UNIVERSITÀ DEGLI STUDI DI FIRENZE
Facoltà di Ingegneria - Dipartimento di Sistemi e Informatica

Tesi di laurea in Ingegneria Informatica

Riconoscimento di azioni tramite punti di interesse spazio-temporali

Candidato
Lorenzo Seidenari

Relatori
Prof. Alberto Del Bimbo

Ing. Marco Bertini

Correlatori
Ing. Lamberto Ballan

Ing. Giuseppe Serra

Anno Accademico 2007/2008

The question of whether Machines Can Think... is about as relevant as the question of whether Submarines Can Swim.

Edsger W. Dijkstra

Ask not what your teammates can do for you. Ask what you can do for your teammates.

Earvin "Magic" Johnson

Indice

Elenco delle figure

Elenco delle tabelle

Sommario

La realizzazione di tecniche di visione computazionale in grado di riconoscere ed interpretare automaticamente comportamenti umani è un argomento che ha ricevuto grande attenzione dalla comunità scientifica. Questo tipo di tecnologia ha applicazioni in ambito di categorizzazione e recupero di materiale multimediale, nella videosorveglianza intelligente e nei sistemi di interazione naturale.

Molti dei recenti e più promettenti risultati presentati nella letteratura scientifica riguardano lo sviluppo di tecniche di descrizione locale dei *pattern* di movimento. L'uso di operatori di estrazione di punti di interesse nel dominio spazio-temporale consente di ricavare questo tipo di descrizione. I dati estratti con queste tecniche, codificati attraverso un dizionario visuale, sono sfruttati per apprendere tramite classificatori statistici modelli per i comportamenti umani. Questa tesi studia ed approfondisce queste tecniche e fornisce i seguenti contributi:

- L'implementazione e sperimentazione di un campionamento denso della scala spaziale e temporale tramite un operatore di estrazione di punti di interesse presentato in letteratura, originariamente sprovvisto di un meccanismo di selezione della scala spazio-temporale.
- La formulazione di due descrittori locali invarianti alla scala che non necessitano di taratura fine dei parametri.
- L'applicazione di una tecnica di generazione del dizionario visuale basata sulla ricerca di mode nella distribuzione dei descrittori.
- La riduzione dell'errore di quantizzazione tramite una modellazione dell'incertezza.

Il suddetto approccio viene validato su due dataset standard di azioni ottenendo risultati in linea o migliori rispetto all'attuale stato dell'arte.

Capitolo 1

Introduzione

Questo Capitolo introduce il problema del riconoscimento di azioni e come è stato trattato nella recente letteratura. Viene perciò introdotta una tassonomia degli approcci esistenti al fine di presentare lo stato dell'arte e dare una collocazione ai contributi di questo lavoro.

I recenti progressi tecnologici hanno portato ad un'incredibile diffusione di dispositivi dotati di telecamera ed al relativo aumento di video prodotti. Si pensi ad esempio alle macchine fotografiche digitali compatte, i palmari ed i telefoni cellulari di nuova generazione. L'uso di telecamere PTZ[1] in contesti di videosorveglianza consente di acquisire immagini e sequenze video di persone ad una risoluzione tale da consentire la loro identificazione; questo livello di dettaglio permette anche una migliore osservazione del loro comportamento. La nascita di siti come YouTube[2] e la sezione video di Google[3] ha contribuito ad aumentare la quantità di materiale video disponibile promuovendone la condivisione da parte degli utenti. Questi archivi digitali sono sfruttati nei contesti più svariati: gruppi musicali ne fanno uso per autopromuoversi, politici ed attivisti per esporre il proprio pensiero e molti utenti vi si appoggiano per realizzare veri e propri video-diari personali.

[1] Telecamere dotate di servomeccanismi in grado di inclinarsi, ruotare e variare la lunghezza focale, da cui l'acronimo Pan Tilt Zoom.

[2] http://www.youtube.com/

[3] http://video.google.com/

I video prodotti con dispositivi personali spesso ritraggono volutamente conoscenti e familiari, in altri casi ritraggono eventi traumatici, atti delittuosi o violenti. Questo tipo di filmati, spesso di scarsa qualità, sono divenuti parte integrante degli attuali notiziari televisivi grazie appunto alla presenza di archivi accessibili dal web. Anche i filmati di videosorveglianza spesso vengono condivisi tramite questi siti web. Si assiste, quindi, ad una prolifica produzione di video da parte di molteplici sorgenti e ad una diffusione massiva di queste produzioni e la conseguente fruizione da parte di altrettanti utenti finali.

Oltre a questo nuovo tipo di comunicazione visuale, sono presenti le usuali realtà in ambito multimediale, la cui recente espansione ha seguito una curva a dir poco esponenziale. Infatti la disponibilità commerciale di dispositivi di memorizzazione digitale ad elevata capienza, in grado di riprodurre, scaricare ed **organizzare** contenuti multimediali ha favorito questo fenomeno.

In questo contesto l'uso di database in grado di indicizzare, ordinare e recuperare dati visuali dal punto di vista semantico, sfruttandone il contenuto, è divenuta una necessità stringente. Sistemi di recupero basati sul contenuto sono oramai disponibili al grande pubblico, anche se il loro stadio di sviluppo è ancora embrionale; si pensi ad esempio alla capacità di Google di individuare immagini contenenti volti [52]; oppure all'applicazione per dispositivi portatili Shazam, in grado di recuperare tracce audio "ascoltandone" la melodia tramite il dispositivo. Un altro esempio di content based information retieval (CBIR) è il motore di ricerca Riya[4]/Like[5], nato come tecnologia di riconoscimento di volti, evoluto oggi in sistema di ricerca di oggetti orientato allo shopping.

Al fine di poter rendere fattibile lo sviluppo, l'utilizzo efficace e la crescita di questo tipo di basi di dati è necessario sviluppare tecniche di annotazione video automatica. Il compito dell'annotazione è di norma demandato ad operatori che lo svolgono in maniera del tutto manuale. Questo oltre a non essere un approccio scalabile al problema può causare errori ed eterogeneità nell'etichettatura dei dati. Per questi motivi, algoritmi e tecniche di an-

[4]http://www.riya.com/
[5]http://www.like.com/

notazione automatica sono un argomento di ricerca attivissimo negli ultimi anni.

I casi d'uso per questo tipo di sistemi sono molteplici, gli utenti finali potrebbero voler organizzare rapidamente i contenuti sui loro dispositivi portatili: ad esempio i filmati dei propri figli che giocano, partite sportive o *podcast* di canali tematici.

Realtà di tipo industriale come aziende di produzione multimediale hanno sicuramente interesse nello sfruttare meccanismi che permettano di riusare tracce video dei loro archivi [20]; durante il montaggio di un servizio di un telegiornale si potrebbe voler ad esempio ricercare sequenze video in cui due politici si stringono la mano, oppure sequenze di scontri a fuoco in zone instabili del pianeta. Un'altra interessante applicazione è la localizzazione di marchi pubblicitari in eventi sportivi [2].

L'importanza dei comportamenti umani

Molti dei video che guardiamo (film, sport, notiziari e documentari) contengono persone, ed in particolare il contenuto del video è spesso definito dal comportamento di quest'ultime. Si può quindi dire senza ombra di dubbio che la semantica di gran parte dei contenuti video spesso è definita da azioni e comportamenti.

Nel campo della sicurezza e della videosorveglianza riconoscere comportamenti umani, singoli o collettivi, diventa un bisogno stringente se si desidera sviluppare sistemi di analisi video intelligenti in grado di guidare l'agente umano nel loro uso. Un sistema di questo tipo potrebbe ad esempio essere in grado di individuare automaticamente situazioni di rischio o evidente pericolo per le persone riprese dalle telecamere a circuito chiuso. Anche in questo ambito, un sistema, in grado di riconoscere ed annotare automaticamente comportamenti umani all'interno di una grande quantità di filmati, è di grande interesse applicativo; si pensi ad esempio alla ricerca di eventi inusuali legati ad attività delittuose [11].

Inoltre lo sviluppo di interfacce avanzate uomo-macchina prevede sempre di più un'interazione gestuale e libera da parte dell'utente; questo può

avvenire in sistemi di tipo touch-screen, ma anche in sistemi più immersivi in cui l'interazione è guidata dalla postura, da movimenti degli arti [8] o anche dalle espressioni facciali [47]. È sicuramente di grande interesse poter riconoscere gesti o comportamenti di persone in ambienti interattivi, sia per motivi di misura del gradimento dell'istallazione sia per ottenere dal sistema una risposta adeguata.

1.1 Riconoscimento di azioni

Diamo innanzitutto una definizione di azione umana: **un'azione umana è rappresentata da tutti quei movimenti o comportamenti che coinvolgono una o più persone ed eventualmente uno o più oggetti.** Esempi di azioni in cui sono coinvolte più persone sono ad esempio il bacio, l'abbraccio e la stretta di mano; azioni in cui si interagisce con oggetti e l'ambiente esterno possono essere rispondere al telefono, fumare o bere. Infine, esempi di azioni singole, in cui la persona agisce liberamente e senza modificare l'ambiente circostante né coinvolgere altre persone sono: correre, camminare etc.

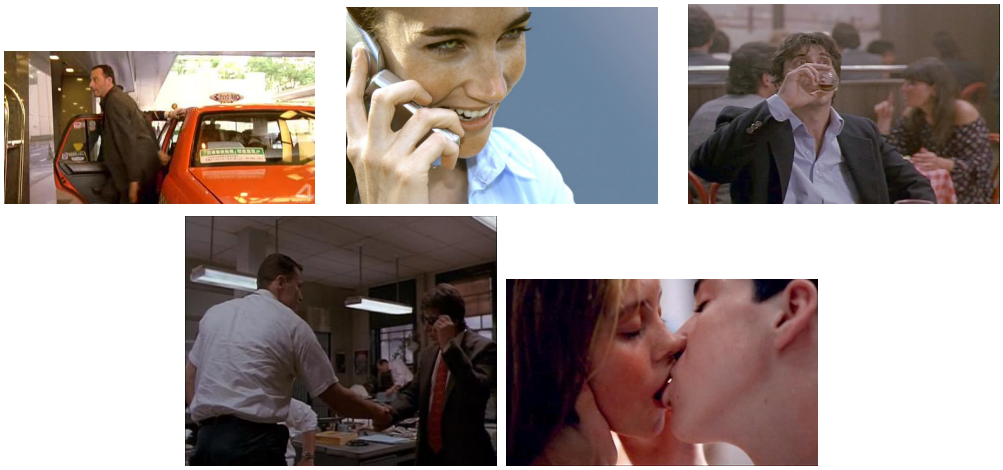


Figura 1.1. Esempi di azioni umane complesse.

Con questa prima serie di esempi ci si può già rendere conto del principale problema: l'elevata variabilità intra-classe delle azioni. Infatti supponiamo di

voler interrogare un sistema di CBIR con una query tipo: “trova tutti i video con persone che bevono”. Tra i video indicizzati dal sistema potrebbero essere presenti filmati sportivi in cui un atleta beve dalla borraccia, una sequenza di un documentario in cui un sommelier degusta un vino dentro una cantina o una qualunque scena di un film girata in un bar, in cui un attore sorseggia un caffè. Questi esempi chiaramente soddisfano l’interrogazione e debbono essere recuperati.

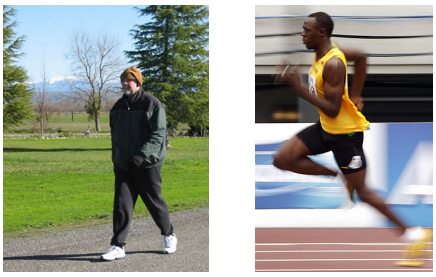


Figura 1.2. Esempi di azioni umane semplici.

Essi contengono la stessa identica azione, tuttavia i contesti in cui è eseguita sono molto variabili; inoltre ogni persona avrà una sua gestualità particolare e differente dalle altre. Altro fattore di difficoltà è l’impossibilità di affidarsi esclusivamente all’aspetto delle azioni. Persone diverse, o la stessa persona, con abiti diversi eseguiranno lo stesso gesto dando origine a tracce video anche radicalmente differenti. A tutti questi problemi inoltre sono da aggiungere quelli che si incontrano classicamente nelle applicazioni di visione computerizzata ovvero: variazioni di illuminazione, di scala e posa dell’oggetto osservato, in questo caso della la persona osservata.

Il problema di interpretare i comportamenti umani ha ricevuto di recente grande attenzione dalla comunità scientifica. Possiamo grossolanamente dividere gli approcci in olistici e basati sulle parti.

1.2 Approcci olistici

In passato l’analisi del movimento umano è stata sempre condotta in maniera olistica, ovvero considerando l’interezza del corpo umano e cercando di

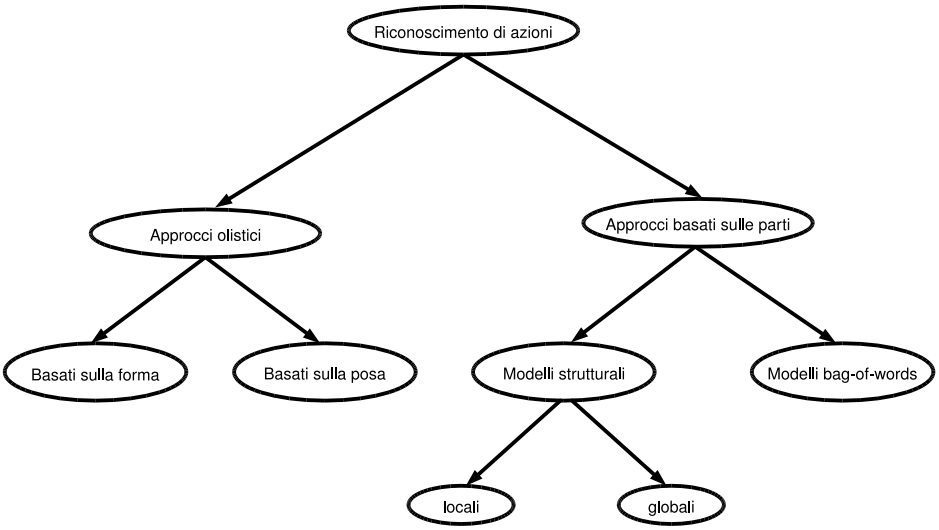


Figura 1.3. Tassonomia degli approcci al riconoscimento di azioni.

ricavarne informazioni sintetiche da caratteristiche come il contorno o la posa. Possiamo quindi ulteriormente suddividere gli approcci olistici in basati sulla posa e basati sulla forma. In Figura 1.3 è mostrata una tassonomia dei principali approcci esistenti in letteratura. Gli approcci basati sulla forma hanno la particolarità di non considerare il corpo umano come un sistema articolato e separato da giunture; cercano piuttosto di inferire informazioni sui comportamenti umani dalla forma della figura umana all'interno del frame analizzato senza fare specifiche assunzioni sulla sua struttura tridimensionale. Efros et al. [13] previa stabilizzazione e segmentazione dell'attore umano creano un modello dell'azione basato sull'optical flow [32] calcolato per ogni due fotogrammi della sequenza. L'optical flow è una misura del moto apparente ed è formalmente un campo di velocità (vedi Sezione 4.2.2). Questo tipo di descrizione del moto è studiata per permettere di riconoscere azioni da filmati a bassa risoluzione, i.e. filmati sportivi, in cui il soggetto non è più alto di 30 pixel.

Gorelick et al. [19] fanno un'analisi di forme tridimensionali derivate dai contorni delle persone e sfruttano la soluzione dell'equazione di Poisson per estrarre caratteristiche come la struttura e l'orientazione della forma. Bobick et al. [5] propongono come modello del moto le *motion history images*,

immagini in scala di grigi create accumulando pixel per pixel valori proporzionali all'ampiezza e alla durata del moto. Lo scopo di questo tipo di rappresentazione è creare un *template* del movimento che sia robusto rispetto a piccole variazioni di vista, rapido da calcolare di modo da permettere anche applicazioni real-time.

Un'estensione più robusta di questo concetto è proposta da Weinland et al. [59], che rinominano il modello in *motion history volume* e utilizzano cinque viste indipendenti dell'attore. Questo tipo di setup, che è più frequente nel caso dell'analisi tridimensionale del moto, permette di aumentare la robustezza ma è ovviamente molto più costoso.

Vi è inoltre un altro filone di ricerca che studia il moto del corpo umano sfruttando modelli tridimensionali; i modelli tridimensionali hanno la funzione di rappresentare la posa della persona, ci si riferisce per questo a queste tecniche come "basate sulla posa" (vedi Figura 1.3). Tipicamente viene scomposto il corpo secondo le sue giunture principali (collo, spalle, gomiti, bacino etc), ricavandone così un modello dettagliato. Raccogliendo i dati tramite sensori o effettuando tracking delle giunture del corpo si cerca di distinguere le azioni in base ai parametri del modello suddetto ricavati da un *fit* sulle traiettorie. Un recente lavoro di Ali et al. [1] sfrutta le traiettorie di mani piedi e bacino per condurre un'analisi del corpo umano modellato come sistema non lineare e caotico. Questo tipo di tecniche per quanto sofisticate hanno un grosso costo dal punto di vista della raccolta della *ground truth* e per l'eventuale applicazione su filmati inediti.

1.3 Approcci basati sulle parti

Tutti gli approcci olistici hanno tipicamente il requisito di un'elaborazione aggiuntiva dei dati come il tracciamento del bersaglio e/o la sua segmentazione. Inoltre una metodologia olistica se non progettata attentamente non consente di gestire il riconoscimento in presenza di occlusioni parziali della persona. A fronte di ciò e grazie al recente successo dei punti di interesse 2D si sono sviluppati molti lavori basati su rappresentazioni locali delle istanze. In questi lavori viene tipicamente applicato un filtro per estrarre delle regioni

spazio-temporali del video e una volta applicata una descrizione robusta delle stesse viene creato un modello "basato sulle parti" che può essere disordinato (bag-of-words) o contenere informazioni strutturali. Inoltre le informazioni strutturali possono essere incorporate localmente nei singoli punti di interesse estratti [30, 39, 43, 56] oppure tramite una struttura globale [26, 30]. 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 [12, 22, 24, 42, 55]. I lavori di Laptev e Dollár [12, 24] hanno avuto il maggior successo e su di essi sono basati diversi lavori successivi. Schuldt et al. [44] sfruttano un classificatore SVM e le *feature* locali proposte da Laptev [24]. Niebles et al. [40] usano una tecnica non supervisionata di apprendimento (pLSA) derivata dal dominio testuale sfruttando le *feature* estratte dal rilevatore proposto da Dollár [12].

Tutti questi approcci sfruttano una fase di quantizzazione dei descrittori estratti (cosiddetto dizionario visuale). I dizionari possono essere generati discretizzando i dati, tramite un'annotazione semantica delle *feature* [53] o guidando il processo di quantizzazione in maniera supervisionata [27, 30]. Il problema di ottenere dizionari visuali efficaci è stato affrontato in passato da Jurie e Triggs [21]. Il loro lavoro mostra come nella categorizzazione di scene il campionamento denso delle *feature* [41], assieme ad una strategia di quantizzazione in grado di codificare adeguatamente tutto lo spazio dei descrittori, fornisca sensibili incrementi di prestazioni.

Un ulteriore problema nell'approccio basato su dizionari, inquadrato recentemente nel lavoro di van Gemert al. [51], è la perdita di informazioni dovuta alla quantizzazione dei descrittori. In questo lavoro viene mostrato come modellando l'incertezza nel processo di discretizzazione si migliorino le prestazioni nella categorizzazione di scene.

1.4 Stato dell'arte

Nell'ambito delle rappresentazioni locali e non strutturate lo stato dell'arte per il riconoscimento di azioni umane è rappresentato dai lavori di Laptev [26] e Schmid [23] (vedi Tabella 5.2). Entrambi sviluppano dei descrittori inno-

vativi per le regioni locali estratte e dimostrano il funzionamento sui dataset standard KTH, Weizmann e HOHA[6].

Vi sono unicamente due lavori che trattano il problema dei dizionari nel dominio delle azioni [30, 39]. Sha et al. [30] confrontano diverse estensioni del modello bag-of-words tradizionale. Propongono cioè rappresentazioni di tipo strutturato sia locali che globali. Inoltre l'uso di un dizionario visuale creato con un algoritmo supervisionato (Mutual Information Maximization) consente loro di ridurre la dimensione dei descrittori delle azioni e contemporaneamente di attestarsi su di una performance allo stato dell'arte. Mikolajczyk e Uemura [39] sfruttano molteplici dizionari ad albero per rappresentare una grande quantità di *feature*.

Per quanto riguarda il problema di moderare la perdita di dati dovuta alla quantizzazione dei descrittori non vi sono lavori nell'ambito del riconoscimento di azioni.

