spazio rimappato ed è detta **kernel**. Una volta risolto il problema 3.12 si può ottenere una predizione della classe dell'istanza $x$ tramite la funzione:

$$y = sgn\left(\sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b\right). \tag{3.14}$$

Al termine dell'ottimizzazione un sottoinsieme dei dati potrebbe avere $a_i = 0$ e quindi non contribuire nella 3.14; i restanti vettori sono detti vettori di supporto e nel caso $0 < a < C$ si trovano esattamente sul margine: nella Figura 3.9 appaiono cerchiati. Quindi il modello memorizzato è costituito unicamente dagli $a_i$ e $x_i$ con $i \in SV$, dove $SV$ è l'insieme degli indici dei vettori di supporto. Questa caratteristica delle SVM le rende intrinsecamente robuste all'iperadattamento, in quanto il modello, come visto, non dipende mai da tutti i dati ma solo da quelli che consentono di localizzare un iperpiano ottimale di separazione delle istanze.

Il secondo termine della funzione obiettivo del problema consente, tramite un peso $C$, di ottenere soluzioni al problema 3.12 anche in presenza di insiemi di dati non separabili. Se si usano kernel come RBF o $\chi^2$ avremo due parametri liberi nel modello: $\gamma$ e $C$. Per determinarne i valori ottimali viene tipicamente effettuata una procedura di cross-validazione sul training set variando i parametri del modello su di una griglia logaritmica (i.e. $C = 2^{-5}, 2^{-4}...2^{15}, \gamma = 2^{-15}, 2^{-14}...2^{6}$). La coppia di valori che ha fornito il minore errore di classificazione durante la cross-validazione viene poi usata per riaddestrare il modello sull'intero training set.

Una funzione kernel rappresenta un prodotto scalare tra i due vettori nello spazio rimappato ovvero:

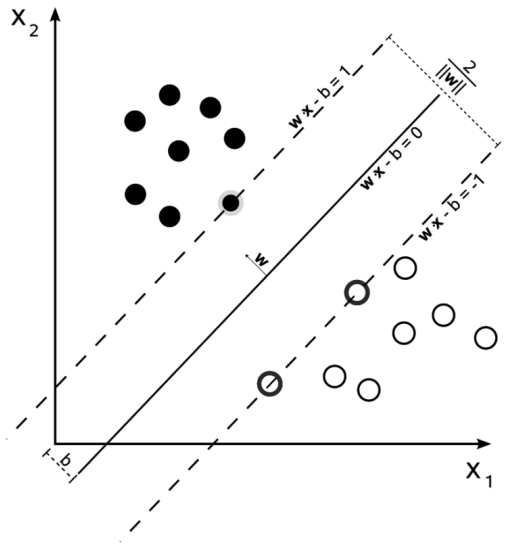


**Figura 3.9:** *Iperpiano ottimo per un insieme linearmente separabile in* $\mathbb{R}^2$

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \,; \tag{3.15}$$

affinché una funzione possa essere utilizzata come kernel occorre che soddisfi il requisito di validità:

**Definizione 3.** *Sia* $X$ *un insieme. Una funzione simmetrica* $K : X \times X \to \mathbb{R}$ *è un kernel definito positivo su* $X$ *se* $\forall n \in \mathbb{Z}^+, x_1 ... x_n \in X$ *e* $c_1 ... c_n$ *vale* $\sum_i^n \sum_i^n c_i c_j K(x_i, x_j) > 0$.

Il kernel $\chi^2$ rappresenta una generalizzazione del kernel radiale ed è indicato da Lazebnik et al. [60] per la classificazione di istanze descritte con istogrammi; la validità di questo kernel è dimostrata da Fowlkes [15]. Anche se i risultati presenti in letteratura per la categorizzazione di scene, oggetti e texture, con il kernel $\chi^2$ hanno dato i ottimi risultati l'approccio seguito in questo lavoro utilizza dei kernel pre-calcolati in base alle considerazioni che abbiamo fatto nei capitoli precedenti. Mentre utilizzando la seconda me-

trica, cioè SSK, il valore calcolato viene direttamente utilizzato come valore della funzione kernel, la cui validità è dimostrata, per quanto riguarda la metrica dell'ED invece, il valore restituito dal confronto è stato elaborato con la seguente formula:

$$K(t, t') = e^{-d(t,t')} \tag{3.16}$$

La misura dell'*edit distance*, tuttavia, in generale non soddisfa tutte le condizioni di validità del kernel, quindi l'approccio proposto non può essere ritenuto valido nel caso generale. Tuttavia, Hasdonk recentemente ha dimostrato in [30] che l'utilizzo di SVM con funzioni kernel che violano le condizioni di validità, può essere interpretato, in termini geometrici, come la separazione ottimale dell'inviluppo complesso nello spazio pseudo-euclideo. Si rimanda alla lettura di [27] e [30] epr ulteriori approfondimenti.

I vantaggi di questo tipo di classificatori sono principalmente dati dalla teoria matematica con cui sono costruiti. Il fatto che il problema 3.13 abbia un unico ottimo globale ha fatto preferire questo tipo di algoritmi di apprendimento rispetto ad altri più tradizionali (reti neurali). Un altro vantaggio è la loro natura di macchine a kernel, la quale permette di sfruttare questo algoritmo di apprendimento automatico per varie tipologie di dati; infatti vista l'equazione 3.13 è sufficiente formulare una funzione che soddisfi la proprietà della Definizione 3 sul nostro insieme di dati. Nel nostro caso sarebbe stato possibile usare il prodotto scalare (formulazione di SVM originale), ad esempio, per misurare la similarità tra due istogrammi. Data tuttavia la complessità del dato analizzato è facilmente spiegabile come sfruttare una rimappatura delle features in uno spazio a più elevata dimensionalità (kernel RBF standard) consenta di migliorare radicalmente le prestazioni. L'uso di un'estensione del popolare kernel RBF ($\chi^2$), esplicitamente creata allo

scopo di confrontare istogrammi, rappresenta la soluzione ideale al nostro problema.