1.5 Obiettivi

L'obiettivo di questo lavoro è lo studio delle descrizioni locali per *feature* spazio-temporali. Esiste una grande quantità di operatori per l'estrazione di punti di interesse come visto nelle Sezioni 1.3 e 1.4, ed altrettanti descrittori di regioni locali; tuttavia molti di questi si sono rivelati inefficaci nei problemi di categorizzazione. La descrizione delle regioni è basata su due principali tipi di misure: optical flow e gradiente. Nonostante quasi tutti i lavori visti alla Sezione 1.3 siano basati su questo tipo di dati, molti riportano risultati discordanti sull'efficacia di una o l'altra misura [12, 26, 40]. Si ritiene quindi necessario sperimentare tecniche in questo ambito alla ricerca di una descrizione locale robusta e il cui uso sia esente dalla ricerca di parametri ottimali su ciascun dataset. Si ritiene infatti che un descrittore come il SIFT [31], ad esempio, trovi nella sua generalità uno dei suoi maggiori punti di forza.

I rilevatori di punti di interesse analizzati in grado di selezionare la scala caratteristica [24, 54] dimostrano scarse prestazioni nella categorizzazione,

[6]Hollywood Human Actions.

mentre l'operatore proposto da Dollár [12] consente elevate prestazioni nel riconoscimento di azioni [30], ma viene utilizzato principalmente a scala singola. Pare quindi interessante studiare la possibilità di un'estensione multiscala di questo operatore al fine di fornire ulteriore robustezza e prestazioni.

La creazione di un dizionario condiviso di parole visuali è un prerequisito necessario alla modellazione di tipo bag-of-words. Questa fase degli algoritmi di riconoscimento viene tipicamente svolta secondo un metodo tradizionale (*k-means*) ed esistono unicamente due lavori, nell'ambito della classificazione di comportamenti umani, in cui i dizionari sono generati con tecniche più evolute [30, 39]. Un altro scopo di questa tesi è quindi studiare l'influenza della strategia di formazione del dizionario sulle prestazioni finali del sistema.

A causa della quantizzazione dei descrittori, vi è un'inevitabile perdita di informazioni; questo problema viene affrontato nell'ambito della categorizzazione di scene, ma non è mai stato considerato nel problema che desideriamo affrontare; si vuole quindi studiare questo tipo di tecniche al fine di migliorare ulteriormente la descrizione delle azioni.

Tutti gli approcci studiati verranno validati su dei dataset di riferimento al fine di fornire una comparazione con l'attuale stato dell'arte.

1.6 Organizzazione della tesi

Il resto della tesi è organizzato come segue: nel Capitolo 2 vengono definiti i punti di interesse locali sia per le immagini che per i video, la loro rilevazione e descrizione; nel Capitolo 3 viene presentato il modello bag-of-words come derivazione dal mondo della classificazione del testo e la sua applicazione ai dati visuali; nel Capitolo 4 vengono presentati i tre principali contributi di questa tesi: due descrittori spazio-temporali, una tecnica di creazione di dizionari efficaci ed un metodo per modellare l'incertezza nell'assegnazione dei descrittori alle parole visuali; il Capitolo 5 contiene i risultati sperimentali sui dataset di riferimento ed il Capitolo 6 enuncia le conclusioni di questo lavoro e propone alcune linee di ricerca a breve e lungo termine sugli argomenti trattati.

Capitolo 2

Punti di interesse spazio-temporali

In questo capitolo viene introdotto il concetto di punto chiave, o di interesse, a partire dalla formulazione in due dimensioni per estendere poi il concetto al caso spazio-temporale. Viene descritto l'attuale stato dell'arte di questa tecnica per quanto riguarda la rilevazione dei punti e la descrizione delle regioni individuate.

2.1 Punti di interesse 2D

L'uso di approcci basati su rappresentazioni locali sparse di oggetti riscuote recentemente gran successo sia nel campo della ricostruzione geometrica della scena, in cui punti distintivi vengono sfruttati per ottenere corrispondenze tra viste diverse, sia nel campo del riconoscimento di oggetti 2D e 3D. Ultimamente questo tipo di tecniche viene sfruttato per la formulazione di modelli atti alla classificazione di immagini.

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 trasfor-

mazioni prospettiche e distorsioni. Infatti se globalmente l'immagine subisce una trasformazione prospettica (ad esempio per il cambio di punto di vista) 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; si veda il Capitolo 3 per i dettagli di questo tipo di modellazione.

Come si vedrà in seguito (Capitoli 3 e 4) e come viene ravvisato da Nowak et al. [41], l'uso di un rilevatore di punti nel caso della classificazione può portare a dei problemi; in questo caso può essere vantaggioso sfruttare una strategia di campionamento denso delle regioni; rimane comunque fondamentale il contributo dei descrittori locali robusti rispetto a rotazioni e distorsioni.

Una rappresentazione locale sparsa di un oggetto viene ottenuta localizzando dei cosiddetti **punti di interesse**. Un punto di interesse avrà le seguenti proprietà:

- Viene localizzato in regioni di spazio, definite da operatori matematici, la cui struttura locale è ben nota.
- La regione identificata nel suo intorno ha un elevato contenuto informativo ed è altamente distintiva.
- Ha una scala caratteristica (in alcuni casi), che consente di ottenere globalmente descrizioni invarianti alla scala per gli oggetti.
- È possibile descrivere le regioni estratte in maniera invariante alle rotazioni, variazioni di illuminazione e distorsioni locali.

Gli algoritmi di estrazioni di punti di interesse sono tipicamente composti da due fasi:

1. Rilevazione di punti distintivi.
2. Descrizione robusta dei punti.

La fase di rilevazione può essere considerata come l'applicazione di un filtro all'immagine $I(x, y)$. Gli algoritmi di rilevazione sono studiati per fornire regioni dell'immagine con buone caratteristiche di invarianza a cambio di punto di vista, scala e rumore. Per questo sono formulati secondo delle euristiche volte a cogliere localmente strutture dell'immagine che si mantengano stabili (siano localmente covarianti) durante le trasformazioni suddette.

La fase di descrizione dei punti ha lo scopo di ottenere una descrizione più robusta possibile allo scopo di ricercare corrispondenze tra immagini diverse contenenti gli stessi oggetti. I descrittori sono tipicamente formulati in modo da essere invarianti alla scala e alla rotazione; in alcuni casi anche a distorsioni locali.

Nell'ambito 2D sono stati sviluppati negli ultimi anni diversi operatori per la localizzazione di punti di interesse ed altrettanti descrittori. Si rimanda al lavoro di Mikolajczyk [38] per una comparazione estensiva.

Valutazione quantitativa

Le prestazioni dei rilevatori di punti di interesse sono valutabili misurando la *ripetibilità* dei punti; ovvero, date due immagini dello stesso oggetto ripreso da punti di vista diversi, si misura la percentuale di regioni localizzate nella medesima posizione. Un'ulteriore tecnica per misurare la bontà del rilevatore è misurare la percentuale di sovrapposizione delle regioni rilevate.

Per quello che riguarda i descrittori una tecnica per misurare le prestazioni consiste nel misurare i valori di

$$recall = \frac{\#corrispondenze\ corrette}{\#corrispondenze} \tag{2.1}$$

e

$$1 - precision = \frac{\#corrispondenze\ errate}{\#corrispondenze\ errate + \#corrispondenze\ corrette}. \tag{2.2}$$

La correttezza delle corrispondenze viene misurata valutando l'intersezione tra le regioni i cui descrittori hanno distanza minore di una data soglia. Il numero di corrispondenze possibili è dato dalla quantità di regioni, rilevate nell'immagine di query, che riproiettate sull'immagine nel database ottengono un errore di sovrapposizione inferiore ad una certa soglia. I valori delle curve di $1 - precision$ e $recall$ sono ricavati variando la soglia sulla distanza tra descrittori.

Si presentano di seguito due diffuse tecniche per rilevazione e descrizione di punti chiave 2D; queste tecniche sono state estese con successo al caso spazio-temporale ed i relativi lavori sono il riferimento per la maggior parte della recente letteratura in questo ambito.

Scale-space

Al fine di poter ottenere rappresentazioni invarianti alla scala è stato necessario definire una rappresentazione multi-scala per un'immagine. Senza scendere nei dettagli matematici di questa formulazione definiamo

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \tag{2.3}$$

come lo *scale-space* di un'immagine, dove con $*$ si rappresenta la convoluzione lungo le direzioni x e y e

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{(x^2+y^2)/2\sigma^2} \tag{2.4}$$

è il kernel gaussiano. Sotto una serie di ragionevoli assunzioni Lindeberg afferma che il kernel gaussiano è l'unico kernel possibile per definire uno *scale-space* [28].

2.1.1 SIFT

L'acronimo SIFT sta per Scale Invariant Feature Transform. Il metodo proposto da Lowe [31] è in grado di localizzare efficientemente punti di interesse, applicando una selezione della scala. I punti SIFT si trovano agli estremi dell'operatore Difference-of-Gaussians(DoG) in 3D.

Rilevamento

L'implementazione è la seguente:

1. Vengono calcolate due "piramidi" di immagini; la prima convolvendo e sottocampionando l'immagine originale con una serie di kernel gaussiani con σ via via maggiori. La seconda calcolando le differenze tra le immagini di ciascun livello della prima.

2. Vengono estratti tutti i punti che sono estremi nello spazio e tra i livelli adiacenti.

I passi 1. e 2. realizzano efficientemente l'operatore DoG:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{2.5}$$

Il rilevatore DoG di fatto è un'approssimazione del Laplaciano di Gaussiane normalizzato:

$$\sigma\nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \tag{2.6}$$

ovvero

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2\nabla^2 G. \tag{2.7}$$

In una comparazione quantitativa di rilevatori di punti di interesse, Mikolajczyk [38] riporta che gli estremi dell'operatore $\sigma^2\nabla^2 G$, detto Laplaciano normalizzato, sono i migliori punti in quanto a stabilità; il rilevatore proposto da Lowe quindi sfrutta queste proprietà aggiungendo un'implementazione efficiente, schematizzata in Figura 2.1.

Descrizione

Il maggiore contributo dell'operatore SIFT è sicuramente il suo descrittore. Il descrittore SIFT è considerato di fatto il miglior descrittore in quanto a robustezza e resistenza a deformazioni locali [36]. Innanzitutto sfrutta il gradiente come misurazione della regione locale rilevata dall'operatore DoG; questo dà robustezza rispetto ai cambi di illuminazione. Per migliorare la robustezza a deformazioni e trasformazioni affini locali, la regione considerata viene suddivisa in 16 sottoregioni e per ognuna di esse viene calcolato

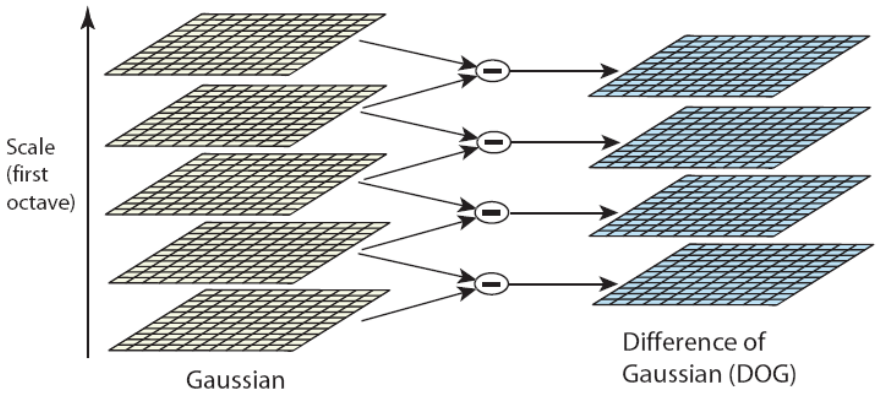


Figura 2.1. Calcolo efficiente dell'operatore Difference of Gaussians(DoG) sfruttando le piramidi di immagini.

un istogramma delle orientazioni del gradiente (vedi Figura 2.2).Per ottenere invarianza alla rotazione viene assegnata a ciascun punto un'orientazione principale. L'invarianza alla rotazione è ottenuta calcolando le suddette orientazioni relativamente all'orientazione principale.

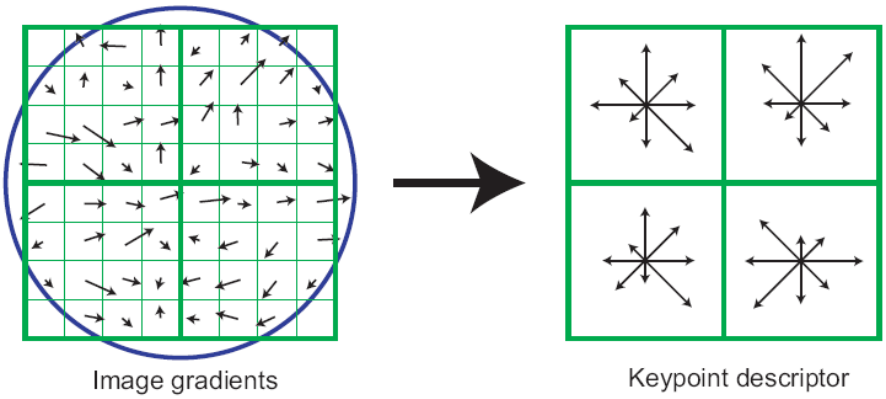


Figura 2.2. Schematizzazione del descrittore SIFT. I campioni del gradiente sono pesati con una finestra gaussiana centrata nel punto rilevato. Ciascuna delle 4 sottoregioni viene descritta tramite un istogramma delle orientazioni.

2.1.2 Harris-Laplace

Rilevamento

L'operatore di Harris è formulato nel seguente modo:

$$H = det(\mu) - k \cdot trace^2(\mu) = \lambda_1 \cdot \lambda_2 - k \cdot (\lambda_1 + \lambda_2)^2 \tag{2.8}$$

Dove μ è la matrice dei momenti secondi calcolata sulle derivate gaussiane dell'immagine

$$\mu = \begin{pmatrix} {L_x}^2 & L_x L_y \\ L_x L_y & {L_y}^2 \end{pmatrix} \tag{2.9}$$

e L_x ed L_y sono appunto le derivate gaussiane calcolate ad una data scala σ_l.

I massimi della 2.8 individuano strutture nell'immagine in cui l'intensità esibisce variazioni ortogonali fra loro, ovvero una struttura ad angolo. Mikolajczyk propone una tecnica di selezione della scala caratteristica basata sul rilevatore di Harris [35]. Il suo approccio consiste nel ricercare (con una procedura analoga a quella del SIFT) i massimi della 2.8 e gli estremi del Laplaciano normalizzato nella scala. In pratica vengono calcolati i punti caratteristici di Harris a varie scale (campionate densamente) e, di questi, vengono mantenuti unicamente quelli che sono anche estremi del Laplaciano rispetto alla scala. Data la selezione della scala caratteristica è possibile rilocalizzare gli stessi punti a scale anche molto differenti come è esemplificato in Figura 2.3.

Descrizione

Il descrittore utilizzato in [35] è il cosidetto *local jet* ovvero una concatenazione delle derivate gaussiane $L_x, L_y, L_{xx}, L_{yy}, \ldots$, nello specifico vengono usate derivate del quarto ordine. Ai descrittori viene applicato uno *steering* per ottenere invarianza alla rotazione [16]. Ovvero considerato un kernel gaussiano (espresso per convenienza con costanti di normalizzazione unitarie)

$$G(x, y) = e^{-(x^2+y^2)} \tag{2.10}$$

e le sua derivata prima nella direzione x

$$G_1^0(x, y) = \frac{\partial G}{\partial x} = -2xe^{-(x^2+y^2)}, \tag{2.11}$$

Figura 2.3. Punti rilevati dall'operatore di Harris esteso con funzionalità di selezione automatica della scala; i punti sono usati per una stima robusta dell'omografia tra le due immagini.

la 2.11 ruotata di $\frac{\pi}{2}$ radianti è

$$G_1^{\frac{\pi}{2}}(x,y) = \frac{\partial G}{\partial y} = -2ye^{-(x^2+y^2)}. \tag{2.12}$$

È quindi possibile esprimere un filtro G_1^θ di orientazione arbitraria come combinazione delle 2.11 e 2.12 :

$$G_1^\theta = \cos(\theta)G_1^0 + \sin(\theta)G_1^{\frac{\pi}{2}}. \tag{2.13}$$

Questa formulazione della derivata della funzione 2.10 può essere sfruttata per calcolare descrittori locali basati sulle derivate invarianti alla rotazione. Infatti calcolare le derivate gaussiane di un'immagine equivale ad applicarle separatamente i filtri definiti dalle 2.11, 2.12. I quali in base alla 2.13 possono essere riorientati calcolando la derivata in una direzione θ arbitraria. La capacità di invarianza alla rotazione è data dallo scegliere θ come la moda dell'istogramma dei gradienti all'interno della regione da descrivere.

Separando le sottosezioni di descrizione e rilevamento si è voluto enfatizzare il fatto che il passo del rilevamento è completamente indipendente da quello della descrizione dei punti. Ad esempio potremmo usare un rilevatore DoG e descrivere le regioni estratte con derivate gaussiane o usare l'estensione multiscala del rilevatore di Harris e descrivere i punti con istogrammi locali delle orientazioni. Qualunque combinazione di rilevatori e descrittori

è possibile; facendo ovviamente le dovute attenzioni (alcuni rilevatori forniscono regioni ellittiche adattate alla struttura della texture sottostante) per ciò che riguarda la normalizzazione della regione da descrivere.

2.2 Punti di interesse 3D

Similmente al caso 2D i punti di interesse sono estratti tramite dei filtri applicati al segnale video. 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 rilevatore.

Negli ultimi quattro anni sono stati realizzati svariati lavori, tipicamente estensioni di operatori già usati con successo nel caso 2D.

2.2.1 Rilevatore di Harris spazio-temporale

Laptev [24] estende l'operatore di Harris per la rilevazione degli angoli al caso spazio-temporale. Viene elegantemente esteso lo *scale-space* spaziale al caso spazio-temporale. I punti di interesse spazio-temporale vengono localizzati nei massimi nello spazio-tempo dell'operatore di Harris esteso e nello spazio delle scale negli estremi del Laplaciano.

Per definire lo *scale-space* spazio-temporale si usa un kernel gaussiano del tipo

$$g(x, y, \sigma, \tau) = \frac{1}{\sqrt{(2\pi)^3\sigma^4\tau^2}}\exp(-(x^2 + y^2)/2\sigma^2 - t^2/2\tau^2);$$

lo *scale-space* spazio-temporale sarà dato da

$$L(x, y, t, \sigma, \tau) = g(x, y, \sigma) * I(x, y, t) \tag{2.14}$$

e dette L_x, L_y, L_t le derivate gaussiane della funzione $I(x, y, t)$ possiamo

definire la matrice dei momenti secondi

$$\mu_{ST} = \begin{pmatrix} {L_x}^2 & L_x L_y & L_x L_t \\ L_x L_y & {L_y}^2 & L_y L_t \\ L_t L_x & L_t L_y & {L_t}^2 \end{pmatrix} \tag{2.15}$$

e la risposta dell'operatore di Harris per angoli spazio-temporali

$$H = det(\mu_{ST}) - k \cdot trace^3(\mu_{ST}) \tag{2.16}$$

Nel lavoro di Mikolajczyk [35], di cui questo operatore nello spazio-tempo è diretta estensione, vengono campionate densamente le scale e sono ricercati gli estremi del Laplaciano in corrispondenza di massimi della 2.8; per motivi di efficienza computazionale in questo caso la rilevazione è organizzata in due passi:

- Vengono calcolati gli angoli spazio-temporali per un certo numero (sparso) di scale.
- Ciascun massimo della 2.16 viene iterativamente ricalcolato per un insieme di scale adiacenti, spostandolo nella direzione che massimizza la norma del Laplaciano.

L'operatore di Harris 2D ha un'elevata risposta in presenza di strutture ad angolo. L'operatore di Laptev presenta alte risposte in presenza di angoli spazio-temporali, ovvero nei punti del volume I(x,y,t) in cui si hanno ampie variazioni di intensità in direzioni ortogonali nello spazio, ma anche 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. L'algoritmo sviluppato da Laptev è in grado di ottenere la scala caratteristica degli eventi rilevati estendendo il concetto di *scale-space* al tempo.

2.2.2 Rilevatore di feature periodiche

Dollár et al. [12] sviluppano un semplice rilevatore di *feature* spazio temporali, partendo dalla considerazione che la dimensione temporale va trattata

differentemente da quella spaziale. Il modello matematico del clip video è identico ai lavori di Laptev [24] ma l'operatore applicato al volume ha elevate risposte a pattern di moto periodici. La funzione di risposta è

$$R = (I * g(\sigma) * h_{ev})^2 + (I * g(\sigma) * h_{od})^2 \tag{2.17}$$

dove h_{ev} e h_{od} sono una coppia di filtri di Gabor in quadratura:

$$h_{od} = -sin(2\pi\omega t) * e^{-t^2/\tau^2} \tag{2.18}$$

$$h_{ev} = -cos(2\pi\omega t) * e^{-t^2/\tau^2} \tag{2.19}$$

e g è un kernel gaussiano. Ponendo $\omega = 4/\tau$ i parametri σ e τ individuano la scala spaziale e temporale del filtro. La funzione $R(x, y, t, \sigma, \tau)$ applica un filtraggio gaussiano nelle direzioni x e y al fine di selezionare la scala spaziale; lungo la direzione temporale invece si applicano dei filtri la cui risposta è massima in caso di pattern oscillatori del segnale. La 2.17 è progettata per dare alte risposte qualora le variazioni dell'intensità luminosa contengano componenti periodiche. La 2.17 risponde ad un numero maggiore di eventi rispetto alla 2.16, come si nota in Figura 2.4 e si può dimostrare sperimentalmente che risponde anche agli angoli spazio-temporali rilevati dall'estensione dell'operatore di Harris al tempo. Inoltre oggetti che si muovono a velocità costante e con moto lineare non generano eventi di interesse. Il rilevatore 2.16, ad esempio, invece si attiva in presenza di angoli spaziali ben definiti soggetti a moto traslazionale; la sovrimpressione dei punteggi durante una partita di calcio è un caso in cui questo fenomeno si verifica.