#### Estensione multiclasse

Le SVM sono nativamente classificatori binari: sono in grado di apprendere l'iperpiano ottimale per la separazione di esempi positivi da negativi. Nei casi applicativi, ed in particolare nelle librerie digitali, ci si trova a dover classificare dati con più di due categorie. Le strategie possibili sono:

- one-vs-all;
- one-vs-one.

Supponiamo di avere $N$ classi; nel primo caso sono addestrati $N$ classificatori, ciascuno utilizzando come esempi positivi quelli di una classe e come esempi negativi quelli di tutte le altre. In fase di decisione viene scelta la classe che ottiene il massimo margine dall'iperpiano. Nel secondo caso vengono addestrati $N(N-1)/2$ classificatori, ciascuno addestrato a separare ciascuna coppia di classi. In fase di decisione vengono considerati gli esiti di ciascun classificatore come voti per una classe e viene scelta quella che ne ottiene la maggioranza.

Nel caso della strategia one-vs-one il tempo di addestramento può essere minore in quanto, nonostante si debbano addestrare più classificatori, i dataset usati per ciascuno sono di dimensioni di gran lunga minori rispetto a quelli usati nell'approccio one-vs-all. Nell'approccio one-vs-all inoltre si può incorrere in dataset sbilanciati: ad esempio se abbiamo 6 classi ciascuna con 100 filmati, ciascun classificatore avrà 100 esempi positivi e 500 negativi. Questo problema chiaramente può acuirsi in presenza di dataset già sbilanciati in partenza e al crescere delle classi del problema.

In questo lavoro di tesi è stata usata un'estensione della libreria libSVM [31][1].

[1]http://www.csie.ntu.edu.tw/ cjlin/libsvm/

# Capitolo 4

# Risultati

In questo capitolo vengono presentati i dataset utilizzati, la descrizione dei parametri sperimentali ed i risultati ottenuti con le tecniche illustrate nei capitoli precedenti. L'obiettivo è valutare se con il nostro approccio si ottengono dei miglioramenti significativi rispetto ai risultati ottenuti utilizzando un approccio Bag-of-Words classico, basato sull'analiso di key-frame, descritto nella sezione 2.3.1 e che riprenderemo brevemente nelle sezioni successive.

## 4.1 Dataset

Per effettuare gli esperimenti sono stati presi in considerazioni due dataset differenti: un dataset calcistico, di piccole dimensioni, e TRECVID 2005, uno dei più diffusi dataset attualmente disponibili.

Il dataset calcistico è stato creato a partire da 5 partite di calcio del Campionato italiano 2007-2008:

- Parma-Fiorentina;
- Cagliari-Catania;

- Parma-Catania;
- Siena-Livorno;
- Parma-Livorno.

All'interno di queste partite, sono state selezionate 4 classi di azioni: *rimesse*, *rinvii*, *punizioni* e *azioni da goal* (vedi Figura 4.1). Questa scelta permette

**Figura 4.1:** *In ordine da destra verso sinistra: azioni da goal, punizioni, rinvii e rimesse.*

di avere un dataset con una variabilità degli eventi sufficientemente ampia. Infatti, vi sono azioni che si svolgono quasi da ferme, tipo le rimesse, altre che iniziano ferme e proseguono con un movimento veloce, quali *punizioni* e *rinvii*, ed infine quelle che si sviluppano con un movimento continuo, come le *azioni da goal*.

Per quanto riguarda la classe *azione da goal* precisiamo che abbiamo considerato quelle azioni che partono circa da metà campo, si avvicinano velocemente alla porta e si concludono con un tiro, che non deve essere necessariamente un goal, ma comunque un tentativo da parte del giocatore. La selezione di un numero sufficiente di video per classe si è svolta "manualmente", con la visione di ogni partita, l'analisi della presenza delle varie azioni di interesse e la loro successiva estrazione in clip di dimensione compresa tra i 100 e i 350 frame 720x576. Per ogni classe sono state estratte 25 clip, per un totale di 100 clip, cercando di mantenere all'interno di ogni classe una certa variabilità nell'azione stessa: le *rimesse* sono state selezionate da ogni

lato del campo, i *rinvii* sia lato destro che lato sinistro, mentre le *azioni da goal* e le *punizioni*, che sono i concetti che si assomigliano di più, sono stati selezionati cercando di avere una distribuzione sul tutto il campo da gioco. Il dataset così formato è stato utilizzato in fase preliminare, dove sono stati eseguiti dei test per quanto riguarda la scelta iniziale di metriche e opzioni più promettenti, ad esempio per la scelta di quale metrica utilizzare nel confronto tra frasi (vedi capitolo 3.4) oppure per il confronto tra caratteri descritto nella sezione3.4.1.