Da alcuni esperimenti qualitativi preliminari il rilevatore di Dollár è stato valutato come il più promettente nonostante la sua semplicità e l'assenza di un meccanismo di selezione della scala. Prova ne è la sua diffusa adozione in lavori successivi in cui viene modificato il framework di apprendimento a valle del meccanismo di descrizione delle azioni.

Il rilevatore di punti formulato da Laptev [24] nonostante la sua eleganza matematica ha il grosso problema di rilevare un numero esiguo di punti e per di più con un elevato costo computazionale causato della ricerca della scala caratteristica. Il rilevatore di Dollár piuttosto si pone direttamente in un'ottica di estrazione densa di informazioni; questa idea è in linea con i più recenti

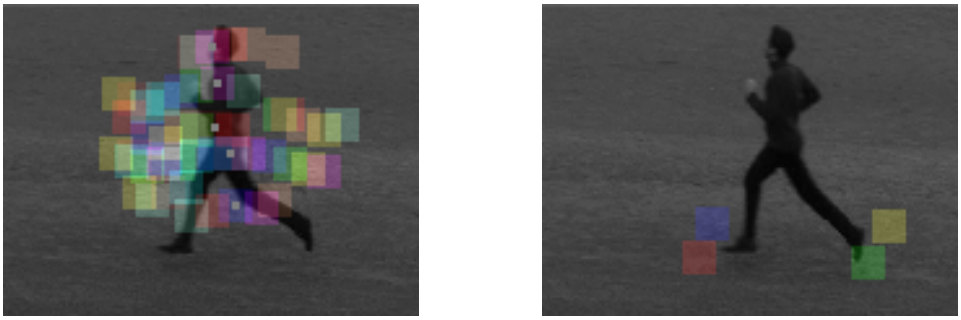


Figura 2.4. Cuboidi estratti dal rilevatore di Dollár e dal rilevatore di Laptev. Gli angoli spazio-temporali sono molto rari e si presentano in regioni in cui una struttura ad angolo inverte il moto bruscamente (i.e. il piede durante la corsa).

lavori di categorizzazione di oggetti e scene, ad esempio Nowak et al. [41] mostrano come un campionamento casuale delle *feature* dia prestazioni superiori nella classificazione rispetto all'uso di punti chiave. Un campionamento denso di spazio, tempo e relative scale sembra dunque la soluzione migliore; a riprova di questo Laptev et al. [26] abbandonano il meccanismo di selezione automatica della scala in favore di un campionamento denso, sempre basato su angoli spazio-temporali. Nel seguito di questo lavoro verrà utilizzato il rilevatore ideato da Dollár.

Altri rilevatori di punti spazio-temporali

Esistono molti altri rilevatori di punti di interesse spazio-temporali. La scelta non è tuttavia ricaduta su questi a causa delle prestazioni meno promettenti; vengono comunque riportati di seguito i principali contributi. Ke et al. [22] estendono il rilevatore di facce Viola&Jones [52] al caso volumetrico, definendo per analogia al lavoro precedente l'*integral video* e le *feature* volumetriche. Il lavoro è orientato ad un riconoscimento real time delle azioni e ottiene una performance inferiore ad esempio al lavoro di Schuld et al. [44] basato sul rilevatore e le *feature* di Laptev. Oikonomopoulos et al. [42] propongono un'estensione del rilevatore di regioni salienti; questo metodo come quello di Laptev ha il problema dell'eccessiva sparsità delle *feature* rilevate.

2.2.3 Descrittori spazio-temporali

Le regioni di spazio-tempo selezionate dal video necessitano come nel caso 2D di essere descritte in maniera robusta. Laptev et al. [24] propongono inizialmente dei descrittori differenziali denominati jet costituiti da $j = (L_x, L_y, L_t, ..., L_{xxx}, L_{yyy}, L_{ttt})$. In un lavoro successivo [25] lo stesso Laptev confronta una serie di descrittori basati su gradienti dell'immagine e optical flow. Vengono calcolate queste due grandezze per ogni *feature* volumetrica, successivamente si applica una delle seguenti tre tecniche per migliorare la robustezza della descrizione:

1. Analisi delle componenti principali (PCA)
2. Istogramma globale
3. Istogramma dipendente dalla posizione o locale.

La PCA è una trasformazione lineare che cerca di proiettare uno spazio vettoriale in uno a dimensionalità minore cercando di preservare le componenti a maggiore varianza. Creare l'istogramma globale dei gradienti spazio-temporali della *feature* volumetrica significa creare una statistica del valore delle derivate indipendente dalla posizione dei campioni misurati. Gli istogrammi locali piuttosto, sono ispirati al descrittore SIFT [31], e sono una statistica dei valori misurati localizzata, su regioni leggermente sovrapposte, all'interno della *feature* volumetrica.

Differenze rispetto al caso 2D

La maggiore differenza è la natura temporale e quindi sequenzialmente ordinata del dato estratto. I descrittori spazio-temporali devono quindi essere progettati per carpire il pattern di movimento locale estratto dal filmato. Nel dominio delle azioni inoltre non è necessario ottenere robustezza alle rotazioni. Se per un oggetto esiste infatti la possibilità di trovarlo in un'immagine ruotato in più modi, di norma le persone non verranno riprese mentre camminano sul soffitto o sulle pareti! Questo tipo di normalizzazione potrebbe addirittura peggiorare la qualità della descrizione e non è molto chiaro quale

sia il significato di cercare un allineamento con un'orientazione del gradiente nel tempo [45].

Il fatto che i punti di interesse spazio-temporali siano usati principalmente in contesti di classificazione fa evolvere i rilevatori ed i descrittori in questa direzione. Ad esempio il rilevatore proposto da Laptev [24] viene modificato dall'autore stesso nell'ottica di ottenere un campionamento denso di scale spaziali e temporali [26], così come il rilevatore di *feature* periodiche [12] ha la caratteristica di estrarre un elevato numero di regioni sovrapposte.

L'unico tipo di misura quantitativa della robustezza e ripetibilità nel dominio spazio-temporale è effettuata da Willems et al. [54]. Vengono usati i test proposti per il dominio 2D da Mikolajczyk [37] per misurare la robustezza dei rilevatori e dei descrittori rispetto a varie trasformazioni (compressione video, traslazione della telecamera, variazione di scala spaziale e temporale).

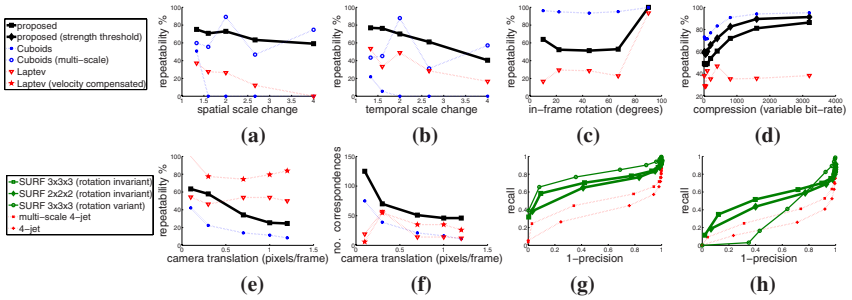


Figura 2.5. Valutazione quantitativa di punti di interesse spazio-temporali. Viene misurata la ripetibilità, il numero di corrispondenze, *1-precision* e *recall*.

Capitolo 3

Bag-of-words

Questo capitolo illustra la rappresentazione bag-of-words per dati visuali. Il modello viene mostrato come estensione della tecnica relativa ai documenti di testo introducendo l'astrazione di parola e dizionario visuali. Sono presentati inoltre gli algoritmi atti a creare il suddetto dizionario ed il classificatore utilizzato in fase di riconoscimento.

3.1 Categorizzazione del testo

Il problema della classificazione dal punto di vista dell'apprendimento automatico è formalizzato nel seguente modo [46]:

Definizione 3.1.1 *Dato un insieme di documenti* $D : \{d_1, d_2, \ldots d_{|D|}\}$ *e l'insieme delle loro categorie identificate con le etichette* $C : \{c_1, c_2, \ldots c_{|C|}\}$ *si vuole approssimare la funzione* $\Phi : D \times C \rightarrow \{T, F\}$ *per mezzo di una funzione* $\hat{\Phi} : D \times C \rightarrow \{T, F\}$ *in modo che* $\hat{\Phi}$ *e* Φ *"coincidano il più possibile".*

Si cercherà quindi di formulare un algoritmo per ottenere la funzione approssimata $\hat{\Phi}$. L'approssimazione deriverà ovviamente dall'impossibilità di osservare tutti i documenti di ogni categoria stabilita. Questo approccio nella pratica ci consente quindi di osservare un numero finito di esempi e successivamente di utilizzare il modello appreso per classificare istanze non osservate.
Tipicamente per verificare la generalità dell'algoritmo di apprendimento il

dataset viene diviso in un *training/validation set* ed un *test set*. La prima frazione di dati viene utilizzata per l'apprendimento e la ricerca dei migliori parametri del modello. La rimanente si utilizza per verificare le effettive prestazioni dell'algoritmo.

Durante la fase di addestramento si cerca di minimizzare la differenza tra Φ e $\hat{\Phi}$; l'algoritmo di apprendimento quindi sfrutterà la conoscenza a priori data dalle etichette presenti sui documenti nell'insieme di *training* al fine di ottenere un modello generale per ciascuna classe. Al termine del processo di apprendimento sarà possibile richiedere al classificatore testé appreso di predire la categoria di un documento sconosciuto.

Qualora il modello da apprendere contenesse dei parametri liberi, si utilizzerà una parte dei dati disponibili per l'addestramento al fine di validare il modello ed ottenere i migliori parametri possibili. Di norma viene utilizzata una procedura di cross-validazione al fine di evitare un iperadattamento del modello ai dati.

In una procedura di cross-validazione viene suddiviso l'insieme dei dati disponibili in k sottoinsiemi; viene poi addestrato il modello su $k-1$ insiemi e misurate le prestazioni sull'insieme rimanente. Per ogni n-upla di parametri analizzati viene ripetuto questo procedimento k volte. I parametri liberi possono essere fatti variare secondo una logica di ricerca del minimo errore di classificazione oppure, come più comunemente viene fatto valutandoli in una griglia uniforme di valori prestabiliti.

Supponendo di avere un algoritmo che ci permette di stimare la Φ dobbiamo dare una rappresentazione conveniente dei documenti per il suddetto algoritmo. Uno degli approcci più diffusi deriva dall'*information retrival* (IR) e rappresenta ciascun documento con un vettore le cui componenti sono indicizzate dalle parole presenti nel *corpus*[1]. Ogni entrata nel vettore è quindi data da un peso dipendente dal numero di occorrenze della parola all'interno del documento. Sia il modo di calcolare il peso che il concetto di parola sono soggetti a vari tipi di modellazione; ad esempio il peso può esser moltiplicato per il logaritmo dell'inverso della frequenza relativa della parola in

[1]Questo termine specifica un insieme di documenti ed è tipicamente usato nel gergo dell'IR.

tutti i documenti. Il concetto di parola può essere ridefinito applicando un algoritmo di *stemming* per sostituire a ciascuna parola la sua radice, facendo così confluire nomi, verbi ed aggettivi nello stesso termine (es. combattere, combattente, combattivo → COMBATT). Inoltre è tipico della TC applicare una rimozione delle cosidette *stop-words* ovvero le parole più frequenti presenti in una lingua; saranno tipicamente parole come: 'il', 'la', 'per', 'di', 'da' etc... Esistono casi particolari in cui l'eliminazione delle parole frequenti non porta a benefici: ad esempio, se stiamo classificando per argomento delle pagine web, rimuovere parole come 'io', 'mi', 'mia' o 'miei' potrebbe eliminare un'importante informazione nell'individuare le pagine personali dove inevitabilmente i riferimenti e l'uso della prima persona saranno molto maggiori rispetto ad altre categorie. Un'altra tecnica che viene spesso utilizzata in questo ambito è la *feature selection* [58], che ha lo scopo di individuare quali dati hanno un elevato contenuto informativo. Viene normalmente associato ad ogni termine un punteggio in base ad una statistica relativa al termine e alle classi del problema di classificazione; alcuni esempi sono la statistica chi-quadro: $\chi^2(t, c)$, o la mutua informazione o infogain:

$$MI(t, c) = \sum_{t \in \{0,1\}} \sum_{c \in \{0,1\}} P(t, c) \log \frac{P(t, c)}{P(t)P(c)}; \tag{3.1}$$

per entrambi questi test statistici un elevato valore significa una dipendenza tra le due variabili (termine e categoria), per selezionare le migliori si usa una misura media su tutte le classi presenti, rispettivamente:

$$\chi^2_{avg}(t) = \frac{1}{|C|} \sum_{i=1}^{|C|} \chi^2(t, c_i) \tag{3.2}$$

e

$$MI_{avg}(t) = \frac{1}{|C|} \sum_{i=1}^{|C|} MI(t, c_i). \tag{3.3}$$

Stabilito un dizionario D di parole, eventualmente semplificato tramite le tecniche suddette, ogni documento viene rappresentato con un vettore di $|D|$ elementi, ciascuno dei quali indica il peso del termine in quel documento.

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); ci si riferisce a questo modello come **bag-of-words** ovvero un *insieme non ordinato di termini*.

Data questa rappresentazione la similarità tra due documenti di testo è immediatamente misurabile tramite il coseno dell'angolo tra i due vettori. Siano quindi d_i,d_j due documenti e v_i,v_j i rispettivi vettori (BoW) possiamo definire una misura di similarità come $sim(d_i, d_j) = \langle v_i, v_j \rangle = \sum_{k=1}^{|D|} v_i^k v_j^k$. Il prodotto scalare quindi avrà valore massimo se i due vettori sono paralleli. In questo caso, ciascun documento conterrà la medesima percentuale di ciascun termine. I documenti non dovranno essere necessariamente identici, in quanto l'ordine delle parole non è preso in considerazione nel confronto.

Questo tipo di rappresentazione consente inoltre di utilizzare una vasta gamma di classificatori; per il testo sono usati principalmente: k-nearest neighbour (k-NN), naïve Bayes e support vector machines (SVM).

3.2 Dizionari visuali

Sivic e Zisserman [49] estendono per la prima volta il modello BoW al campo multimediale. Il loro lavoro vuole fondere le tecniche di IR nate e sviluppatesi per il testo con i metodi di descrizione locale di immagini, recentemente diventati di successo. A questo scopo definiscono il concetto di dizionario visuale.

Questa modellazione del dato visuale mira a cogliere l'aspetto dei contenuti tramite una rappresentazione sparsa e non ordinata. Sivic e Zisserman stabiliscono un'analogia tra i punti chiave (ad es. SIFT [31], MSER [33]) e parole testuali. Il problema fondamentale sta nell'ottenere un vocabolario finito a partire dalle *feature* locali estratte tramite dei rilevatori analoghi a quelli descritti nel Capitolo 2.

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 e la TC. Infatti se rappresentassimo le immagini o le sequenze video come insiemi di punti chiave l'unico classificatore utilizzabile sarebbe un classificatore *nearest neighbour*. Ovvero si utilizzerebbe la clas-

se dell'immagine che ottiene il maggior numero di corrispondenze tra punti, come predizione della classe per l'immagine in esame.

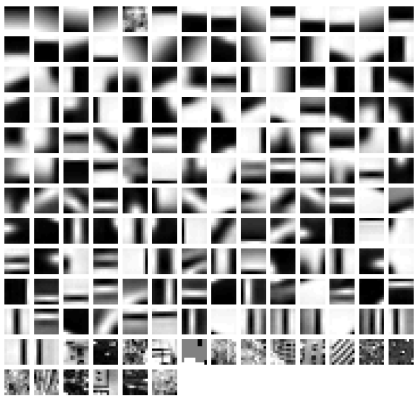


Figura 3.1. Dizionario visuale: regioni 2D rappresentative(parole visuali) per ciascun cluster di descrittori.

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 (vedi Sezione 4.3), i descrittori dei punti di interesse vengono aggregati in modo da rappresentare 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 che esiste un dizionario condiviso e 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. Diversamente dalle parole, il dato percettivo non ha natura discreta; è quindi necessario ricondursi a dei prototipi al fine di poter trattare le immagini o i filmati come un insieme di simboli. 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.

La dimensione del dizionario chiaramente dipende dai dati in analisi, ma è anche un parametro che possiamo controllare dipendentemente dal tipo di algoritmo utilizzato per la sua generazione. La dimensione del dizionario si può quindi variare modificando i parametri dell'algoritmo di discretizzazione, oppure si possono scegliere, tramite tecniche di selezione delle *feature*, le migliori parole per ogni classe.

3.3 Spazio dei descrittori e quantizzazione

Dalla definizione data di parola visuale possiamo subito pensare ad una tecnica banale per ottenere un dizionario: quantizzare lo spazio delle *feature* discretizzando quindi ciascuna dimensione del descrittore. Se applicato in maniera naïf questo approccio dà luogo a dizionari di dimensione ingestibile: ad esempio se si prende il SIFT nella sua formulazione originale solo quantizzare ciascuna componente su due valori dà luogo a 2^{128} valori discreti possibili (circa 10^{38}; 4 suddivisioni darebbero luogo a 10^{77} valori); questo problema è risolto da Tuytelaars e Schmid [50] sfruttando una tabella hash.

Per ottenere la quantizzazione in spazi ad elevata dimensionalità è inoltre possibile utilizzare una tecnica di *vector quantization* (VQ); la VQ prevede la suddivisione dello spazio delle *feature* in un numero prestabilito, o determinato adattivamente, di regioni e l'associazione (quantizzazione) di ciascun descrittore ad una delle suddette regioni. In particolare la tecnica di VQ consiste, dato un insieme di vettori X, nel ricercare tra questi un certo numero di prototipi e di assegnare ognuno dei vettori $X_i \in X$ a quello che meglio lo rappresenta secondo una metrica stabilita. La prima fase viene svolta classicamente con tecniche di clustering; si cerca quindi una partizione dello spazio che consenta di rappresentare tutti i vettori presenti in X compiendo il minimo errore possibile di quantizzazione. Così facendo è possibile discretizzare uno spazio continuo ad elevata dimensionalità, rappresentandolo con un numero finito (anche molto basso) di simboli.

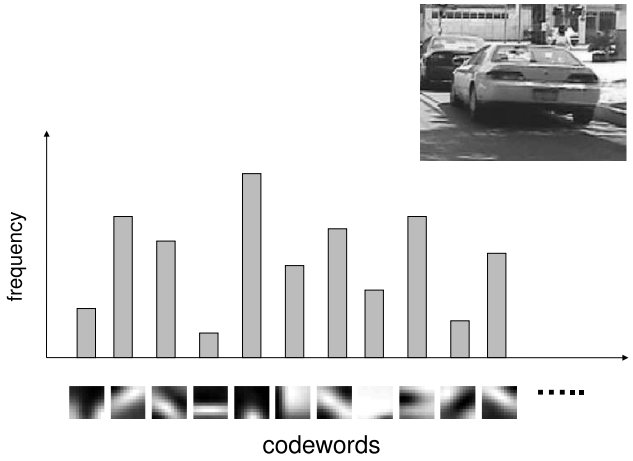


Figura 3.2. Istogramma delle parole visuali per un'immagine contenente una macchina.

K-means

Uno dei più diffusi algoritmi di clustering è *k-means*. Ne diamo di seguito una definizione formale. Dati m vettori in $\mathbb{R}^N$, definiamo $X = \{X_1, X_2, \ldots, X_m\}$, come insieme degli oggetti. Ricordiamo che si definisce partizione degli oggetti il gruppo di insiemi $P = \{P_1, P_2, \ldots, P_k\}$, che soddisfano le seguenti proprietà:

- $\bigcup_1^K P_i = X$: tutti gli oggetti devono appartenere ad almeno un cluster;
- $\bigcap_1^K P_i = \varnothing$: ogni oggetto può appartenere ad un solo cluster;
- $\forall i \in 1..K,\ \varnothing \subset P_i \subset X$: ciascun cluster deve contenere almeno un oggetto, e nessun cluster può contenerli tutti.

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 $C = \{C_1, C_2, \ldots, C_K\}$ l'insieme dei K centroidi. A questo punto definiamo la funzione obiettivo come:

$$V(U, C) = \sum_{i=1}^{K} \sum_{X_j \in P_i} ||X_j - C_i||^2 \tag{3.4}$$

e di questa calcoliamo il minimo seguendo la procedura iterativa in Figura 3.3

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

Figura 3.3. Algoritmo *k-means*.

La più popolare implementazione di questo algoritmo è il cosidetto 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. [34] utilizzano un algoritmo agglomerativo per ovviare all'inizializzazione casuale dell'algoritmo *k-means*. Infatti 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.

Codifica dei descrittori

Di fatto un dizionario visuale è rappresentato da due componenti:

- Un algoritmo per la formazione delle parole visuali.
- Una regola di quantizzazione per i descrittori.

La regola di quantizzazione determina come un oggetto non presente nell'insieme usato per creare il dizionario viene associato alla relativa parola.

Questo serve sia per le istanze del *test set* sia per le istanze del *training set* che non erano state usate per la creazione del dizionario. La scelta che viene fatta tipicamente è di associare ciascun punto al cluster più vicino. Questa scelta ha diversi problemi così come l'algoritmo *k-means*. Uno dei principali contributi di questo lavoro di tesi (vedi Sezione 4.3) è l'implementazione di un algoritmo di clustering più efficace nella localizzazione dei centri e nella codifica dello spazio dei descrittori spazio-temporali.

3.3.1 Differenze tra dominio testuale e dominio visuale

La principale differenza tra i due domini è ovviamente la necessità di un passo di quantizzazione per generare un dizionario discreto. Volendo si può vedere un parallelo tra la procedura di *stemming* e la procedura di quantizzazione delle regioni visuali. Tuttavia lo *stemming* non è un prerequisito necessario alla modellazione bag-of-words dei testi, mentre la definizione di un dizionario discreto per i dati visuali lo è. Un'altra differenza è l'assenza di *stop-words* visuali. Yang et al. [57] riportano un esperimento in cui rimuovere le parole più frequenti porta ad una diminuzione costante nelle prestazioni di classificazione su due dataset di riferimento: PASCAL[2] e TRECVID[3]. Nei nostri esperimenti inoltre abbiamo ravvisato l'importanza delle parole più frequenti; infatti in Figura 5.8 vediamo come anche con solo le prime 100 parole più frequenti possiamo ottenere una accuratezza dell'82%. Pare quindi errato eliminare le parole visuali più frequenti.

3.4 Classificatori SVM

Sono stati fatti esperimenti preliminari con classificatori k-NN e SVM. Il classificatore k-NN, così come riportato in letteratura [12, 55], ha prestazioni inferiori rispetto alle SVM; vengono quindi presentati risultati ottenuti esclusivamente con questo tipo di classificatore. Una support vector machine è un classificatore ad ampio margine lineare. Dato un insieme di l istanze da

[2]http://pascallin.ecs.soton.ac.uk/challenges/VOC/

[3]http://www-nlpir.nist.gov/projects/trecvid/

classificare x_i e le rispettive etichette $y_i \in \{-1, 1\}$ con $i = 1..l$, per ottenere l'iperpiano ottimale occorre risolvere il seguente problema di ottimizzazione:

$$\begin{aligned} &\min_{w,b,\xi} && \tfrac{1}{2}w^T w + C\sum_{i=0}^{l}\xi_i. \\ &\text{con il vincolo} && \\ & && y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i, \\ & && \xi > 0. \end{aligned} \tag{3.5}$$

In questa formulazione le istanze x_i sono mappate in uno spazio a più elevata dimensionalità (potenzialmente infinita) tramite la funzione ϕ. La mappatura non deve essere esplicita; riformulando infatti il problema di minimizzazione come il duale del 3.5 le istanze x_i appaiono solo in prodotti scalari (nello spazio delle *feature*):

$$\begin{aligned} &\min_{\alpha} && \tfrac{1}{2}\alpha^T Q\alpha - e^T\alpha \\ &\text{con il vincolo} && \\ & && y^T\alpha = 0 \\ & && 0 \leq \alpha_i \leq C, \qquad i = 1, \ldots, l, \end{aligned} \tag{3.6}$$

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 3.6 si può ottenere una predizione della classe dell'istanza x tramite la funzione

$$y = \operatorname{sgn}\left(\sum_{i=1}^{l} y_i\alpha_i K(x_i, x) + b\right). \tag{3.7}$$

Al termine dell'ottimizzazione un sottoinsieme dei dati potrebbe avere $\alpha_i = 0$ e quindi non contribuire nella 3.7; i restanti vettori sono detti **vettori di supporto** e nel caso $0 < \alpha < C$ si trovano esattamente sul margine: nelle Figure 3.4 e 3.5 appaiono cerchiati. Per cui il modello memorizzato è costituito unicamente dagli α_i e gli x_i con $i \in \mathcal{SV}$, dove $\mathcal{SV}$ 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.

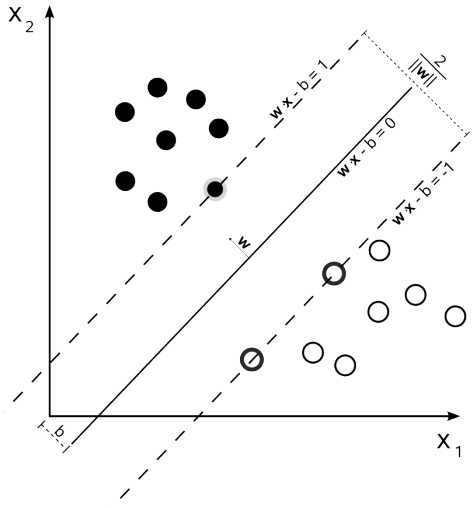


Figura 3.4. Iperpiano ottimo per un insieme linearmente separabile in $\mathbb{R}^2$.

Il secondo termine della funzione obiettivo del problema 3.5 consente, tramite un peso C di ottenere soluzioni al problema 3.5 anche in presenza di insiemi di dati non separabili. Se si usano kernel come 3.9 o 3.10 avremo due parametri liberi nel modello: γ e C. Per determinarne i valori ottimali viene tipicamene effettuata una procedura di cross-validazione sul *training set* variando i parametri del modello su di una griglia logaritmica (i.e. $C = 2^{-5}, 2^{-4} \dots 2^{15}, \gamma = 2^{-15}, 2^{-14} \dots 2^{6}$). 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:

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle ; \tag{3.8}$$

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

Definizione 3.4.1 *Sia $\mathcal{X}$ un insieme. Una funzione simmetrica $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ è un kernel definito positivo su $\mathcal{X}$ se $\forall\ n \in \mathbb{Z}^+, x_1 \dots x_n \in \mathcal{X}$ e $c_1 \dots c_n$ vale $\sum_i^n \sum_i^n c_i c_j K(x_i, x_j) > 0$.*

Abbiamo effettuato esperimenti sia con il kernel radiale (RBF):

$$K(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel^2), \tag{3.9}$$

sia con il kernel χ^2 :

$$K(x_i, x_j) = \exp(-\gamma \chi^2(x_i, x_j)) \tag{3.10}$$

dove

$$\chi^2(x_i, x_j) = \sum_{k=1}^{m} \frac{(x_k^i - x_k^j)^2}{x_k^i + x_k^j}. \tag{3.11}$$

Il kernel χ^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]. In linea

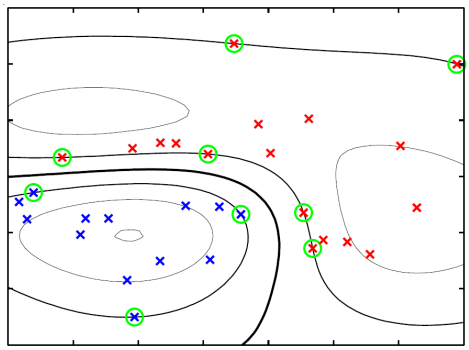


Figura 3.5. Istanze non separabili in $\mathbb{R}^2$ ed iperpiano a massimo margine appreso grazie al kernel RBF.

con i risultati presenti in letteratura per la categorizzazione di scene, oggetti e texture, il kernel χ^2 ha dato i migliori risultati e viene perciò usato negli esperimenti di Capitolo 5.

I vantaggi di questo tipo di classificatori sono principalmente dati dalla teoria matematica con cui sono costruiti; si rimanda al testo di Cristianini [10] per una trattazione completa. Il fatto che il problema 3.6 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 [18]; infatti vista la 3.6 è sufficiente formulare una funzione che soddisfi la proprietà definita nella 3.4.1 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 *feature* 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 (χ^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 [7].

Capitolo 4

Descrittori spazio-temporali e dizionari efficaci

In questo capitolo viene presentato in dettaglio l'approccio implementato e le problematiche che affronta rispetto ai precedenti lavori. Vengono illustrati i descrittori locali proposti e l'applicazione di algoritmi di aggregazione basati sul raggio alla creazione del dizionario visuale.

In questa tesi si propone un approccio ad elevata modularità per il problema della categorizzazione delle azioni umane.

L'algoritmo implementato in fase di addestramento estrae per ciascun video dei cuboidi spazio-temporali, in corrispondenza dei massimi della funzione definita in 2.17. Per ciascun punto viene creata una descrizione robusta e invariante alla scala. I descrittori ottenuti da un sottoinsieme dell'insieme di addestramento sono utilizzati per creare un dizionario visuale. Ogni clip viene descritto come l'istogramma delle parole visuali associate a ciascun descrittore dei punti estratti dal video. Gli istogrammi normalizzati vengono poi usati per addestrare un classificatore SVM.

In fase di test, l'algoritmo agisce analogamente, sfrutta però il dizionario, creato in fase di addestramento, per associare i punti del filmato di query alle parole visuali. Successivamente, tramite il modello appreso, ottiene in maniera molto efficiente una predizione sulla classe di appartenenza.

I contributi di questa tesi sono:

- L'estensione del rilevatore di punti proposto da Dollár [12].
- Due descrittori per i punti di interesse spazio-temporali.
- Il miglioramento dell'algoritmo per la formazione del dizionario.

Come possiamo vedere dalla Figura 4.1 ciascun passo è modificabile ed indipendente dall'altro.

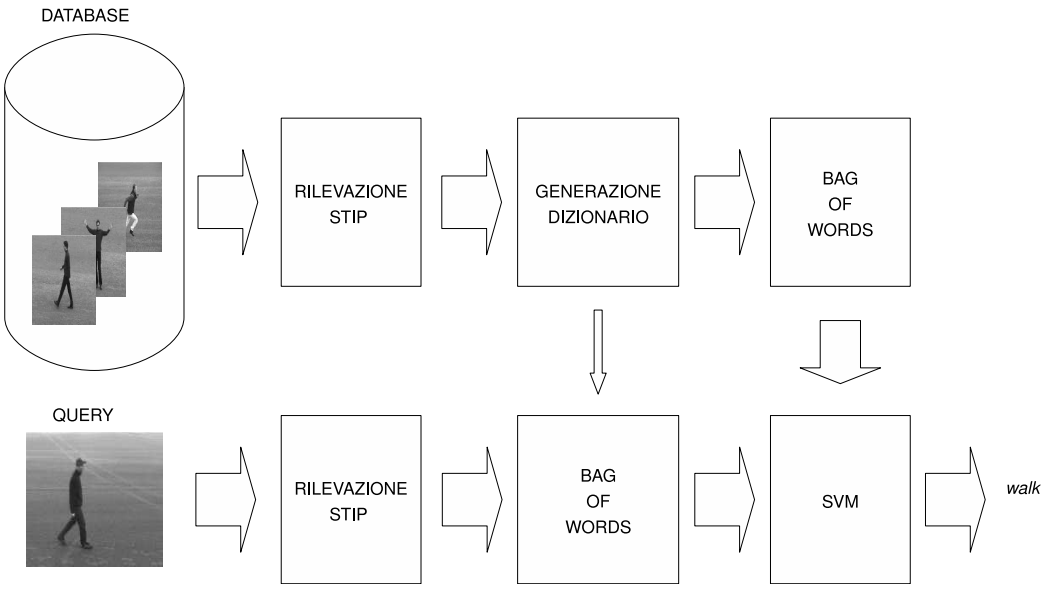
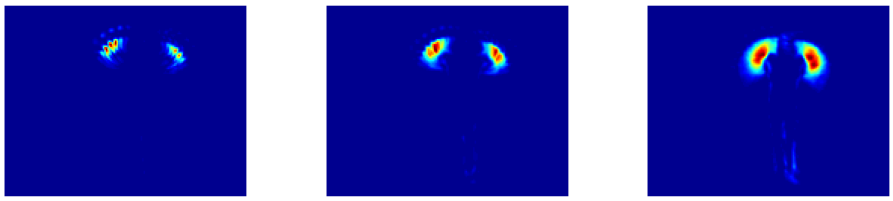


Figura 4.1. Schema a blocchi del framework implementato.

4.1 Estensione multiscala del rilevatore di punti

Il rilevatore di punti di interesse di Dollár è stato valutato come il più promettente nonostante la sua semplicità e l'assenza di un meccanismo di selezione della scala. Prova ne è la sua diffusa adozione in lavori successivi [30, 40, 43] in cui viene modificato il framework di apprendimento a valle del meccanismo di descrizione delle azioni. Per ovviare alla limitazione della singola scala

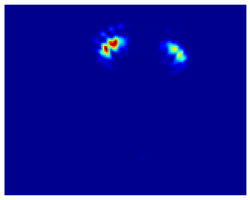
spaziale e temporale la fase di estrazione dei punti di interesse è stata estesa effettuando filtraggi multipli a scale temporali e spaziali via via maggiori. Usare più scale permette sia di riconoscere la stessa azione eseguita ad una distanza diversa (scala spaziale) o ad una velocità diversa (scala temporale); ma anche supposto che non vi sia variazione di scala nei dati osservati, utilizzare il rilevatore a più scale dà una rappresentazione più ricca delle istanze, incrementando di fatto la quantità di *feature* estratte per ogni clip contenente un'azione (vedi Figura 2.4). In questo lavoro abbiamo usato $\sigma = 2, 4$ e $\tau = 2, 4$.



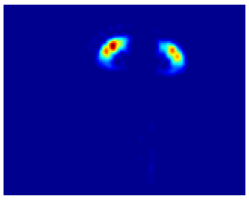
(a) $\sigma = 1, \tau = 1.4$.

(b) $\sigma = 1, \tau = 2$.

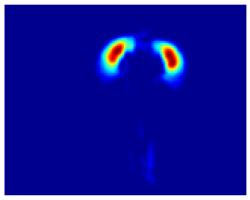
(c) $\sigma = 1, \tau = 4$.



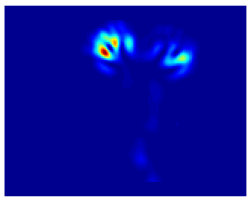
(d) $\sigma = 2, \tau = 1.4$.



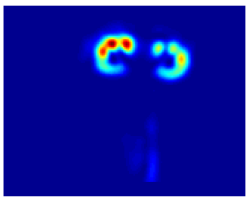
(e) $\sigma = 2, \tau = 2$.



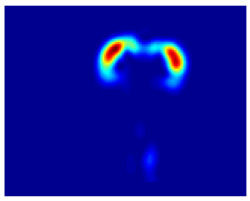
(f) $\sigma = 2, \tau = 4$.



(g) $\sigma = 4, \tau = 1.4$.



(h) $\sigma = 4, \tau = 2$.



(i) $\sigma = 4, \tau = 4$.

Figura 4.2. Risposte del filtro 2.17, applicato al filmato di Figura 4.3 a varie scale spaziali (σ) e temporali (τ). In base alle combinazioni delle scale i massimi locali individuano eventi diversi.

Figura 4.3. Tre frame di una sequenza video. In figura 4.2 sono mostrate le rispettive risposte del filtro 2.17.

4.2 Descrittori locali

4.2.1 Descrittore basato sul gradiente

La robustezza di una descrizione basata sull'orientazione del gradiente in due dimensioni è il principale motivo di successo del descrittore SIFT [31]; 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 ogni punto una o più orientazioni ed utilizzarle come sistema di riferimento ottenendo così invarianza alle rotazioni. L'uso di istogrammi locali piuttosto fornisce alta ripetibilità della localizzazione dei punti tra immagini dello stesso oggetto prese da diversi punti di vista o in pose differenti.

Cerchiamo quindi di trasportare questi vantaggi dal dominio 2D al dominio spazio-temporale. Si vuole quindi una descrizione della regione estratta attorno al punto di interesse basata sul gradiente. Considerato che stiamo descrivendo un dato variabile nel tempo, utilizzare il solo gradiente nello spazio pare inadeguato; è quindi sensato pensare di calcolare il gradiente della funzione $I(x, y, t) : \mathbb{R}^3 \Rightarrow \mathbb{R}$, di modo da **codificare la variazione dell'aspetto locale nel tempo**. Consideriamo quindi il gradiente in coordinate

sferiche:

$$M_{3D} = \sqrt{G_x^2 + G_y^2 + G_t^2}, \tag{4.1}$$

$$\theta = \tan^{-1}(G_t/\sqrt{G_x^2 + G_y^2}), \tag{4.2}$$

$$\phi = \tan^{-1}(G_y/G_x). \tag{4.3}$$

Il valore del gradiente G_x, G_y, G_t viene approssimato usando le differenze finite, in particolare

$$G_x(x,y,t) = \frac{dI(x,y,t)}{dx} = L(x+1,y,t) - L(x-1,y,t), \tag{4.4}$$

$$G_y(x,y,t) = \frac{dI(x,y,t)}{dy} = L(x,y+1,t) - L(x,y-1,t), \tag{4.5}$$

$$G_t(x,y,t) = \frac{dI(x,y,t)}{dt} = L(x,y,t+1) - L(x,y,t-1). \tag{4.6}$$

Dove $L(x,y,t)$ è ottenuta dopo aver applicato un filtro gaussiano alla funzione $I(x,y,t)$, ovvero

$$L(x,y,t) = I(x,y,t) * g(\sigma,\tau) \tag{4.7}$$

e

$$g(\cdot,\sigma,\tau) = \frac{1}{\sqrt{(2\pi)^3\sigma^4\tau^2}}\exp(-\frac{(x^2+y^2)}{2\sigma^2} - \frac{t^2}{2\tau^2}) \tag{4.8}$$

Il descrittore viene calcolato a due scale spaziali adiacenti, utilizziamo $\sigma = 1,2$ e $\tau = .5$.

Il problema di rappresentare l'intorno del punto di interesse come istogramma delle orientazioni è stato affrontato in [25, 26, 45]. Una metodica per la quantizzazione dell'angolo solido è proposta da Scovanner et al. [45]. Nel loro lavoro viene effettuata direttamente applicando un peso variabile in base all'intervallo; 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. Per evitare questo problema si può usare un altro stratagemma come suggerito da Scovanner et al. ovvero usando i poliedri regolari e quantizzando l'angolo secondo le normali alle facce del poliedro. Questa tecnica viene infatti ripresa ed estesa da Schmid et al. [23]; in questo lavoro, utilizzando il concetto di *integral video* (vedi

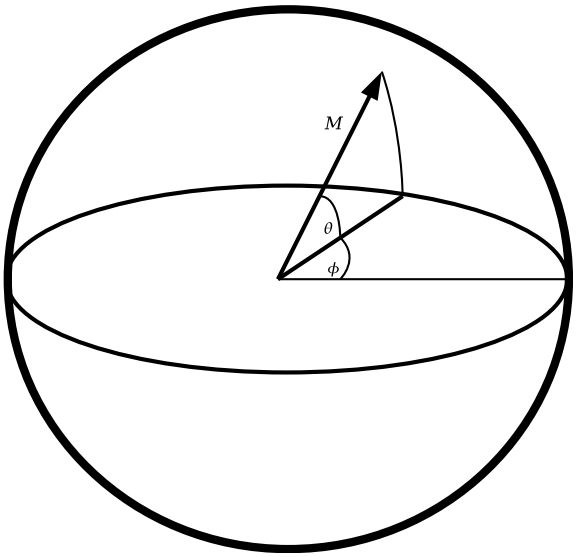


Figura 4.4. Gradiente 3D in coordinate polari.

Sezione 2.2.2) gli autori suddividono le regioni di interesse spazialmente e temporalmente in 8 sotto-volumi e per ognuno calcolano una statistica dei gradienti 3D quantizzando direttamente l'angolo, proiettando ciascun campione all'interno della sottoregione lungo la normale alla faccia di un poliedro regolare (sono utilizzati l'icosaedro ed il dodecaedro).

Questo metodo [23] ottiene ottimi risultati, comparabili allo stato dell'arte (vedi Tabella 5.2), tuttavia l'uso di questo descrittore è soggetto ad una ricerca dei parametri ottimali (scala a cui è calcolato il gradiente, numero di facce del poliedro regolare, scala di integrazione delle sottoregioni e numero delle sottoregioni) al fine di ottenere la migliore prestazione.

In questo lavoro piuttosto si cerca di formulare un descrittore con parametri ragionevoli [31] che lo rendano flessibile, generico ed il più possibile indipendente dal dataset analizzato. Si propone inoltre un approccio più semplice ma che si dimostra, a livello di prestazioni, comparabile con l'attuale stato dell'arte su due dei dataset più diffusi. Nel nostro approccio gli istogrammi degli gli angoli ϕ e θ vengono calcolati separatamente e successivamente concatenati. Esponiamo il procedimento in maniera più dettagliata.