Il secondo dataset, TRECVID 2005 [32], è stato appositamente creato per promuovere la ricerca del *Content-based Retrieval* fornendo una grande collezione di video digitali liberamente fruibili. Con un dataset di riferimento su cui poter comparare le nuove tecniche, vengono presentate delle linee guide per la valutazione delle metriche sperimentate e vengono anche forniti obiettivi aggiornati con lo stato dell'arte. Sponsorizzato dal National Institute of Standards and Technology (NIST) con il supporto delle agenzie governative statunitensi, il dataset ogni hanno subisce un incremento di dimensioni e di contenuti. Per i nostri esperimenti è stato scelto il dataset del 2005, composto da 169 ore di notiziari televisivi di diversa nazionalità (arabi, cinesi, statunitensi, etc. ) in formato MPEG-1 con dimensione 352x240. E' stato scelto di utilizzare la lista di eventi LSCOM (Large Scale Concept Ontology for Multimedia) utilizzato in [33]) [7], e dei 24 eventi disponibili, riassunti in tabella 4.1, ne sono stati selezionati 7 per poter ottenere un confronto con [7].

Essendo il dataset composto da filmati di dimensione variabile da 30 minuti ad un'ora, è stato necessario un'elaborazione preliminare, utilizzando l'annotazione LSCOM ed individuando le giuste porzioni di video da impiegare per i vari test.

**Figura 4.2:** *Dataset TRECVID 2005. Alcuni fotogrammi di azioni di guerriglia, persone che nuotano e veicoli in movimento.*

## 4.2 Scelta dei parametri

Vi sono numerosi parametri che possono essere variati all'interno del progetto per verificarne le prestazioni, ma, come abbiamo accennato precedentemente, considerando l'ammontare di tempo di calcolo necessario per eseguire un singolo test, è stato necessario effettuare una selezione utilizzando il dataset calcistico di piccole dimensioni. Questa scelta, imposta obbligatoriamente, ha lo scopo solo di individuare quali sono le scelte migliori, escludendo eventuali tecniche poco promettenti. Considerando che il dataset calcistico, molto piccolo ed avente solo filmati monotematici, tali risultati non hanno ri-

| | |
|---|---|
| Airplane_Crash | 574 |
| ***Airplane_Flying*** | 570 |
| Airplane_Landing | 570 |
| Airplane_Takeoff | 570 |
| Car_Crash | 4201 |
| Cheering | 548 |
| Dancing | 1027 |
| ***Demonstration_Or_Protest*** | 2052 |
| Election_Campaign_Debate | 497 |
| Election_Campaign_Greeting | 497 |
| ***Exiting_Car*** | 4201 |
| Fighter_Combat | 743 |
| Greeting | 394 |
| Handshaking | 132 |
| Helicopter_Hovering | 88 |
| People_Crying | 138 |
| ***People_Marching*** | 1937 |
| Riot | 2052 |
| ***Running*** | 6401 |
| Shooting | 1483 |
| Singing | 835 |
| ***Street_Battle*** | 1483 |
| Throwing | 56 |
| ***Walking*** | 6401 |

**Tabella 4.1:** *I 24 concetti annotati nel dataset TRECVID 2005. La colonna a destra indica il numero totale di shot presenti all'interno del dataset. I concetti evidenziati sono quelli utilizzati per gli esperimenti.*

levanza ai fini di uno studio statistico rivolto alla *Concept-based Recognition*. I parametri valutati e su cui sono stati effettuati i test preliminari sono:

- la metrica da utilizzare per il confronto tra le frasi: *Distanza di Levenshtein* e *Distanza di Needleman-Wunch*;
- l'introduzione o meno del passo di normalizzazione al termine del confronto tra due frasi (descritto nella sotto sezione "Passo di normalizzazione" in 3.4.1);

- la tecnica di confronto tra caratteri, quindi istogrammi, da utilizzare tra quelle descritte nella sotto sezione “Confronto tra caratteri” in 3.4.1;

- la normalizzazione da eseguire nella creazione di ogni singola “word”, come spiegato nella sezione 3.3.2.

Come già accennato nei capitoli precedenti, i risultati ottenuti confermano che l’introduzione della normalizzazione al termine del confronto tra due frasi, rispetto alla lunghezza minore delle stesse, porta un effettivo miglioramento delle prestazioni.

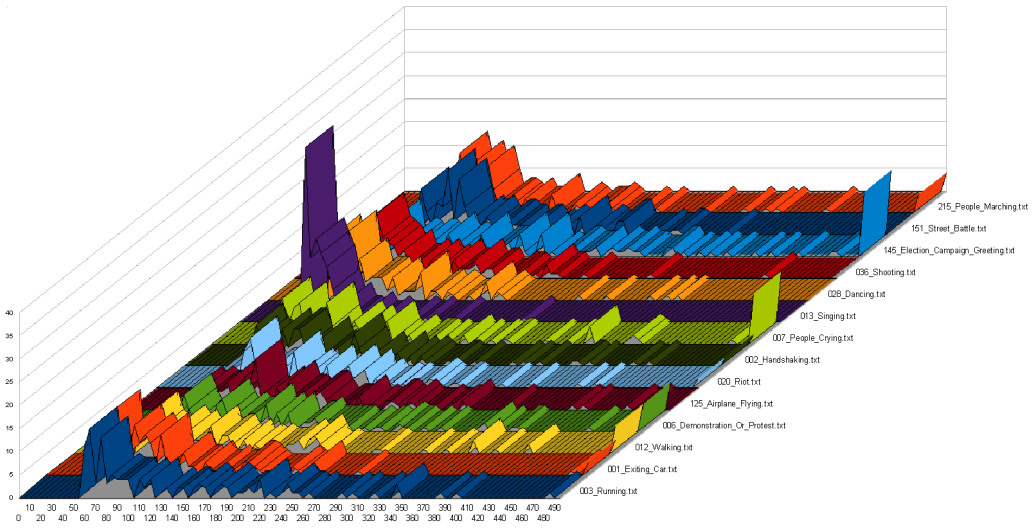


**Figura 4.3:** *Lunghezza delle clip suddivisa per classe di azione. La maggior parte delle clip hanno lunghezza compresa tra 50 e 150 fotogrammi, ma ci sono anche una serie rilevante di clip con lunghezza superiore a 150.*

Tra le due metriche descritte in 3.4.1, invece, abbiamo optato per modificare manualmente i pesi delle operazioni elementari, ed adottare uno schema che penalizza l’operazione di sostituzione piuttosto che quelle di inserimento e cancellazione. Il motivo di questa scelta è svincolarsi il più possibile dalla lunghezza delle clip, che variano molto anche all’interno del

dataset TRECVID 2005 (vedi Figura 4.3), e poter usufruire di un risultato indipendente dalla durata dell'azione coinvolta.