Per ogni punto di interesse viene estratta una regione proporzionale alla scala a cui è stato rilevato. Questa regione è "fisicamente" un parallelepipedo

di pixel. Il valore dei pixel viene normalizzato tra -1 e 1. Per ogni regione estratta vengono calcolati $3 \times 3 \times 2$ istogrammi locali di ϕ e θ. Vengono cioè divise le dimensioni spaziali in 3 e quella temporale in 2. L'angolo ϕ viene quantizzato su 8 valori compresi tra $-\pi$ e π, mentre l'angolo θ variando unicamente tra $-\frac{\pi}{2}$ e $\frac{\pi}{2}$ viene quantizzato su 4 valori. Inoltre il gradiente viene calcolato a due scale adiacenti, convolvendo il volume estratto nel punto di interesse con un kernel gaussiano. Ogni sottoregione è selezionata con delle finestre gaussiane tridimensionali (blob gaussiani), i cui valori della covarianza e delle medie sono scelti in modo da ricoprire la maggior parte del volume, dando inoltre luogo ad una lieve sovrapposizione delle sottoregioni. Ogni valore dell'orientazione contribuisce ai relativi istogrammi con un peso proporzionale al modulo M_{3D}. Infine i singoli istogrammi di ogni regione vengono normalizzati e concatenati per formare un vettore di dimensione $3 \times 3 \times 2 \times (8 + 4) \times 2 = 432$.

Uno degli innegabili vantaggi di questo descrittore è la capacità di rappresentare ciascuna regione estratta dal filmato sia dal punto di vista dell'aspetto sia dal punto di vista del moto. Oltre quindi a cogliere un pattern di moto all'interno del cuboide estratto dal rilevatore, viene rappresentato efficacemente l'aspetto locale dell'azione. Quindi se in un'azione vengono mosse prevalentemente le mani, il rilevatore tenderà a dare risposte alte in punti prossimi alle mani, e la sola presenza di molte parole visuali rappresentanti la mano dell'attore potranno indicare l'appartenenza alla classe assegnata (ad esempio *handwaving*) nel dataset KTH (vedi Capitolo 5).

4.2.2 Descrittore basato sull'optical flow

L'optical flow riflette la variazione di un'immagine dovuta al moto durante un intervallo di tempo dt. Il campo dell'optical flow rappresenta una stima del campo di moto tridimensionale degli oggetti presenti nella scena. Questa stima è rappresentata dal moto apparente dei pixel tra due frame. C'è in realtà da distinguere tra campo di moto e campo dell'optical flow; un caso paradigmatico è "l'insegna del barbiere". L'oggetto ruota in senso antiorario, tuttavia l'optical flow a causa del pattern diagonale indica una velocità verso

l'alto.

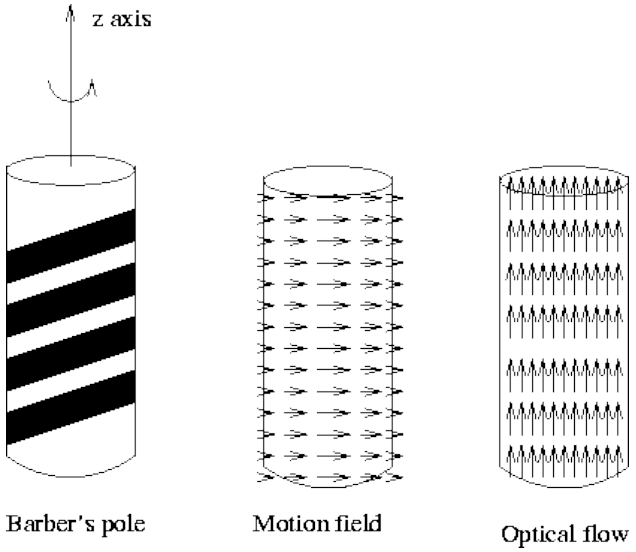


Figura 4.5. Optical flow errato dato dal pattern diagonale del cilindro.

Nonostante quindi non ci sia equivalenza tra campo dell'optical flow e campo della velocità, il valore dell'optical flow viene tipicamente utilizzato per stimare la velocità della telecamera o di oggetti che si muovono nella scena.

Il calcolo dell'optical flow è basato su due assunzioni:

- La luminosità di ciascun pixel appartenente allo stesso oggetto è costante nel tempo.
- Punti vicini nel piano dell'immagine si muovono secondo la stessa "legge".

Date queste due assunzioni esprimiamo la funzione $I(x, y, t)$ nel punto (x, y, t) secondo il suo sviluppo di Taylor:

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + I_x \cdot dx + I_y \cdot dy + I_t \cdot dt + \dots . \quad (4.9)$$

Supponendo che i termini di ordine più elevato siano trascurabili e che l'intorno di x, y sia traslato di una distanza trascurabile (dx, dy) possiamo scrivere

$$-I_t = I_x \cdot \frac{dx}{dt} + I_y \cdot \frac{dy}{dt}. \quad (4.10)$$

Da questa equazione possiamo ricavare iterativamente il valore di $(\frac{dx}{dt}, \cdot\frac{dy}{dt})$. Esistono vari algoritmi studiati per questo scopo. In particolare si può stimare una versione robusta dell'optical flow usando solo alcuni punti (rappresentazione sparsa) oppure possiamo cercare di stimare la velocità apparente di ciascun pixel (rappresentazione densa).

Poiché in questo lavoro questa stima è unicamente utilizzata per catturare le caratteristiche locali del movimento nell'intorno del punto di interesse, viene usata formulazione densa dell'optical flow, in particolare viene usato l'algoritmo di Lucas&Kanade [32].

Questo approccio ha un precedente nella metodologia olistica sviluppata da Efros et al. [13] i quali ottengono ottime performance nonostante la ridotta risoluzione dell'attore osservato (12 pixel). Inoltre esperimenti di neuroscienze dimostrano come il moto del corpo umano sia riconoscibile da un soggetto umano anche solo dalle traiettorie delle articolazioni; tipicamente vengono utilizzati dei marcatori luminosi puntiformi posizionati in corrispondenza delle principali giunture del corpo (mani, gomiti, ginocchi, collo...). Si rimanda alla pubblicazione di Blake e Shiffrar [4] per un approfondimento sulla percezione del moto dal sistema visivo umano. Per il nostro lavoro ci basti sapere che anche pochi frame di un'immagine ottenuta coi marcatori di cui sopra, se osservati in ordine temporale consistente, permettono ad un osservatore umano di comprendere che azione stava compiendo il soggetto ripreso. Pare quindi evidente che una rappresentazione del moto dei pixel, per quanto rumorosa, possa dare un contributo importante al riconoscimento delle azioni umane.

Tuttavia Dollár et al. [12] riportano che i descrittori basati sull'optical flow hanno performance nettamente peggiori di quelli basati sul gradiente. Mentre Fei Fei et al. [40] segnalano che nei loro esperimenti le performance sono analoghe, Laptev et al. [26] espongono un risultato che migliora stato dell'arte sul dataset KTH delle azioni umane, ottenuto con descrittori basati su orientazioni dei gradienti e dei vettori di optical flow; tuttavia in questo lavoro la rappresentazione dei clip non è di tipo puramente bag-of-words, non è quindi chiaro se l'incremento di prestazioni è dato dai descrittori o dall'uso di una rappresentazione maggiormente strutturata dei dati.

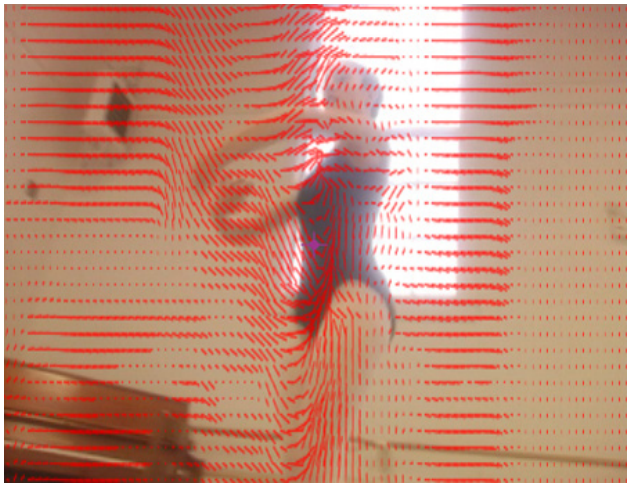


Figura 4.6. Optical flow generato dal moto di una ballerina.

Un altro contributo di questo lavoro è quindi l'implementazione di un descrittore basato sulle orientazioni dell'optical flow, stimato con l'algoritmo di Lucas&Kanade. Si consideri quindi l'optical flow calcolato per ogni due frame consecutivi all'interno del cuboide, per ogni pixel V_x, V_y sono le velocità stimate del moto apparente in quel punto. Lo si rappresenta quindi in coordinate polari:

$$\begin{aligned} M &= \sqrt{V_x^2 + V_y^2}, \\ \theta &= tan^{-1}(V_y/V_x) \end{aligned}$$

e viene creato un istogramma locale con una suddivisione del volume e delle orientazioni identica al descrittore basato sui gradienti. La performance di questo descrittore è inferiore a quella del precedente (vedi Tabella 5.1). La struttura di questo descrittore è analoga al SIFT (vedi Figura 2.2), con la differenza che gli istogrammi locali delle orientazioni sono calcolati per ogni due frame di ciascuna sottoregione.

4.2.3 Combinazione dei descrittori

Al fine di ottenere una migliore descrizione delle istanze è desiderabile formularla tramite la combinazione di più informazioni, nel nostro caso descrizioni

dell'aspetto e della traiettoria del moto. Questo tipo di fusione di informazioni può essere realizzata a più livelli, possiamo pensare di descrivere ciascun punto di interesse tramite la sua descrizione basata sul gradiente e sull'optical flow ad esempio. La prima soluzione implementata è stata quindi la concatenazione dei vettori di descrizione. Viene quindi realizzato un descrittore congiunto della *feature* pesando con un parametro λ i due descrittori singoli e concatenandoli. Il parametro λ viene determinato sperimentalmente. Un'altra tecnica per combinare le descrizioni la si ottiene dalla banale

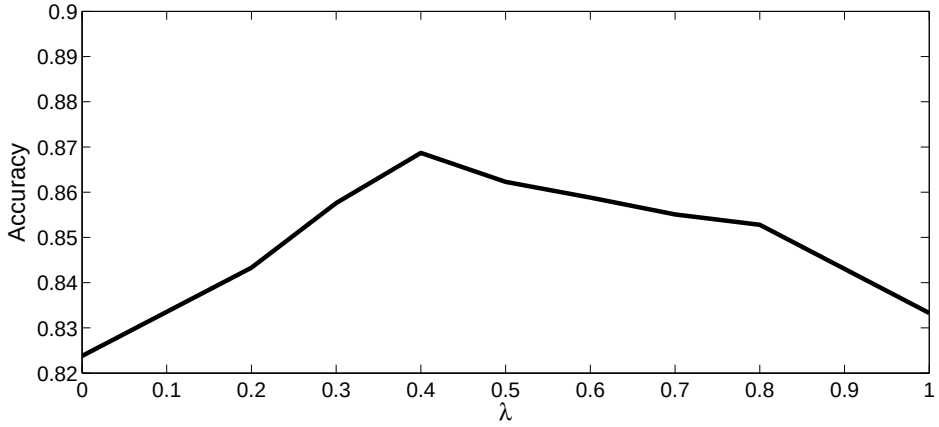


Figura 4.7. Valori dell'accuratezza al variare del parametro λ per il dataset Weizmann usando 500 parole.

concatenazione dei due istogrammi dati dal modello BoW. Ovvero dopo aver formato un dizionario per le *feature* descritte con il gradiente e uno per quelle descritte con l'optical flow, si ottengono due istogrammi potenzialmente di differente lunghezza. Concatenandoli otteniamo una descrizione congiunta della distribuzione delle parole ottenute dalla descrizione di moto e di aspetto. Grazie all'uso di classificatori SVM, il processo di ottimizzazione per il calcolo dell'iperpiano a massimo margine selezionerà per ogni classe il miglior sottoinsieme dei dati come insieme dei vettori di supporto. Questo in pratica si traduce in un processo implicito di selezione delle *feature*, si utilizzano cioè per ogni classe solo le parole più discriminanti ottenendo di fatto il massimo della performance su ogni classe.

4.3 Creazione di dizionari visuali efficaci.

La creazione di un dizionario consiste nella discretizzazione dello spazio delle *feature*. Applicando algoritmi come *k-means* o le sue varianti supervisionate si fa l'assunzione che la distribuzione delle *feature* nello spazio dei descrittori sia uniforme. Questo non è necessariamente vero; esistono strumenti per l'"esplorazione" dello spazio dei descrittori che ci permettono di capire effettivamente se è presente o meno una distribuzione non uniforme all'interno di questo spazio. Uno di questi strumenti è l'algoritmo di clustering basato su *meanshift* [9]. Questo algoritmo è tipicamente utilizzato per analisi di spazi delle *feature* in cui è probabile trovare distribuzioni non uniformi e cluster non convessi né gaussiani.

4.3.1 Problemi dell'algoritmo *k-means*

L'algoritmo *k-means* è formalizzato secondo una logica di ottimizzazione globale (vedi Sezione 3.3): ovvero data la funzione di costo 3.4 ed un insieme di parametri (i centri dei cluster e la matrice di associazione), il procedimento di assegnazione e calcolo dei centri è iterato finché non si è raggiunto un minimo locale. Nel fare questo quindi si prende in considerazione l'intero insieme di punti e la posizione di ogni centro è sempre influenzata dalla posizione degli altri e quindi dalle assegnazione in base alla tassellatura di Voronoi che ne deriva.

Se la distribuzione delle *feature* non è uniforme, i centri ottenuti tramite *k-means* sono attratti verso le zone ad elevata densità fornendo una regola di quantizzazione meno precisa per le zone meno dense. La non uniformità dello spazio delle *feature* è associata al tipo di campionamento delle immagini utilizzato [21]. In questo caso per campionamento si intende la strategia con cui si estraggono le regioni di immagini che poi verranno usate per la creazione del dizionario visuale. I recenti risultati nell'ambito della categorizzazione di immagini, scene o oggetti suggeriscono l'uso di un campionamento uniformemente casuale o addirittura di campionare le immagini densamente tramite una griglia di punti equidistanti a varie scale. In questo lavoro è stato usato un rilevatore di punti esplicitamente progettato per la selezione di regioni

informative per il moto. Il lavoro di Dollár [12] nasce direttamente nel contesto della classificazione ed il suo operatore ha la caratteristica di realizzare un campionamento denso del volume spazio-temporale nelle regioni reputate interessanti (vedi Capitolo 2). A causa di questo campionamento denso dei dati, ci si aspetta una non uniformità della distribuzione dei punti nello spazio dei descrittori; questo effetto si accresce ulteriormente a causa dell'uso di più scale spaziali e temporali. La frequenza delle parole visuali è distribuita secondo una legge a potenza [21, 50]. Questo è un fenomeno che si osserva anche in altri ambiti, ma è molto interessante notare che questa legge (Zipf) può essere usata ad esempio per modellare la distribuzione della frequenza delle parole di una lingua. Una legge a potenza indica che la probabilità di

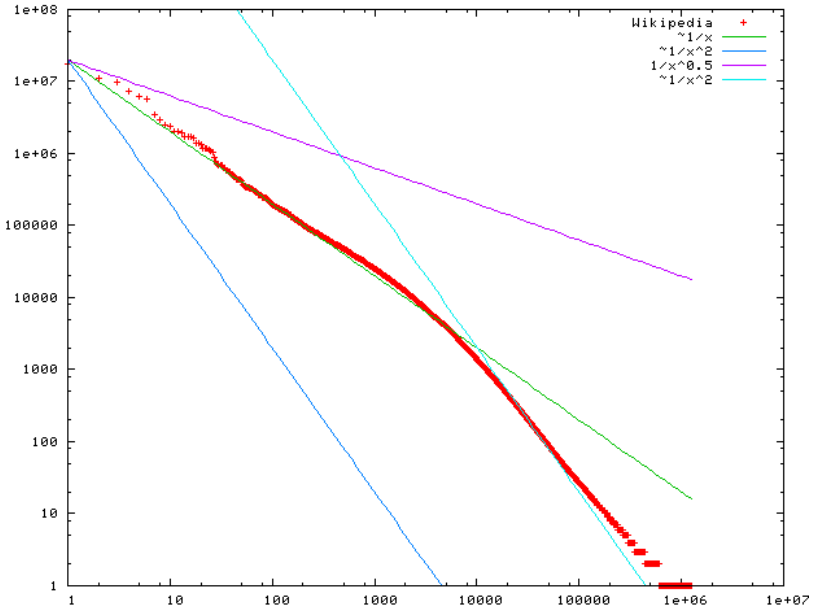


Figura 4.8. Probabilità delle parole nella lingua inglese all'interno di Wikipedia.

una parola (che possiamo stimare con la sua frequenza relativa all'interno di un *corpus* di documenti) decresce polinomialmente con il suo rango. Se ad esempio prendiamo il grafico in Figura 4.8 nel primo tratto la distribuzione delle parole segue una distribuzione del tipo $\frac{1}{x}$ ovvero la seconda parola più probabile occorre la metà delle volte della prima, la terza un terzo delle volte, la decima un decimo delle volte e così via. Più la retta è pendente e meno la

distribuzione delle parole è uniforme. Dal punto di vista linguistico la parola "the" occorrerà frequentemente (10^7 occorrenze nel corpus formato con le pagine di Wikipedia) e sarà una delle prime se non la prima (nel *corpus* formato dalle pagine di Wikipedia è la più frequente seguita da "of", "and" ...), per quanto riguarda la coda della distribuzione troveremo parole più rare come ad esempio "cuboid" o "spatiotemporal" che occorrono rispettivamente 30 e 18 volte in totale

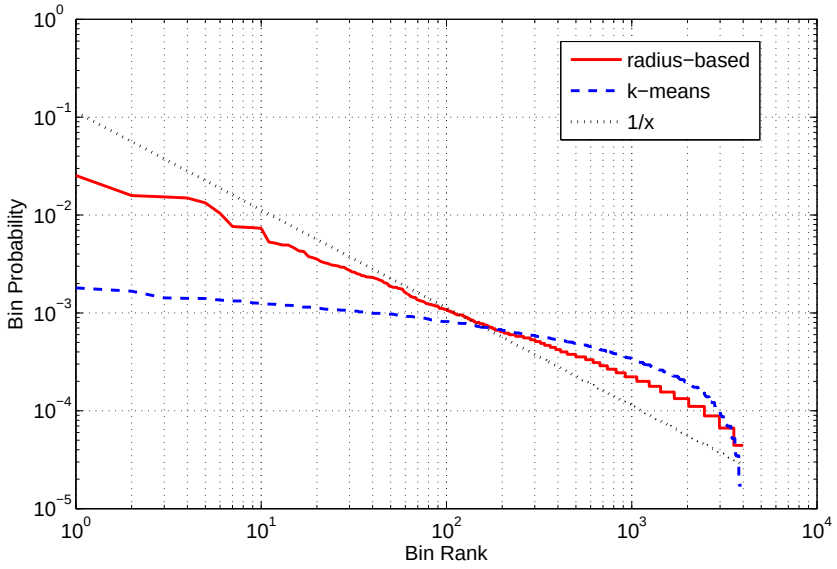


Figura 4.9. Probabilità delle parole visuali create con *k-means* e con algoritmo basato sul raggio in scala log-log nel dataset KTH.

In ultima istanza l'algoritmo *k-means* non è robusto nei confronti degli outlier; 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.

4.3.2 Clustering basato sul raggio

Al fine di capire la struttura dello spazio delle *feature* è stato implementato un algoritmo di clustering basato sul raggio [21]. L'algoritmo è definito in Figura 4.10. Si applica ad ogni passo un sottocampionamento per velocizzare

il procedimento. Viene quindi usato un popolare metodo (*meanshift*) che permette di localizzare regioni dense nello spazio delle *feature*.

0. Sia $D_s \subset D$ ottenuto da N campioni estratti casualmente dal dataset D.
1. $\forall x_i \in D_s$ inizializza un algoritmo di *meanshift*.
2. Sia M la moda che individua il cluster più denso trovata al passo 2.
3. Assegna al cluster rappresentato da M tutti i punti a distanza r da esso.
4. Elimina dal dataset D i punti etichettati.
5. Se $nclusters < max\_clusters$ e $|D| > 0$ vai a 1.

Figura 4.10. Algoritmo di clustering basato sul raggio.