Per quanto riguarda la normalizzazione delle parole abbiamo adottato la versione più semplice, cioè la formula 3.2, anche se leggermente meno performante, per fornire dei risultati comparabili con delle successive modifiche atte a migliorare la descrizione di un video.

Infine, per le tecniche di confronto tra istogrammi, è stato scelto il calcolo del Chi-Quadro, con il quale si ottengono risultati migliori.

Terminata l'analisi preliminare, i parametri che sono stati esaminati utilizzando il dataset TRECVID 2005, sono:

- la dimensione del dizionario: questo è il parametro che influisce maggiormente sul tempo di calcolo. Infatti, aumentando questo valore vengono influenzati tutte le operazioni della procedura, indipendentemente dalla configurazione dei restanti parametri. Le dimensioni sondate sono: 30, 60, 100, 200, 300, 500, 1000; quest'ultimo solo parzialmente;

- la soglia di confronto con il Chi-Quadro: è il parametro con maggior incertezza, in quanto è possibile solo fare delle considerazioni sui limiti esterni. Avendo scelto di utilizzare la normalizzazione espressa dall'equazione 3.2, il risultato del Chi-Quadro rimane confinato tra i valori 0 e 2, cosa che non accade utilizzando la formula 3.3;

- campionamento dei video: questo è un parametro che è stato verificato anche con il dataset calcistico, e consiste nel selezionare, ad intervalli prestabiliti, i fotogrammi della sequenza video su cui effettuare l'elaborazioni. Questo approccio è stato introdotto, oltre che per diminuire i tempi di calcolo, per verificare la correlazione tra la continuità delle azioni e quella della sequenza. Infatti campionando con un passo di

2 fotogrammi al secondo, si perde più del 90% della sequenza video, aspettandoci prestazioni peggiori. I valori di campionamneto utilizzati nel dataset TRECVID sono: 5, 10, 15.

Questi parametri sono stati inseriti anche nella valutazione con il dataset calcistico, ma, con l'analisi preliminare effettuata su questo dataset, l'obiettivo è di scegliere le tecniche più promettenti tra quelle a disposizione e non stimarne i valori dei parametri per investigarne gli andamenti.

Tutti i parametri che sono stati elencati fino a questo punto, tranne la scelta dell'implementazione dell'ED, sono riferiti ad entrambe le tecniche di confronto tra le frasi di due video. Utilizzando l'analisi con le sotto stringhe, descritta nel capitolo 3.4.2, vengono introdotti altri due parametri di valutazione:

- la lunghezza delle sotto stringhe;
- il fattore di decadimento $\lambda$.

Considerando che questi due parametri si aggiungo ai precedenti, a causa della complessità computazionale di questa tecnica non è stato possibile eseguire sufficienti esperimenti per analizzare lo spazio di questi parametri.

## 4.3 Descrizione degli esperimenti

Per effettuare gli esperimenti sul dataset TRECVID 2005 si è reso necessario utilizzare solo parte dell'intero dataset a causa dei tempi di elaborazione. Nel dettaglio sono state selezionate 120 clip per ogni concetto preso in esame, per un totale di 1680 filmati. Considerando il numero di esempi totali è stata eseguita la cross-validazione con 3 fold. La cross-validazione è quella tecnica, utilizzabile in presenza di un dataset sufficientemente numeroso, che consiste

nella suddivisione degli esempi totali in k parti (nel nostro caso 3); ad ogni passo dell'esperimento la parte (1/k)-esima del dataset viene ad essere il validation dataset mentre la restante parte costituisce il training dataset. Facendo così per ognuna delle k parti si allena il modello, evitando quindi problemi di overfitting e di campionamento asimmetrico (e quindi affetto da bias) del training dataset, tipico della suddivisione del dataset in due sole parti (ovvero training e validation dataset). In ognuno dei 3 passaggi che vengono effettuati nel nostro procedimento, avremo quindi 1120 esempi di training e 560 di validation.

Come descritto in precedenza, è stato scelto di utilizzare il metodo di valutazione proposto da TRECVID, cioè l'*Average Precision* (AP). Considerando che la *precision* e la *recall* sono basate sull'intera lista di esempi del dataset, con l'AP si enfatizza l'ordine degli esempi positivi ritornati dal sistema, premiando maggiormente quelli recuperati prima. Questa matrice si esprime come la media della precisioni calcolate dopo aver troncato la lista ad ogni esempio positivo ritornato:

$$AP = \frac{1}{R} \sum_{j=1}^{N} \frac{R_j}{j} * I_j \tag{4.1}$$

dove $N$ è il numero totale di esempi e $R$ è il numero di esempi rilevanti. Con $R_j$ si indica il numero di esempi positivi recuperati, cioè la *precision* dei dati troncati al passo $j$. $I_j$ è la funzione binaria che indica se l'esempio è rilevante o meno. Il calcolo non viene eseguito per l'intero dataset, ma solo per i primi 1000 elementi ritornati dalla classificazione. Bisogna precisare che la classificazione viene eseguita "one-vs-all", considerando che le prestazioni sono equiparabili alla versione "one-vs-one".