L'algoritmo *meanshift* nasce come strumento di stima non parametrica della densità di probabilità. Fukunaga et al. nel 1975 [17] lo propongono come metodologia iterativa della stima del gradiente di una densità di proabilità. Lo scopo di stimare il gradiente della densità di probabilità è ovviamente di localizzarne i massimi, che ne identificano le mode. Ciò che è attraente di questa tecnica è la possibilità di localizzare le mode di una distribuzione senza doverla stimare puntualmente. Consideriamo uno stimatore della densità di probabilità:

$$\hat{f}_{h,K}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right), \tag{4.11}$$

dove K è un kernel radiale simmetrico.[1] Ad esempio prendiamo in considerazione il kernel normale

$$K_N = (2\pi)^{-d/2} \exp(-\frac{1}{2} \parallel x \parallel^2), \tag{4.12}$$

il cui profilo è definito dalla funzione:

$$k_N = \exp(-\frac{1}{2}x). \tag{4.13}$$

Se definiamo la funzione

$$g(x) = -k'(x), \tag{4.14}$$

[1]Un kernel, in questo contesto, è una funzione il cui integrale sia a somma unitaria e il cui limite all'infinito sia 0. Vedi Sezione 4.4.1

ammettendo che $k(x)$ sia continua e derivabile $\forall x \in [0, \infty)$, utilizzandola come profilo otteniamo il kernel

$$G(x) = c_{g,d} g(\| x \|^2), \tag{4.15}$$

dove $c_{g,d}$ è una costante di normalizzazione che fa sì che l'integrale di $G(x)$ sia unitario. Usando la definizione di profilo e la relativa notazione possiamo esprimere la 4.11 come

$$\hat{\nabla} f_{h,K}(x) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^{n} k\left(\left\| \frac{x - x_i}{h} \right\|^2 \right). \tag{4.16}$$

A questo punto per stimare il gradiente della densità di probabilità possiamo calcolare il gradiente della 4.16 ottenendo:

$$\hat{\nabla} f_{h,K}(x) = 2\frac{c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (x - x_i) k'\left(\left\| \frac{x - x_i}{h} \right\|^2 \right) \tag{4.17}$$

ed inserendo la 4.15 in quest'ultima si ottiene:

$$\hat{\nabla} f_{h,K}(x) = \frac{c_{k,d}}{nh^d} \left[\sum_{i=1}^{n} g\left(\left\| \frac{x - x_i}{h} \right\|^2 \right) \right] \left[\frac{\sum_{i=1}^{n} g\left(\left\| \frac{x - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} x_i g\left(\left\| \frac{x - x_i}{h} \right\|^2 \right)} - x \right] \tag{4.18}$$

che possiamo riscrivere come

$$\hat{\nabla} f_{h,K}(x) = \hat{f}_{h,G}(x) \frac{2c_{k,d}}{nh^{d+2}} m_{h,G}(x). \tag{4.19}$$

Il primo membro dell'equazione è proporzionale alla stima della densità di probabilità, mentre il secondo è il cosidetto *meanshift*, ovvero la differenza tra la media pesata usando il kernel G con banda h ed il centro del kernel x. Possiamo anche scrivere

$$m_{h,G}(x) = \frac{1}{2} h^2 c \frac{\hat{\nabla} f_{h,K}(x)}{\hat{f}_{h,G}(x)} \tag{4.20}$$

e quindi il vettore $m_{h,G}$ in ogni punto è proporzionale al gradiente *normalizzato*. Per cui in ogni regione del supporto della densità $f(x)$ da stimare il vettore $m_{h,G}$ ci indica la direzione di massima pendenza; questa proprietà

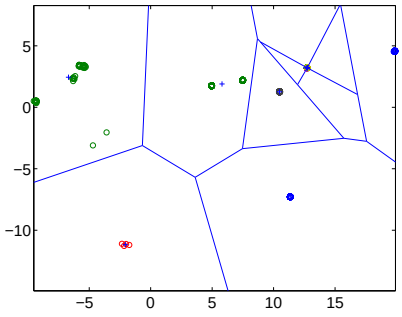
è interessantissima perché ci consente, dato un campione di osservazioni di procedere iterativamente alla localizzazione di punti stazionari della densità di probabilità, senza che una forma analitica né una stima non parametrica di questa sia computata esplicitamente.

L'algoritmo *meanshift* potrebbe essere usato direttamente come algoritmo di clustering [9], se inizializzato in un numero sufficiente di punti al termine della sua esecuzione ciascun punto si sarà spostato lungo la direzione di massima pendenza localizzando le mode della funzione di densità di probabilità. Tuttavia questo tipo di tecnica è molto dispendiosa computazionalmente e rischia di non localizzare le mode meno dense che potrebbero essere assorbite da mode più dense a loro prossime. Questo fenomeno lo si può osservare anche quando il raggio del processo di *meanshift* è troppo ampio.

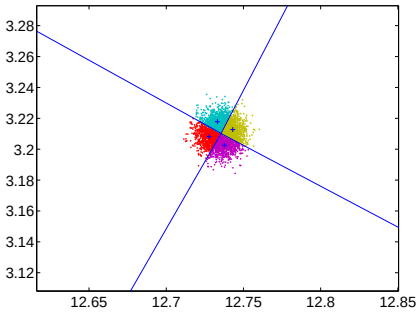
Grazie alla variante *online* implementata di questo algoritmo abbiamo analizzato lo spazio dei descrittori spazio-temporali verificando che l'assunzione di partenza era veritiera. In particolare si nota in Figura 4.8 come la frequenza delle parole visuali abbia un rank che decade a potenza. Nel grafico è stata inclusa una ideale distribuzione di Zipf, che modellerebbe appunto la lingua inglese, o una lingua in generale. Ci si aspetta inoltre che siano le parole di media frequenza ad essere le più discriminanti, così come avviene nella categorizzazione del testo; la codifica che effettua l'algoritmo *k-means* appare evidentemente inadeguata per le *feature* di media frequenza.

Piuttosto che cercare le regioni più dense dello spazio, l'algoritmo *k-means* cerca di minimizzare una funzione obiettivo che dipende dalla somma delle distanze di ogni punto dal centro del cluster. Un approccio del genere collocherà i centri densamente nelle zone più popolate dello spazio delle *feature* e mancherà di codificare le regioni più sparse. Inoltre la tassellazione di Voronoi derivante dall'assegnazione delle parole al centro più vicino sarà altamente distorta da questo fenomeno e darà luogo ad una copertura non uniforme dello spazio delle *feature*. Nella Figura 4.11 possiamo vedere su di un dataset sintetico il comportamento dei due algoritmi. In particolare nella Figure 4.11(a) e 4.11(b) vediamo come i centri dei cluster vengano attratti in una regione densa distorcendo l'assegnazione dei punti nelle regioni più sparse; al contrario l'algoritmo basato sul raggio permette una codifica

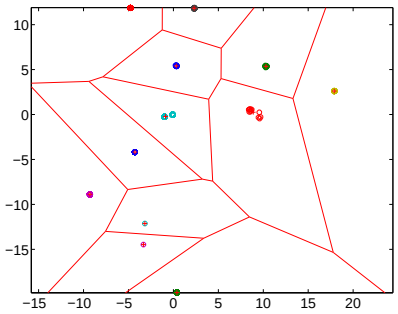
più uniforme dello spazio delle *feature*. Il raggio può essere scelto tramite una procedura non supervisionata, variandolo e ricercando il minimo di una funzione di misura della validità della procedura di clustering, ad esempio basata sulla varianza media per i punti membri dello stesso cluster o sulla distanza media tra i centri dei cluster.



(a) K-means.



(b) Dettaglio di una regione densa.



(c) Radius-based.

Figura 4.11. Clustering *k-means* e *radius-based* a confronto su di un dataset sintetico.

Per cui dobbiamo pensare alla ricerca delle parole visuali come la ricerca di mode della distribuzione delle *feature* estratte dal dataset. Pare infatti desiderabile localizzare i prototipi visuali in zone densamente popolate dalle *feature* di modo che il centro sia il più possibile rappresentativo per i descrittori ad esso associato.

4.4 Modellazione dell'incertezza nella quantizzazione

Il processo di *vector quantization (VQ)* è costituito da una fase di localizzazione dei valori discreti ed una fase di associazione del dato continuo ai primi. Nel nostro caso la prima fase è costituita dalla ricerca di zone dense e compatte nello spazio dei descrittori; come visto questo può essere fatto con un algoritmo come *k-means* o con un algoritmo più sofisticato basato sulla ricerca delle mode della densità di probabilità. Il secondo passo è tipicamente effettuato secondo una logica di vicinanza più prossima. Ovvero ciascun descrittore viene associato al cluster che gli si trova più vicino (nel nostro caso utilizziamo una metrica euclidea). Associare ciascun punto ad una sola parola, soprattutto quando si sta parlando di un dato percettivo, può provocare una grave perdita di informazione (vedi Figura 4.12); questa osservazione peraltro ha portato ad una recente rivalutazione di classificatori di tipo nearest-neighbour non basati su dizionari discreti [6]. Van Gemert et al. [51] effettuano un'analisi dettagliata di questo problema; nel loro lavoro vengono individuati due principali problemi nell'assegnazione dei descrittori alle parole discrete, ovvero la plausibilità della rappresentazione e l'incertezza. Un algoritmo di codifica tramite dizionario visuale agisce secondo una logica di assegnazione unica. Ciascun descrittore continuo viene rappresentato dalla parola a lui più vicina. Si parla di plausibilità quando un descrittore è talmente dissimile da tutte le parole nel dizionario che rappresentarlo con la parola a lui più vicina crea un'eccessiva distorsione del dato inizialmente estratto. Si parla invece di incertezza quando un punto si trova al margine della regione di Voronoi corrispondente al centro a cui verrà associato; quindi la sua rappresentazione poteva essere equivalentemente fornita da un'altra parola.

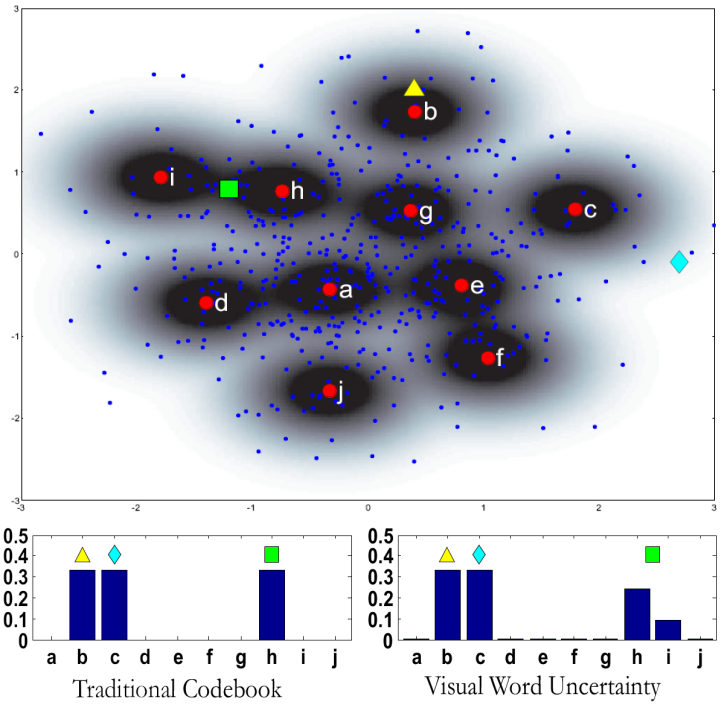


Figura 4.12. Esempio bidimensionale di VQ. Nel modello tradizionale (a sinistra in basso) viene assegnata ciascuna *feature* al cluster più vicino. La modellazione dell'incertezza invece consente di esprimere ciascuna *feature* tramite più parole visuali.

4.4.1 Stima non parametrica della densità di probabilità

Come abbiamo visto nel Capitolo 3 ogni azione è modellata con la distribuzione delle parole visuali che compaiono in essa. L'assunzione che si fa nel progettare un sistema di classificazione statistico è che, una volta definito uno spazio delle *feature*, per ciascuna classe i valori dei descrittori estratti dagli elementi che la costituiscono abbiano una certa distribuzione. Lo scopo degli algoritmi di apprendimento è modellare per ogni classe questa distribuzione ed in particolare sfruttare la geometria del suo supporto al fine di classificare come positivi i membri effettivi della classe. Tuttavia come già introdotto nel Capitolo 3 sia per l'impossibilità di osservare infiniti esempi di ciascuna classe, sia per l'errore che deriva dalla nostra modellazione, la quale forzatamente non potrà cogliere tutti gli aspetti di ogni singola classe,

la distribuzione esatta di ciascuna classe non potrà essere nota né stimata perfettamente portando così ad un errore di classificazione. Secondo quanto esposto possiamo interpretare il funzionamento del nostro sistema in questi termini. In particolare tramite il primo passaggio (creazione del dizionario) ci riconduciamo ad uno spazio delle *feature* semplificato; successivamente si esprime la distribuzione dei descrittori per il clip esaminato come l'istogramma delle parole visuali. Questa operazione ci permette di dare una rappresentazione robusta ed agile da manipolare della distribuzione delle *feature* all'interno di un singolo filmato.

Avendo dato questa interpretazione agli istogrammi di bag-of-words possiamo analizzare in dettaglio il significato della loro generazione. In particolare, l'approccio seguito usualmente consiste nell'assegnare ciascun punto non usato nella generazione del dizionario al cluster più prossimo ottenendo così la distribuzione delle parole visuali:

$$FD(w) = \frac{1}{n}\sum_{i=1}^{n}\begin{cases}1 & \text{se } w = \underset{v\in V}{\operatorname{argmin}}(D(v,p_i));\\ 0 & \text{altrimenti;}\end{cases} \tag{4.21}$$

dove n è il numero di punti estratti dal clip dell'azione e $D(w,p_i)$ è la distanza tra la parola w presente nel dizionario V ed il descrittore p_i dell'i-esimo punto estratto.

Considerato il processo di VQ, la distribuzione delle parole visuali rappresenta un'approssimazione della distribuzione dei descrittori. L'istogramma di una densità di probabilità è di fatto uno stimatore non parametrico; una robusta alternativa a questo è la stima della densità tramite kernel [48]. Come già introdotto nella Sezione 4.3.2 una stima della densità di probabilità $f(x)$ è data da

$$\hat{f}_{h,K}(x) = \frac{1}{nh^d}\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right). \tag{4.22}$$

Si potrebbe quindi pensare di usare un kernel gaussiano per applicare uno smoothing all'istogramma delle parole ottenendo così:

$$KCB(w) = \frac{1}{n}\sum_{i=1}^{n} K\left((D(w,p_i))\right). \tag{4.23}$$

Grazie a questa formulazione è possibile introdurre una modellazione più robusta ed in particolare definire una formulazione per la distribuzione delle parole che tiene conto dell'incertezza:

$$UFD(w) = \frac{1}{n}\sum_{i=1}^{n}\frac{K_\sigma(D(w,p_i))}{\sum_{j=1}^{|V|}K_\sigma(D(v_j,p_i))}, \tag{4.24}$$

dove D è la distanza euclidea e K_σ è il kernel gaussiano.

4.4.2 Quantizzazione dello spazio dei descrittori

Questo tipo di formulazione ci consente di ovviare ad alcuni problemi che si accentuano particolarmente se i descrittori quantizzati sono ad elevatissima dimensionalità. In particolare se pensiamo ad ogni parola come ad una moda della nostra distribuzione di probabilità, limitandoci a considerarla localmente gaussiana ci rendiamo conto che al crescere della dimensione[2] del descrittore la disposizione dei punti inclusi nell'ellissoide tenderanno ad addensarsi in un sottile strato sul bordo di questo. In particolare, se si considera una distribuzione gaussiana standard, la distanza dei suoi campioni dall'origine sarà data da

$$D = \sum_{i=1}^{d} {X_i}^2, \tag{4.25}$$

in cui ciascuna X_i è una componente della variabile multivariata gaussiana. La variabile D sappiamo essere distribuita secondo una distribuzione chi-quadro:

$$f(x;d) = \begin{cases} \frac{1}{2^{d/2}\Gamma(d/2)}x^{(d/2)-1}e^{-x/2} & \text{per } x > 0; \\ 0 & \text{per } x \leq 0 \end{cases} \tag{4.26}$$

la quale ha media d e varianza $2d$. Per cui la maggior parte dei campioni si sposterà, all'aumentare della dimensionalità, sul bordo della sfera [3] come si può vedere in Figura 4.13.

[2]Stiamo parlando del cosidetto fenomeno della *curse of dimensionality*, termine coniato da Richard Bellmann per descrivere il problema causato dalla crescita esponenziale del volume associato con l'aumentare delle dimensioni di uno spazio vettoriale.

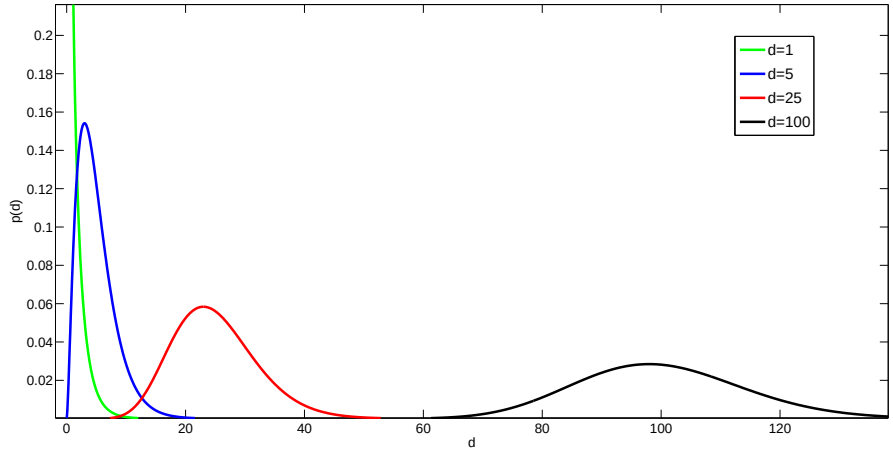


Figura 4.13. Distribuzione delle distanze dall'origine dei campioni di distribuzioni normali al variare della dimensionalità.

Dopo aver localizzato i centri delle parole visuali, occorre formulare una regola di quantizzazione per tutti i descrittori che non sono stati utilizzati per la creazione del dizionario. Si potrebbe usare una regola di quantizzazione basata sulla distanza euclidea per assegnare ogni punto al cluster più vicino. Quello che accade è che la maggior parte dei descrittori si troverà, come detto, sul bordo della sfera che identifica la parola visuale. Al crescere della dimensionalità del descrittore il problema dell'ambiguità è quindi certamente più urgente.

Si può inoltre dire che la plausibilità della quantizzazione, grazie all'uso di un algoritmo di clustering *radius-based*, è meno importante in quanto la disposizione dei centri nello spazio dei descrittori genera una tassellatura più uniforme (vedi Figura 4.11). Utilizzando un'assegnazione "morbida" delle parole si recupera parte dell'informazione persa durante il processo di quantizzazione dello spazio dei descrittori, ed in particolare questo procedimento si dimostrerà vantaggioso utilizzando un basso numero di parole, fatto che ovviamente accresce l'ambiguità di ciascun punto. Più parole si utilizzano per codificare lo spazio dei descrittori e meno ambiguità sarà presente per ciascun punto e minore quindi sarà il beneficio di questa tecnica di quantizzazione incerta.

Capitolo 5

Risultati sperimentali

Questo capitolo descrive i dataset utilizzati, il nostro set-up sperimentale ed i risultati ottenuti con le tecniche illustrate nei capitoli precedenti; vengono inoltre comparati i nostri risultati con quelli di recente pubblicazione.

5.1 Dataset

Per misurare le prestazioni del nostro approccio abbiamo utilizzato due dei più diffusi dataset contenenti sequenze video di attori umani ripresi durante l'esecuzione di varie azioni. Come visto nel Capitolo 1 le principali difficoltà nel modellare le azioni umane come categorie sono l'elevata variazione intra-classe e la possibilità che alcune azioni siano percettivamente molto simili nonostante siano semanticamente differenti ad esempio camminare e fare jogging.

La variazione intra-classe è data sia dal diverso aspetto che possono avere le persone nei video (abiti, sesso o postura) sia, soprattutto, dalla differente modalità di esecuzione dei singoli gesti da parte di persone differenti. Per tener conto di questo, i dataset sono composti da filmati di più attrici ed attori possibili e considerando condizioni di ripresa diverse (vedi ad esempio il dataset alla Sezione 5.1.2).

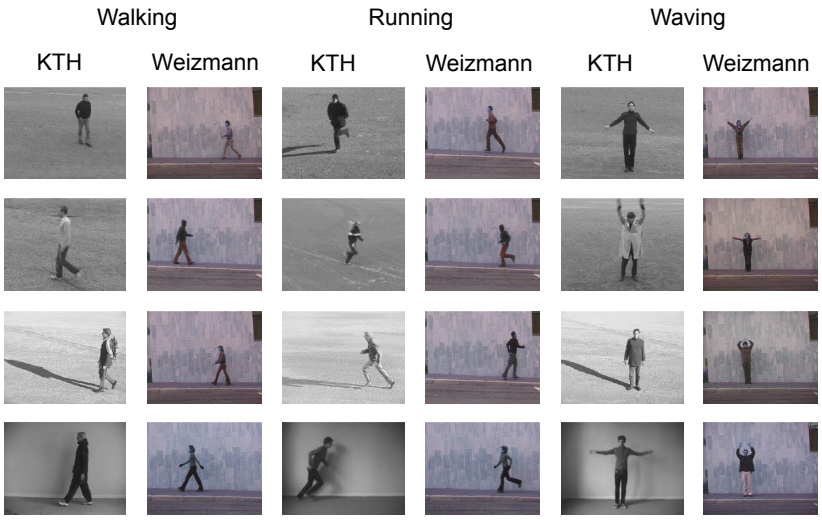


Figura 5.1. Confronto tra azioni identiche presenti nei due dataset analizzati.