Il primo passo è stato quello della creazione del dizionario di riferimento. La procedura consiste nel:

- selezionare casualmente 4 clip per concetto in modo da ottenere un quantità di SIFT compresa tra 100.000 e 150.000;
- applicare all'insieme di SIFT estratti l'algoritmo K-Means, descritto nel dettaglio nel capitolo 2.2.1;

| Dizionario | Camp | BoW classico | ED |
|---|---|---|---|
| 30 | 10 | 12.24% | 17.74% |
| | 15 | | 17.65% |
| 60 | 10 | 12.62% | **19.03%** |
| | 15 | | 18.10% |
| 100 | 10 | 13.31% | 18.50% |
| | 15 | | 18.33% |
| 200 | 10 | 11.39% | 17.34% |
| | 15 | | 17.83% |
| 300 | 10 | 13.88% | 17.22% |
| | 15 | | 18.56% |
| 500 | 10 | 13.89% | 17.11% |
| | 15 | | 18.57% |
| 1000 | | 14.98% | - |
| | | | - |

**Tabella 4.2:** *Tabella dei risultati delle MAP. La colonna "Camp" indica il campionamento eseguito sulle sequenze video.*

Il vincolo sul numero di SIFT è stato scelto in considerazione del fatto che tale quantità, in rapporto al tempo di elaborazione per il clustering, descrive sufficientemente bene lo spazio delle features.

Il secondo passo nel calcolo degli esperimenti è stato quello di creare una baseline da poter utilizzare per valutare le prestazione del nostro metodo.

Sono stati eseguiti degli esperimenti utilizzando l'approccio key-frame based descritto nel capitolo 2.3.1, selezionando il frame centrale di ogni clip video e calcolando l'istogramma con il modello BoW. Tale istogramma viene normalizzato e direttamente utilizzato con il classificatore SVM, impostando il kernel radiale (RBF):

$$K(x_i, x_j) = exp(-\lambda \left\| x_i - x_j \right\|^2) \tag{4.2}$$


**Figura 4.4:** *Grafico dei risultati ottenuti con MAP. Sulle ascisse troviamo la dimensione del dizionario, sulle ordinate i valori delle prestazioni ottenute.*

I risultati sono mostrati in Figura 4.4 (BoW) e in tabella 4.2 (BoW classico). Possiamo notare come al crescere della dimensione del dizionario le prestazioni migliorino. Questo incremento è dovuto alla miglior rappresentazione che viene generata con un vocabolario più ampio, la cui estensione interpreta meglio lo spazio dei descrittori SIFT. Tale comportamento è affermabile anche in altre applicazioni del Bag-of-Words, come il riconoscimento di oggetti.

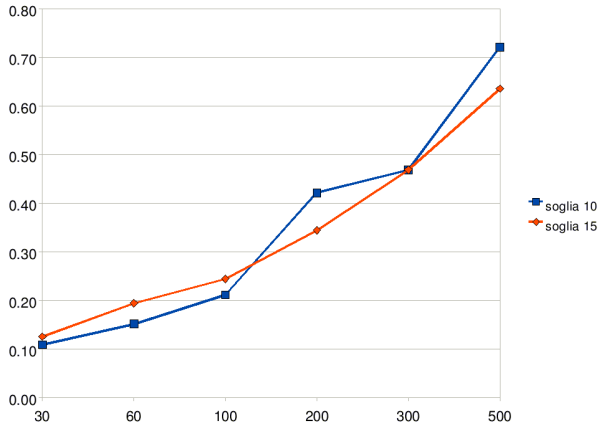


**Figura 4.5:** *Grafico delle soglie utilizzate all'interno dell'ED per definire se due "word" sono uguali. Si nota che al crescere del dizionario, indipendentemente dal campionamento della sequenza, il valore della soglia aumenta gradualmente.*

Il passo successivo consiste nell'eseguire i test utilizzando il nostro metodo, descritto nel capitolo 3, e sperimentando i parametri elencati nella sezione precedente. Eseguendo i test sul primo concetto, ci siamo accorti che utilizzare il campionamento ogni 5 frame risultava troppo dispendioso in termini di tempo, senza per altro dare significativi miglioramenti nei risultati, come mostrato in Figura 4.6. Di conseguenza abbiamo deciso di eliminare questo valore, anche considerando il fatto che tempi così lunghi (stimati intorno a decine di ore di calcolo) sono già di per se un fallimento dell'approccio sperimentato. Sono stati effettuati quindi gli esperimenti campionando i video ogni 10 e ogni 15 fotogrammi.

I risultati finali sono mostrati nel grafico 4.4 e in tabella 4.2, dove è stato evidenziato il valore migliore. I valori riportati sono calcolati come media dell'Average Precision (MAP) dei risultati ottenuti per i singoli concetti.