5.1.1 Weizmann

Il dataset Weizmann contiene sequenze video di 10 azioni eseguite da 9 attori per un totale di 93 filmati (alcune azioni di alcuni attori sono presenti più di una volta). Le azioni sono eseguite una o più volte all'interno della stessa sequenza in condizioni di illuminazione, punto di vista e sfondo costanti. La figura della persona sarebbe facilmente segmentabile anche con tecniche di sottrazione dello sfondo.

Questo dataset è più datato e nasce nell'ambito di lavori con metodologie olistiche [59]. Ciò che è interessante di questo dataset è la grande quantità di azioni e la forte similarità inter-classe di alcune di queste. Le azioni sono: piegarsi (*bend*), correre (*run*), camminare (*walk*), saltare sul posto (*pjump*), in avanti a piedi uniti (*jump*), o con un piede solo(*skip*), eseguire l'esercizio *jack* ovvero portare le mani sopra la testa ed unire i piedi e successivamente allargare le gambe e portare le braccia in basso, salutare con uno o due braccia(*wave1, wave2*), muoversi lateralmente (*side*). La risoluzione dei filmati è di 180×144 pixel, le figure umane sono alte circa 60 pixel.

5.1.2 KTH

Il dataset KTH nasce come benchmark per lavori più recenti e cerca di includere un maggior numero di variazioni di condizioni. Viene perciò raccolto un insieme di sequenze video quanto più possibile eterogeneo dal punto di vista dell'aspetto. Vengono innanzitutto usati 25 attori comprendenti ambo i sessi. Lo stesso attore viene ripreso in quattro differenti condizioni: all'esterno, all'esterno con variazioni di scala (zoom della telecamera o traiettorie dirette verso e in direzione opposta all'osservatore), all'esterno con abiti differenti ed all'interno. Le riprese esterne presentano variazioni di illuminazione anche elevata (riprese effettuate ad orari molto diversi) e la telecamera è stabilizzata manualmente. Ogni attore è ripreso in ogni clip mentre ripete la stessa azione quattro volte, viene fornito un file di testo per separare i clip in quattro sequenze ciascuno. In totale sono presenti 2391 video. Le azioni presenti nel dataset sono: correre (*running*), camminare (*walking*), fare corsa leggera(*jogging*), dare pugni (*boxing*), applaudire (*handclapping*) e salutare (*handwaving*). È da notare la somiglianza semantica oltre che percettiva dei filmati delle prime tre azioni citate. In particolare associare la classe *jogging* o la classe *running* ad i rispettivi filmati può risultare di una certa complessità anche per un operatore umano. La risoluzione dei filmati è 160×120 pixel, mentre gli attori sono alti in media (nei filmati a scala costante) 90 pixel.

Il dataset KTH è considerato più complesso a causa dell'ampia variazione in scala, illuminazione e aspetto degli attori; l'aspetto è più vario a causa della presenza di ombre, diverso tipo di abbigliamento ed i luoghi in cui sono effettuate le riprese (vedi Figura 5.1).

5.2 Set-up sperimentale

I due dataset sono suddivisi in 9 (Weizmann) e 25(KTH) sottoinsiemi; ciascun sottoinsieme contiene unicamente i video di un attore.

Per il dataset Weizmann viene riportato il risultato della cross-validazione 9-fold; ovvero viene addestrato il sistema sull'insieme costituito da 8 sottoin-

siemi e successivamente vengono predette le categorie del rimanente attore. Questo procedimento viene ripetuto per 9 volte ottenendo quindi per ogni video del dataset una predizione sulla sua categoria.

Gli esperimenti sul dataset KTH vengono svolti secondo la seguente metodica: vengono utilizzati 16 attori come insieme di addestramento e per la validazione del modello, mentre i rimanenti 9 per misurare la prestazione finale. Il dizionario viene creato usando un sottoinsieme del *training set*.

Per selezionare i parametri del kernel χ^2 viene effettuata una cross-validazione 16-fold sull'insieme di addestramento, ottenendo così il valore ottimale per i parametri γ e C, successivamente viene addestrato il classificatore SVM su tutti i video dei primi 16 attori e viene effettuato il test sui rimanenti 9.

Sia usando *k-means* sia usando l'algoritmo basato sul raggio per la creazione del dizionario, gli esperimenti sopra descritti vengono ripetuti identicamente più volte e viene riportata la media del risultato. Questa procedura è resa necessaria dalla natura casuale dell'inizializzazione di entrambi gli algoritmi. Infatti l'algoritmo *k-means* inizializza la posizione iniziale dei centri casualmente, usando k campioni di una distribuzione uniforme. Similmente l'algoritmo basato sul raggio, ad ogni iterazione, sottocampiona uniformemente il dataset iniziale delle regioni spazio-temporali.

5.3 Valutazione dei descrittori

Come prime prove sperimentali sono effettuati gli esperimenti descritti nella Sezione 5.2 utilizzando il descrittore basato sull'optical flow ed il descrittore basato sul gradiente 3D (vedi Sezione 4.2.1). Entrambi i descrittori mostrano prestazioni in linea con i lavori più recenti (vedi Tabella 5.2).

Nelle seguenti matrici di confusione ciascuna riga indica la reale categoria a cui appartengono i clip e le colonne indicano la categoria che gli è stata assegnata dal classificatore. I risultati sono espressi in percentuale ed in particolare sugli elementi diagonali si può leggere la percentuale di clip classificati correttamente per ciascuna azione.

È interessante notare come la confusione avvenga principalmente tra le azioni *jogging* e *running*, e tra *handclapping* e *boxing* nel dataset KTH. Le

prime due sono evidentemente molto simili percettivamente, ciò che varia è principalmente la scala temporale dei microeventi rilevati dall'operatore descritto alla Sezione 4.1. È inoltre molto interessante notare come le azioni che coinvolgono gli arti superiori vengano confuse tra loro ma praticamente mai con le azioni che coinvolgono principalmente gli arti inferiori. Per l'occhio umano l'azione *handclapping* non è facilmente confondibile con *boxing*, se però pensiamo al tipo di dati che vengono utilizzati ci rendiamo conto che il pattern di moto per entrambe queste classi è analogo, in entrambi i casi le mani vengono mosse rispetto al frame secondo un moto laterale. È quindi plausibile una minima confusione anche tra queste due classi. A sostegno di questa osservazione si fa notare come nella matrice di Figura 5.3 la confusione tra *boxing* ed *handclapping* è elevatissima, e questo è facilmente imputabile al fatto che il descrittore basato sull'optical flow è unicamente in grado di misurare la velocità apparente senza dare grande importanza all'aspetto delle *feature* estratte. In particolare una mano aperta (*handclapping*) che si muove da sinistra a destra può generare un optical flow molto simile ad una mano chiusa che compie lo stesso movimento (*boxing*). È inoltre interessante notare che l'errore di classificazione per la classe *handclapping* per il descrittore HoF (vedi Figura 5.3) è minore rispetto al descrittore 3DGrad nonostante la confusione con la classe *boxing*. Questo risultato ci fornisce la motivazione per cercare di sfruttare entrambe le rappresentazioni delle *feature* e di ideare dei metodi di fusione dei descrittori. Ci si attende infatti che utilizzando entrambe le descrizioni, oltre a migliorare il tasso di riconoscimento medio, si migliori anche la capacità del classificatore di ridurre la confusione tra azioni simili. Per quello che riguarda il dataset Weizmann possiamo notare come, analogamente al dataset KTH, le azioni in cui la figura umana è statica rispetto al frame (bend, pjump, wave1, wave2) ottengano un'accuratezza elevatissima. Mentre le azioni che implicano il moto all'interno del frame (analogamente a running e jogging nel dataset KTH) sono quelle più soggette a confusione.

Il descrittore basato sull'optical flow ha performance peggiori di quello basato sul gradiente. La cattiva prestazione di questo descrittore è imputabile alla ridotta risoluzione degli attori nel dataset Weizmann (60 pixel)

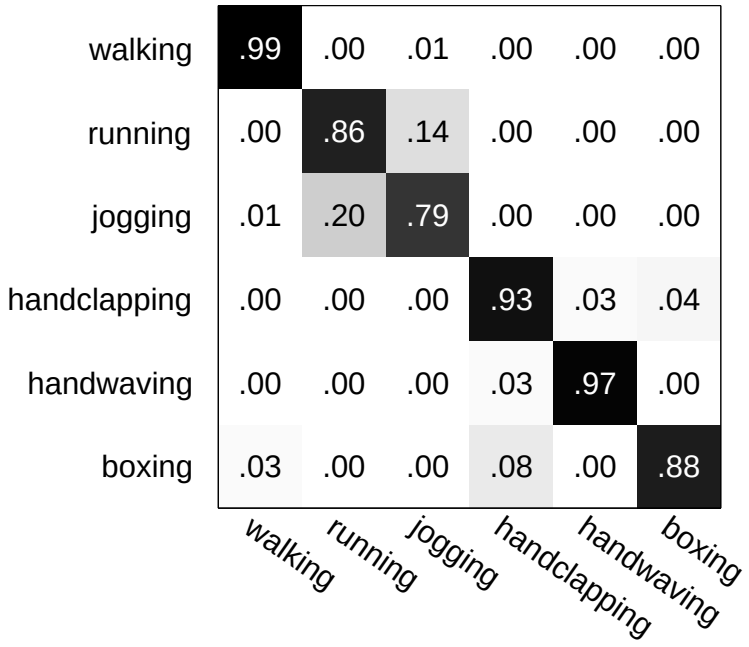


Figura 5.2. Matrice di confusione per il descrittore basato sul gradiente ed il dizionario creato con *k-means* per il dataset KTH.

e quindi alla conseguente eccessiva rumorosità della misura dell'optical flow in una regione ridottissima di spazio. Nella Tabella 5.1 vediamo come la combinazione dei due descrittori realizzata al livello della *feature* permetta di migliorare le prestazioni del descrittore basato sull'optical flow ma non permette un miglioramento rispetto al descrittore basato sul gradiente usato singolarmente. La combinazione degli istogrammi delle due differenti descrizioni piuttosto permette un maggiore incremento delle prestazioni e nel caso specifico rappresenta il miglior risultato sul dataset KTH ed il secondo migliore sul dataset Weizmann. Rispetto a quest'ultimo dataset il descrittore basato sul gradiente rappresenta in assoluto la migliore opzione e nessuna delle due combinazioni permette un miglioramento. Per questo nei successivi esperimenti verrà utilizzato il descrittore basato sul gradiente come migliore scelta per il nostro sistema di riconoscimento automatico.

Questa prima serie di esperimenti è stata eseguita usando il dizionario creato con *k-means* allo scopo di comparare unicamente il descrittore e le sue

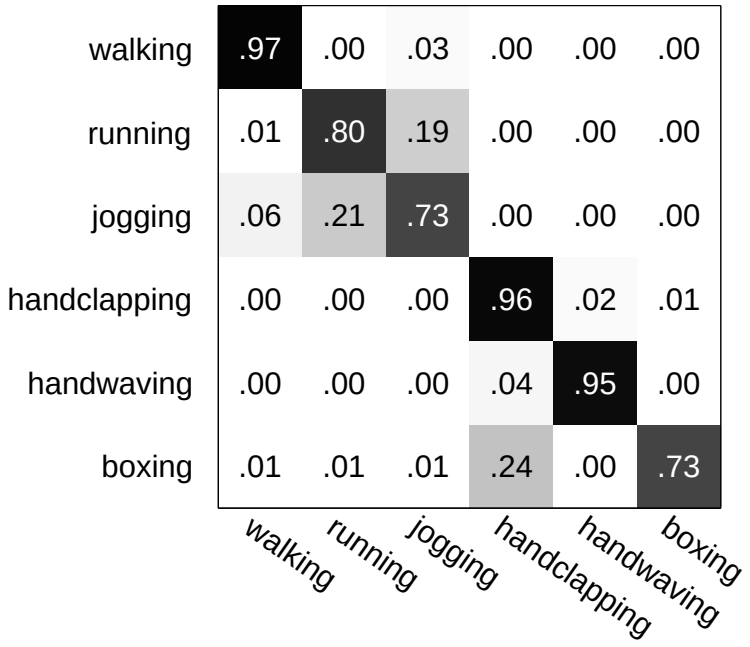


Figura 5.3. Matrice di confusione per il descrittore basato sull'optical flow ed il dizionario creato con *k-means* per il dataset KTH.

varianti, date dalle combinazioni, con l'attuale stato dell'arte. Come si vede in Tabella 5.2 alle righe 2 e 3 il nostro descrittore basato sul solo gradiente supera le prestazioni dei singoli descrittori usati da Laptev et al.

Descrittore	**KTH**	**Weizmann**
3DGrad	**90.38**	**92.3**
HoF	85.59	81.61
3DGrad_HoF (combinazione)	88.98	88.69
3DGrad+HoF (combinazione)	90.84	88.73

Tabella 5.1. Comparazione dei descrittori, singoli e combinati sui dataset KTH e Weizmann.

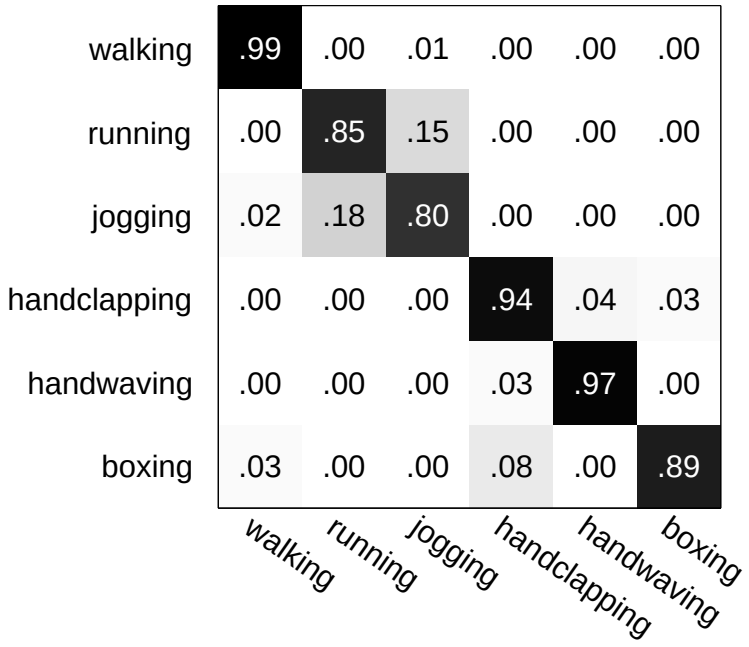


Figura 5.4. Matrice di confusione per la descrizione ottenuta dalla combinazione degli istogrammi ottenuti da dizionari creati con *k-means* per il dataset KTH.

5.4 Prestazioni della modellazione incerta

L'elevata dimensionalità del descrittore proposto ci motiva ad utilizzare un algoritmo differente da *k-means* per la localizzazione delle parole visuali. Viene quindi mostrato tramite un esperimento di classificazione come il dizionario creato con l'algoritmo *radius-based* proposto da Jurie e Triggs [21] si adatti perfettamente al nostro caso e ci permetta di costruire un migliore e più specifico vocabolario. Osservando il grafico in Figura 4.11 si nota come utilizzando poche parole (100) l'algoritmo *k-means* modella in modo più efficace lo spazio dei descrittori. Questo fenomeno è dovuto alla nostra scelta di usare le parole generate dall'algoritmo descritto alla Sezione 4.3.2 in ordine di creazione. Le prime parole sono quindi le parole ad elevata frequenza, parole comuni che in linea con il caso della TC codificano minore informazione ed hanno di conseguenza una minore capacità discriminante. All'aumentare delle parole il dizionario creato con l'algoritmo *radius-based* è più efficace; già

Figura 5.5. Le azioni *skip* e *jump* nel dataset Weizmann. Le azioni sono molto simili e vengono infatti confuse più delle altre.

a partire da 500 cluster si ottiene una migliore prestazione. Il metodo basato su *k-means* tende asintoticamente ad ottenere una performance comparabile, seppur inferiore. Applicando la regola di quantizzazione incerta si ottiene fin da subito (200 parole visuali) un miglioramento sia rispetto al *k-means* che rispetto al *radius-based* semplice.

Il beneficio dato dall'assegnazione "morbida" dei descrittori alle parole visuali è ovviamente più pronunciato per dizionari di ridotta dimensione. Questo perché la perdita di dati dovuta alla quantizzazione dello spazio delle *feature* è moderata dalla possibilità di rappresentare ciascun descrittore attraverso più parole (vedi la 4.24); mentre all'aumentare dei cluster lo spazio dei descrittori viene rappresentato in maniera adeguata dalle bag-of-words create con assegnamento univoco dei descrittori.

Al crescere della quantità di parole visuali, combinando il dizionario creato con l'algoritmo *radius-based* e la modellazione incerta dell'assegnazione si ottiene comunque un incremento dell'0.8% rispetto all'algoritmo *k-means*.

È inoltre interessante notare come usando solo le 100 parole più frequenti (tecnica *radius-based*) otteniamo un risultato di categorizzazione accettabile e comparabile con [12, 40, 54]; questo risultato ci indica perciò che la creazione di una lista di *stop-words* non è una scelta utile nella categorizzazione di dati visuali. E' tuttavia certamente vero che le parole di media frequenza sono quelle in grado di produrre il maggiore incremento di prestazioni.

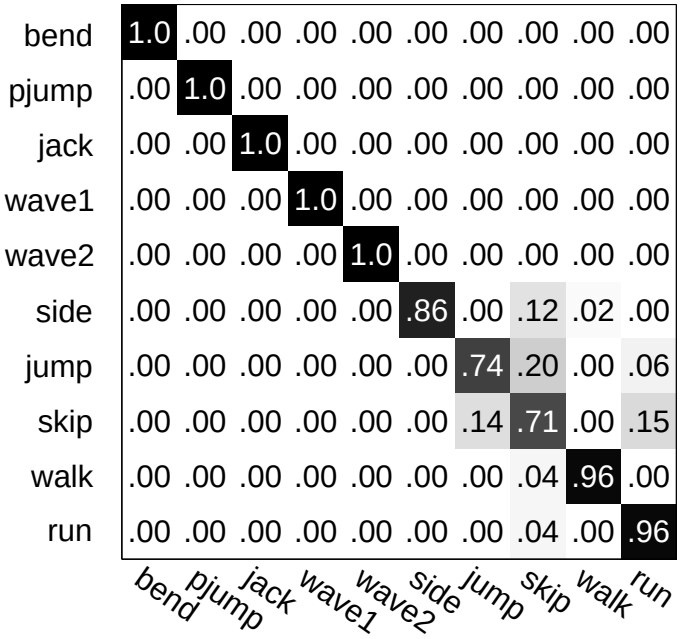


Figura 5.6. Matrice di confusione per il descrittore basato sul gradiente per il dataset Weizmann.

Inoltre dalla Figura 5.9 vediamo come l'uso del dizionario creato con l'algoritmo *radius-based* migliori sensibilmente le prestazioni per tre delle sei classi presenti, mentre peggiori unicamente per la classe handwaving. L'uso dell'assegnazione "morbida" alle parole del dizionario presenta sempre un sostanziale miglioramento tranne che per la classe *handclapping*.

Osservando la differenza tra la matrice di confusione in Figura 5.2 e quelle nelle Figure 5.10 e 5.11 possiamo vedere come l'uso di un algoritmo di creazione del dizionario più sofisticato unito ad una tecnica di quantizzazione che tenga conto dell'incertezza non solo aumenta l'accuratezza della classificazione ma è in grado di separare meglio le classi critiche diminuendo la confusione tra *jogging* e *running* e tra *handclapping* e *boxing*.