Analizzando nel dettaglio i parametri esaminati, notiamo che la soglia ottimale utilizzata nel confronto tra due "word", all'interno del calcolo dell'ED, aumenta con l'aumentare della dimensione del dizionario (vedi grafico

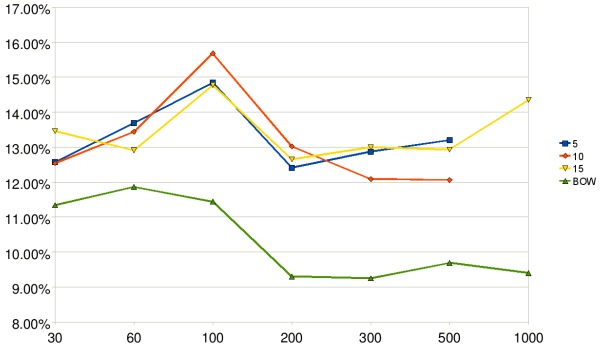


**Figura 4.6:** *Exiting_Car*

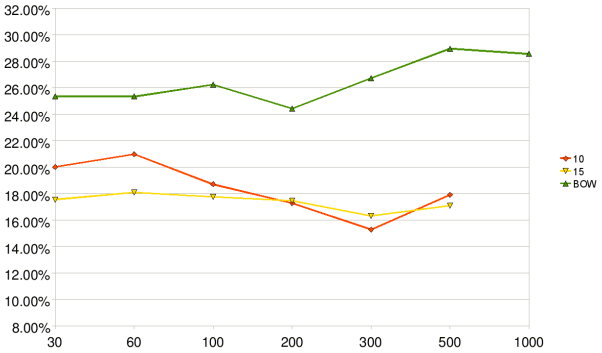


**Figura 4.7:** *Airplane_Flying*

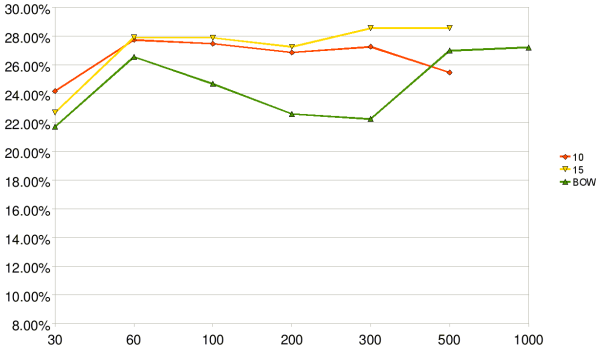


**Figura 4.8:** *Walking*

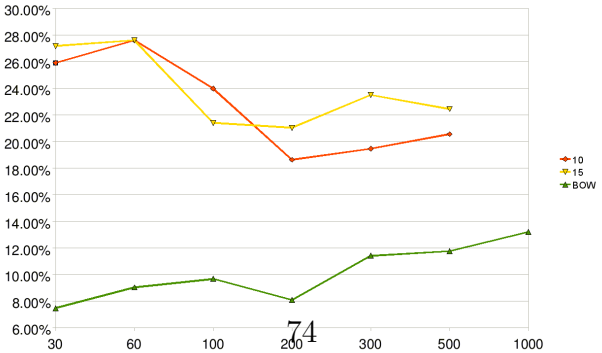




**Figura 4.9:** *Running*

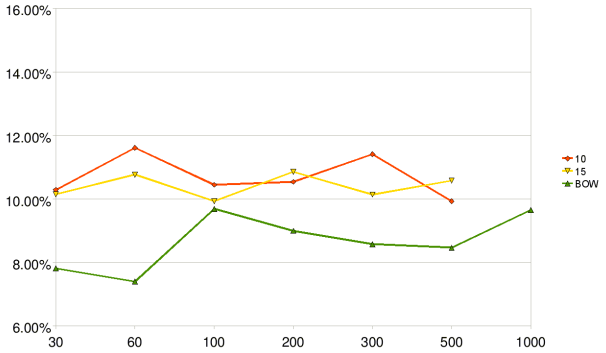


**Figura 4.10:** *Demonstration_Or_Protest*

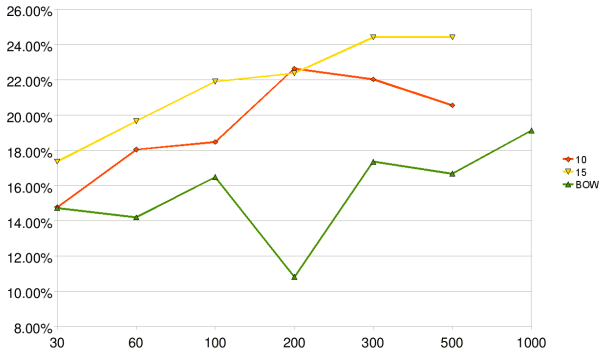


**Figura 4.11:** *People_Marching*

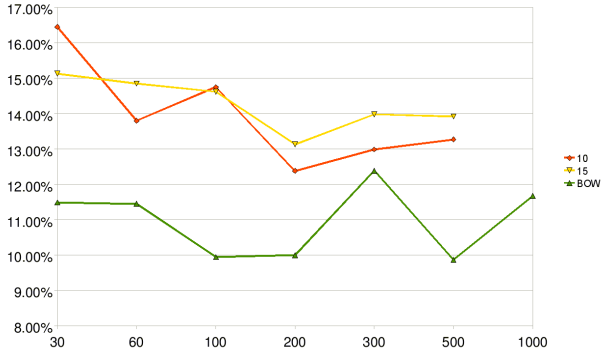


**Figura 4.12:** *Street_Battle*

4.5). Questo andamento può essere spiegato considerando due aspetti: il primo è che la tecnica del Chi-Quadro, aumentando il dizionario, riceve in ingresso istogrammi di dimensione maggiore; il secondo, è che nonostante venga utilizzata la tecnica di pesatura "Soft" (sezione 3.3.1), il numero medio di features all'interno di un fotogramma è dell'ordine di poche migliaia, il che rende gli istogrammi, creati con dizionari elevati, molto sparsi e con picchi localizzati in posizioni molto differenti, aumentando così il valore di confronto restituito dal Chi-Quadro.

Un'ultima osservazione deve essere fatta per l'unica azione, *Airplane_Flying*, i cui risultati ottenuti con il nostro approccio, mostrati in tabella 4.7, sono peggiori di quelli ottenuti con l'approccio classico BoW. Possiamo ipotizzare


**Figura 4.13:** *Nella classe Airplane_Flying possiamo notare che i fotogrammi estratti presentano spesso una staticità di movimento che influenza la capacità di rilevazione di tale concetto dinamico.*

che il motivo di questo risultato sia imputabile al fatto che, se andiamo a vedere tali sequenze video (vedi Figura 4.13), notiamo che c'è una ridotta variazione nell'evoluzione dell'azione: per tracciare un aereo che vola vengono fatte delle riprese da un altro aereo, e in tal caso la scena rimane pressappoco immobile perché il moto relativo è nullo, oppure vengono fatte delle riprese da terra, dove gli aerei risultano molto piccoli e il moto che si rileva è scarso e attribuibile per la maggior parte allo sfondo. Questi aspetti si concretizzano

in una forte componente di descrizione del contenuto del video, ma non del movimento, e quindi l'approccio key-frame based risulta essere meno affetto da rumore nel considerare solo la componente statica.

Eseguita l'analisi utilizzando la metrica dell'*Edit Distance*, abbiamo effettuato dei test con la seconda metrica, cioè quella delle sotto stringhe (SSK). Abbiamo verificato subito che il tempo necessario per ottenere dei risultati sul dataset TRECVID, era troppo oneroso e abbiamo deciso di effettuare l'analisi solo sul dataset calcistico.

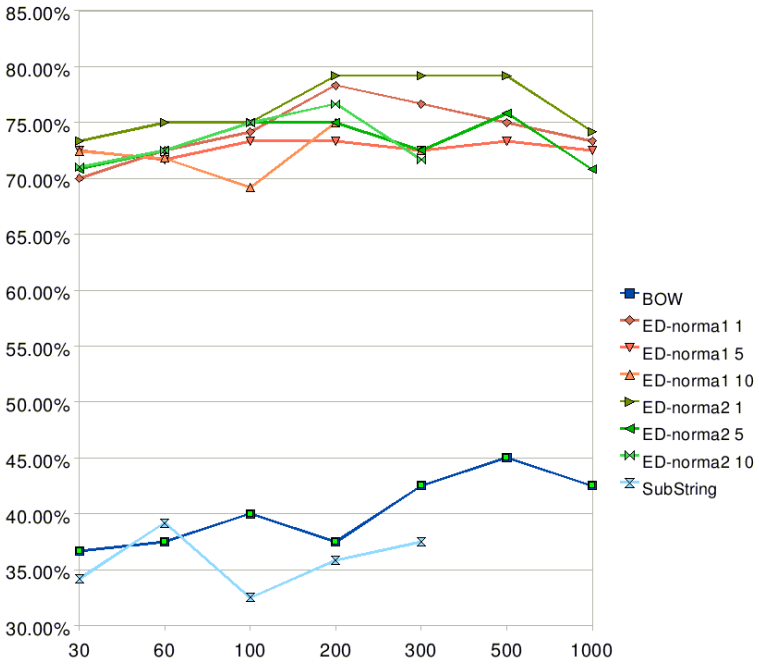


**Figura 4.14:** *Grafico dei risultati ottenuti sul dataset calcistico. I risultati si riferiscono all'accuracy restituita dal classificatore. Sono riportati i test con la metrica ED, analizzata con campionamento ogni 1, 5 e 10 fotogrammi e con le due normalizzazioni indicate con "norma1" (eq. 3.2) e "norma2" (eq. 3.3), e l'unico test eseguito con SSK, analizzato con campionamento ogni 5 fotogrammi e lunghezza della sotto stringa uguale a 3.*

Utilizzando i valori medi dei parametri comuni con l'ED (campionamen-

to ogni 5 fotogrammi, normalizzazione utilizzando l'equazione 3.2, tecnica del Chi-Quadro per il confronto tra caratteri), impostando inizialmente la lunghezza delle sotto stringhe a 3 e facendo variare il fattore di decadimento $\lambda$ in un intervallo piccolo, il tempo di elaborazione necessario per effettuare i primi test, al crescere del dizionario, è diventato troppo oneroso, costringendoci a fermare l'analisi prima del previsto, cioè senza eseguire i test con i dizionari più elevati (500 e 1000). Nel grafico 4.14 possiamo notare che le prestazioni ottenute utilizzando l'approccio con le sotto stringhe risultano paragonabili a quelle ottenute con l'approccio BoW classico e molto inferiori rispetto ai risultati con la metrica dell'ED.

# Capitolo 5

# Conclusioni e sviluppi futuri

In questo lavoro di tesi è stato proposto e realizzato un sistema di riconoscimento di azioni ed eventi presenti in filmati video, analizzando, come caso di studio, sette concetti all'interno del dataset TRECVID 2005.

Tale sistema paragona i video a dei documenti testuali, analizzandoli come fossero una sequenza di parole appartenenti ad uno specifico vocabolario. Questo paragone permette di utilizzare le tecniche applicate nell'ambito testuale per comparare due video e ricavarne un valore di similarità da utilizzare per la classificazione.

La soluzione proposta, consiste nel descrivere un video come una sequenza di istogrammi calcolati secondo il modello *Bag-of-Visual-Words*. Per ottenere questi istogrammi si estraggono da ogni fotogramma i punti SIFT, descrittori locali di punti di interesse, tramite i quali viene anche costruito il dizionario per la loro codifica. Successivamente, viene utilizzata, con vari accorgimenti di implementazione, la metrica Edit Distance, che in base alle modifiche necessarie per trasformare il primo video nel secondo, permette di costruire un kernel specifico per la fase di classificazione.

I risultati mostrano che, rispetto al modello Bag-of-Visual-Words applica-

to ad un approccio key-frame based , con il nostro metodo si ottengono delle prestazioni migliori, ma a discapito della velocità di elaborazione. Anche se l'oscillazione dei risultati in funzione dei parametri non è molto elevata, l'analisi eseguita porta a concludere che ci sia un rapporto di proporzionalità inversa tra la dimensione del dizionario e la lunghezza della frase, cioè del campionamento della sequenza. Utilizzare un dizionario cospicuo introduce troppo rumore nella comparazione di due stringhe di lunghezza elevata, con risultati inferiori a quelli ottenuti campionando i video a distanza maggiore. Al contrario, un dizionario piccolo necessita di sequenze più lunghe per classificare meglio i concetti dinamici.

Eventuali sviluppi futuri, possono riguardare la prima fase di estrazione delle caratteristiche locali da ogni fotogramma, sviluppando un metodo che selezioni, soprattutto durante la creazione del dizionario visuale, solo quei punti a più alto contenuto informativo dell'azione, e che escluda tutti quei punti che aggiungono rumore al processo. Un possibile approccio potrebbe essere quello di prendere in esame solo quelle caratteristiche che possiedono una componente significativa di moto relativo all'interno della sequenza. Un altro sviluppo naturale del presente lavoro consiste nell'ampliamento dei descrittori stessi, estendendoli con informazioni di carattere temporale o sostituendoli con descrittori che inglobino già al loro interno tali informazioni dinamiche.

# Bibliografia


[1] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluating bag-of-visual words reprentations in scene classification," 2007.

[2] S. Lazebnik, C. Schimd, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recogniziing natural scene categories."

[3] K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," *ICCV*, 2005.

[4] J. C. Niebles, H. Wang., and L. F. Fei, "Unsupervised learning ofhuman action categories using spatial-temporal words."

[5] P. Dollàr, V. Rabaud, G. Cottrell, and S. J. Belongie., "Behavior recognition via sparse spatio-temporal features," *In Proc. of ICCV Int.'l Work-shop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VSPETS)*, 2005.

[6] X. Zhou, X. Zhuang, S. Yan, M. Hasengawa-Johnson, and T. S. Huang, "Sif-bag kernel for video event analysis," *MM'08*.

[7] F. Wang, Y.-G. Jiang, and C.-W. Ngo, "Video event detection using motion relativity and visual relatedness," 2008.

[8] D. Xu and S.-F. Chang, "Video event recognition using kernel methods with multilevel temporal alignment," *IEEE Transactions on pattern analysis and machine intelligence*, 2008.

[9] I. Laptev and T. Lindeberg, "Space-time interest points," *ICCV*, 2003.

[10] P. Scovanner, S. A. , and M. Shah, "A 3-dimensional SIFT descriptor and its application to action recognition," *In Proc. of ACM International Conference on Multimedia (MM)*, 2007.

[11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV)*, 2004.

[12] S. F. Wong and R. Cipolla, "Extracting spatiotemporal interest points using global information," *In Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2007.

[13] K. Minkolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, 2004.

[14] L. Ballan, "Riconoscimento automatico di volti utilizzando descrittori locali di caratteristiche facciali," 2006.

[15] L. Seidenari, "Riconoscimento di azioni tramite punti di interesse spazio-temporali," 2008.

[16] Schiele and Vogle, "Semantic modeling of natural scenes for content-based image retrieval," *International Journal of Computer Vision (IJCV)*, 2007.

[17] S. Lazebnik and M. Raginsky, "Supervised learning of quantizer code-books by information loss minimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2008.

[18] J. G. Liu and M. Shah, "Learning human actions via information maximization," *In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[19] F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition," vol. 1, pp. 604 – 60, 2005.

[20] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," *In Proceedings of ICCV*, 2003.

[21] K. Mikolajczyk, B. Leibe, , and B. Schiele, "Local features for object class recognition," *In Proc. of IEEE International Conference on Computer Vision*, 2005.

[22] N. P. Cuntoor and R. Chellappa, "Key frame-based activity representation using antieigenvalues," *ACCV*, 2006.

[23] Y. Ke, R. Sukthankar, and M. Hebert, "Efficient visual event detection using volumetric features," *International Conference on Computer Vision*, 2005.

[24] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," *ICPR*, 2004.

[25] A. Oikonomopoulos, I. Patras, and M. Pantic, "Spatiotemporal saliency for human action recognition," 2005.

[26] Y.-G. Jiang, C.-W. Ngo, and J. Yang, "Towars optimal bag-of-features for object categorization and semantic video retrieval," 2007.

[27] M. Neuhaus and H. Bunke, "Edit distance-based kernel functions for structural pattern classiffication," *Pattern Recognition Society*, 2006.

[28] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text classiffication using string kernels," *Journal of Machine Learning Research 2*, 2002.

[29] J. Shawe-Taylor and N. Cristianini, "Kernel methods for pattern analysis."

[30] B. Hasdonk, "Feature space interpretation of SVMs with indefinite kernels," *IEEE*, 2005.

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

[32] "Trec video retrieval evaluation," *http://www-nlpir.nist.gov/projects/trecvid*.

[33] "DTO challenge workshop on large scale concept ontology for multimedia, "revision of lscom event/activity annotations"," 2006.