5.4.1 Comparazione con lo stato dell'arte

Il metodo proposto ottiene un'accuratezza di classificazione del **91.14%**, sul dataset KTH sfruttando 4000 parole visuali e la tecnica per la modellazio-

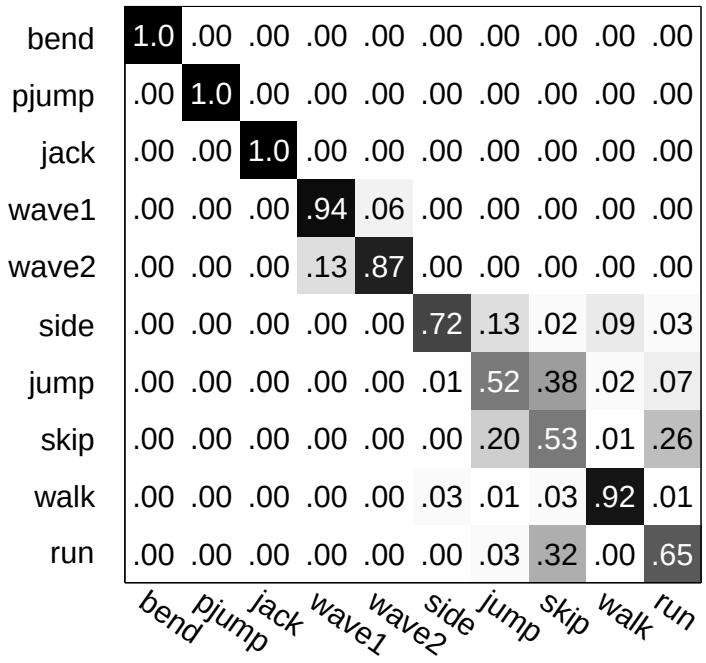


Figura 5.7. Matrice di confusione per il descrittore basato sull'optical flow per il dataset Weizmann.

ne dell'incertezza nella quantizzazione; le nostre prestazioni sono superiori a tutti gli approcci di tipo bag-of-words e seconde solo a quelle ottenute da Schmid et al. [23]. Tuttavia c'è da dire che il loro descrittore, per quanto formulato in maniera molto raffinata, necessita una pesante registrazione dei parametri, mentre il risultato ottenuto da Laptev et al. [26] di 91.8% non è direttamente comparabile al nostro in quanto è ottenuto tramite una combinazione di più informazioni con una forte valenza strutturale; in particolare viene effettuata una ricerca *greedy* per la migliore combinazione di griglie spazio-temporali su ciascun dataset. Il nostro metodo ha comunque prestazioni superiori a quelle ottenute dai singoli descrittori utilizzati da Laptev et al. nel suddetto lavoro (vedi Tabelle 5.2 e 5.1).

Considerando invece il dataset Weizmann il nostro metodo è migliore dei precedenti lavori di tipo bag-of-words [23, 40, 45] ed è pure migliore del risultato riportato da Liu et al. [29] (90.4%) in cui vengono fuse più *feature*

assieme. Non possiamo invece effettuare una comparazione diretta con Fathi e Mori [14] poiché utilizzano una rappresentazione olistica ed effettuano una segmentazione delle figure umane.

Metodo	**KTH**	**Weizmann**
Metodo proposto	**91.14**	**92.3**
Laptev et al. [26] - HoG	81.6	-
Laptev et al. [26] - HoF	89.7	-
Dollár et al. [12]	81.2	-
Wong e Cipolla [55]	86.62	-
Scovanner et al. [45]	-	82.6
Niebles et al. [40]	83.33	90
Liu et al. [43]	-	90.4
Kläser et al. [23]	91.4	84.3
Willems et al. [54]	84.26	-

Tabella 5.2. Comparazione del metodo proposto con altri metodi sui dataset KTH e Weizmann.

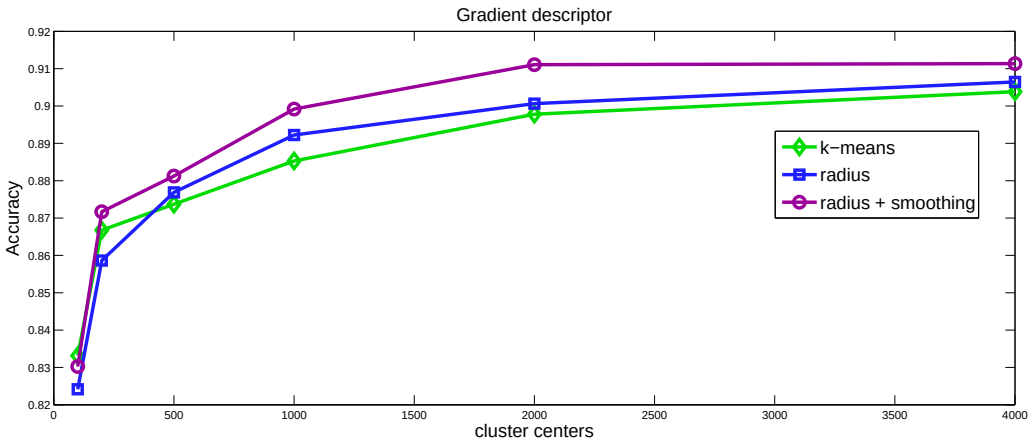


Figura 5.8. Comparazione dei dizionari creati con *k-means*, radius-based e *radius-based* con assegnamento soft.

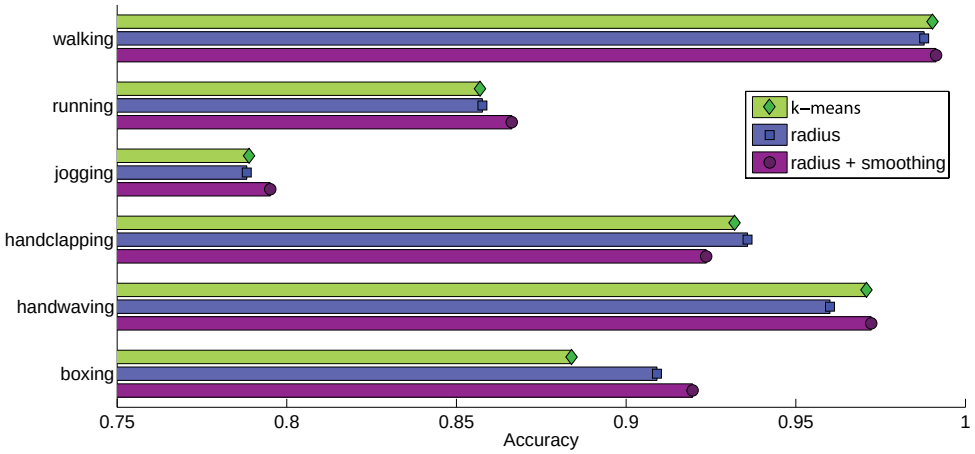


Figura 5.9. Comparazione dell'accuratezza ottenuta con vari tipi di quantizzazione sul dataset KTH per ogni classe.

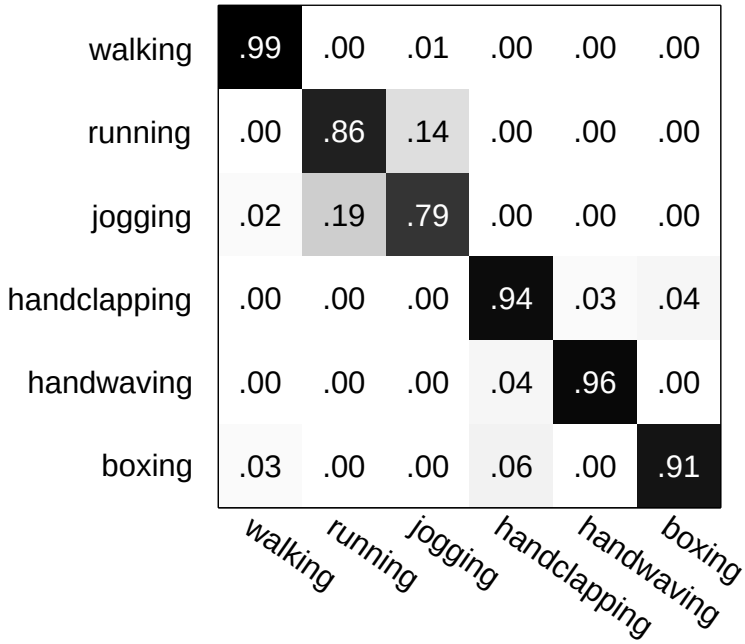


Figura 5.10. Matrice di confusione per il descrittore basato sul gradiente e dizionario creato con *radius-based* clustering per il dataset KTH.

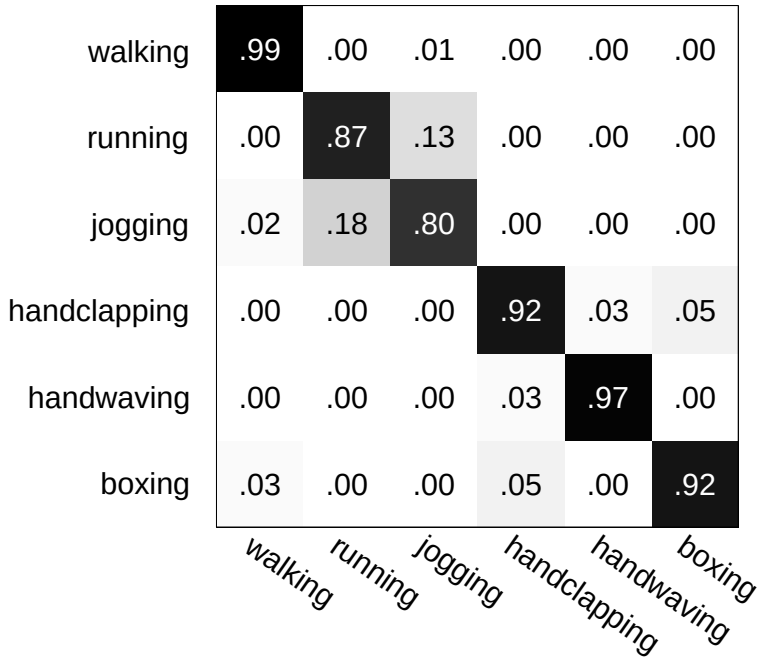


Figura 5.11. Matrice di confusione per il descrittore basato sul gradiente e dizionario creato con *radius-based* clustering con assegnazione soft per il dataset KTH.

Capitolo 6

Conclusioni e sviluppi futuri

In questo lavoro di tesi è stato realizzato un sistema di riconoscimento di comportamenti umani presenti in filmati video. La letteratura scientifica recente affronta questo problema modellando le azioni come insiemi non ordinati di *feature* visuali. Le *feature* visuali sono ottenute da punti di interesse spazio-temporali tramite i quali viene costruito un dizionario per la loro codifica. I risultati presentati in questo lavoro di tesi sono:

- L'implementazione e sperimentazione di un campionamento denso della scala spaziale e temporale tramite un operatore di estrazione di punti di interesse presentato in letteratura, originariamente sprovvisto di un meccanismo di selezione della scala spazio-temporale.
- La formulazione di due descrittori locali invarianti alla scala che non necessitano di taratura fine dei parametri.
- L'applicazione di una tecnica di generazione del dizionario visuale basata sulla ricerca di mode nella distribuzione dei descrittori.
- La riduzione dell'errore di quantizzazione tramite una modellazione dell'incertezza.

L'uso di un rilevatore in grado di realizzare un'estrazione densa nello spazio e nella scala, consente di estrarre un elevato numero di *feature* dall'alto contenuto informativo; questo, unito al descrittore basato sul gradiente presentato

nella Sezione 4.2.1, consente di ottenere prestazioni in linea con l'attuale stato dell'arte su di un dataset e di ottenere una prestazione superiore a tutti gli altri approcci di questo tipo presenti in letteratura (vedi Tabella 5.2).

I descrittori sono realizzati tramite statistiche posizionali del gradiente e dell'optical flow misurati nelle regioni estratte. Non viene applicato a questi dati nessuna tecnica di riduzione della dimensionalità (analisi delle componenti principali) allo scopo di non compromettere eventuali informazioni dipendenti da statisitiche di ordine superiore al primo. I descrittori così formulati hanno dimensione elevata, rispettivamente 432 e 144. Lo spazio dei descrittori deve essere codificato in maniera attenta, in questa tesi si mostra come la popolare tecnica basata su *k-means* fallisca nel codificare in maniera adeguata le regioni di frequenza intermedia. Implementando una variante *on-line* dell'algoritmo di clustering basato su stimatori *meanshift*, viene mostrato come la distribuzione dei descrittori proposti sia non uniforme; la distribuzione delle frequenze delle parole visuali così individuate segue la legge di Zipf, analogamente a quella delle parole di una lingua.

Infine l'uso di una tecnica di assegnazione morbida dei descrittori alle parole del dizionario permette di ottenere un ulteriore miglioramento delle prestazioni. Il nostro risultato rispetto all'attuale stato dell'arte è secondo solo al sistema di Kläser e Schmid [23]. Il loro sistema tuttavia richiede un pesante affinamento di tutti i parametri del descrittore locale; la ricerca di questi parametri viene effettuata sul *training set*. I loro esperimenti inoltre mostrano come i parametri appresi su di un dataset (i.e. KTH) non consentano di ottenere risultati allo stato dell'arte su di un altro (Weizmann) e viceversa. Il nostro approccio perciò è da considerarsi più generale.

Lo studio delle tecniche suddette ha fornito risultati incoraggianti. Le linee di ricerca a breve termine sono:

- Un'analisi più approfondita delle caratteristiche dei dizionari generati con l'algoritmo presentato nella Sezione 4.3.2. Si vuole individuare ad esempio un riscontro motivabile percettivamente del miglioramento ottenuto da questa tecnica.

- L'elaborazione di una tecnica di fusione delle due descrizioni proposte

che permetta di sfruttarne la parziale complementarità.

- L'applicazione di queste tecniche, unita ad uno studio sulla loro robustezza, in contesti reali complessi (film, documentari, videosorveglianza).

Il sistema realizzato è in grado di fornire annotazioni accurate di azioni. Le linee di ricerca a lungo termine, in cui questo lavoro evolve naturalmente, sono:

- La fusione di dati sulle traiettorie del moto delle persone in 3D allo scopo di ricavare relazioni spaziali che definiscano una semantica accurata dei comportamenti collettivi.

- Lo sfruttamento di modelli di moto umano appresi per rafforzare le predizioni di un sistema di tracking; ad esempio migliorare la disambiguazione dei bersagli e la precisione della stima. In questo contesto si potrebbe certamente pensare ad un sistema in grado di predire congiuntamente traiettoria e comportamento.

- Lo studio di rappresentazioni strutturate dell'azione; tutti i lavori presenti ad oggi mostrano come incorporare questo tipo di informazioni renda possibile un sensibile incremento di prestazioni. In linea con i punti proposti, ci interessa studiare modelli robusti il cui uso in filmati reali e complessi sia in grado di definire azioni e comportamenti collettivi.

Bibliografia

[1] S. Ali, A. Basharat, and M. Shah. Chaotic invariants for human action recognition. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

[2] A. D. Bagdanov, L. Ballan, M. Bertini, and A. Del Bimbo. Trademark matching and retrieval in sports video databases. In *Proc. of ACM Multimedia Information Retrieval (MIR)*, pages 79–86, 2007.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] R. Blake and M. Shiffrar. Perception of human motion. *Annual Review of Psychology*, 58:47–73, 2007.

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

[6] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2008.

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

[8] C. Colombo, A. Del Bimbo, and A. Valli. Visual capture and understanding of hand pointing actions in a 3-d environment. *IEEE Transactions on Systems, Man, and Cybernetics*, 33(4), 2003.

[9] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(5):603–619, 2002.

[10] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[11] J. Dever, N. da Vitoria Lobo, and M. Shah. Automatic visual recognition of armed robbery. In *Proc. of International Conference on Pattern Recognition (ICPR)*, volume 1, 2002.

[12] P. Dollár, V. Rabaud, G. Cottrell, and S. J. Belongie. Behavior recognition via sparse spatio-temporal features. In *Proc. of ICCV Int.'l Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VSPETS)*, pages 65–72, 2005.

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

[14] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[15] C. Fowlkes, S. Belongie, and J. Malik. Efficient spatiotemporal grouping using the Nyström method. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 231–238, 2001.

[16] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 13(9):891–906, 1991.

[17] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.

[18] T. Gärtner. A survey of kernels for structured data. *ACM SIGKDD Explorations Newsletter*, 5(1):49–58, 2003.

[19] L. Gorelick, M. Blank, E. Schechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(12):1395–1402, 2007.

[20] A. Hauptmann, R. Yan, W. H. Lin, M. Cristel, and H. Wactlar. Can high-level concepts fill the semantic gap in video retrieval? a case study with broadcast news. *IEEE Transactions on Multimedia (TMM)*, 9(5):958–966, 2007.

[21] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 604–610, 2005.

[22] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *In Proc. International Conference on Computer Vision(ICCV)*, pages I: 166–173, 2005.

[23] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *Proc. of British Machine Vision Conference (BMVC)*, pages 995–1004, 2008.

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

[25] I. Laptev and T. Lindeberg. Local descriptors for spatio-temporal recognition. In *Spatial Coherence for Visual Motion Analysis*, pages 91–103, 2004.

[26] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[27] S. Lazebnik and M. Raginsky. Supervised learning of quantizer codebooks by information loss minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 99(1), 2008.

[28] T. Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 2(21):224–270, 1994.

[29] J. Liu, S. Ali, and M. Shah. Recognizing human actions using multiple features. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[30] 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)*, pages 1–8, 2008.

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

[32] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of DARPA Image Understanding Workshop*, pages 121–130, 1981.

[33] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. of British Machine Vision Conference (BMVC)*, pages 384–393, 2002.

[34] K. Mikolajczyk, B. Leibe, and B. Schiele. Local features for object class recognition. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pages 1792–1799, 2005.

[35] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 525–531, 2001.

[36] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(10):1615–1630, 2005.

[37] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(10):1615–1630, 2005.

[38] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. van Gool. A comparison of affine region detectors. *International Journal of Computer Vision (IJCV)*, 65(7):43–72, 2005.

[39] K. Mikolajczyk and H. Uemura. Action recognition with motion-appearance vocabulary forest. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[40] J. C. Niebles, H. C. Wang, and F. F. Li. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision (IJCV)*, 79(3):299–318, 2008.

[41] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 490–503, 2006.

[42] A. Oikonomopoulos, I. Patras, and M. Pantic. Spatiotemporal saliency for human action recognition. In *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*, pages 430–433, 2005.

[43] S. Savarese, A. DelPozo, J. C. Niebles, and F. F. Li. Spatial-temporal correlatons for unsupervised action classification. In *Proc. of Workshop on Motion and Video Computing (WMVC)*, pages 1–8, 2008.

[44] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *International Conference on Pattern Recognition*, pages III: 32–36, 2004.

[45] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional SIFT descriptor and its application to action recognition. In *Proc. of ACM International Conference on Multimedia (MM)*, pages 299–318, 2007.

[46] F. Sebastiani. Machine learning in automated text categorization. *Journal of ACM Computing Surveys*, 34(1):1–47, 2002.

[47] N. Sebe, M.S. Lew, Y. Sun, I. Cohen, T. Gevers, and T.S. Huang. Authentic facial expression analysis. *Image Vision Computing*, 25(12), 2007.

[48] B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[49] J. Sivic and A. Zisserman. Video Google: Efficient visual search of videos. In J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, editors, *Toward Category-Level Object Recognition*, volume 4170 of *LNCS*, pages 127–144. Springer, 2006.

[50] T. Tuytelaars and C. Schmid. Vector quantizing feature space with a regular lattice. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

[51] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W.M. Smeulders. Kernel codebooks for scene categorization. In *Proc. of European Conference on Computer Vision (ECCV)*, 2008.

[52] P. Viola and M. Jones. Robust real-time object detection. In *International Journal of Computer Vision (IJCV)*. Springer, 2001.

[53] Schiele B. Vogle, J. Semantic modeling of natural scenes for content-based image retrieval. *International Journal of Computer Vision (IJCV)*, 72(2):133–157, 2007.

[54] G. Willems, T. Tuytelaars, and L. Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *Proc. of European Conference on Computer Vision (ECCV)*, 2008.

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

[56] S.-F. Wong, T.-K. Kim, and R. Cipolla. Learning motion categories using both semantic and structural information. In *Proc. of IEEE In-*

ternational Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–6, 2007.

[57] J. Yang, Y. G. Jiang, A. G. Hauptmann, and Chong W. Ngo. Evaluating bag-of-visual-words representations in scene classification. In *Proc. of ACM International Workshop on Multimedia Information Retrieval (MIR)*, pages 197–206, 2007.

[58] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proc. of International Conference on Machine Learning (ICML)*, pages 412–420. Morgan Kaufmann Publishers, 1997.

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

[60] J. G. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision (IJCV)*, 73(2):213–238, 2007.

Ringraziamenti

Desidero innanzitutto ringraziare il Prof. Alberto Del Bimbo per avermi offerto la possibilità di lavorare su un tema così affascinante e nell'ambiente più stimolante che si possa immaginare. Un grazie anche a Marco Bertini che assieme a Giuseppe e Lamberto mi ha fornito una costante guida in questi mesi; senza le vostre dritte e puntuali revisioni del mio lavoro, non ce l'avrei mai fatta.

Grazie ai miei genitori, che mi hanno sempre appoggiato e sostenuto nelle mie decisioni, e la cui pazienza viene finalmente ricompensata. Grazie ad Erika che mi è stata accanto, mi ha sopportato e stimolato a fare sempre del mio meglio. Grazie a tutti i ragazzi che ho incontrato al MICC, per aver condiviso con me la loro conoscenza ed aver stimolato continuamente la mia curiosità.

Un grazie ad Andrea, Leonardo, Matteo e Pietro le cui capacità intellettuali e l'amicizia hanno reso il mio percorso di studi più semplice; spero di esservi stato di aiuto quanto voi lo siete stati per me. Un doveroso ringraziamento va a Matteo, Valerio e a tutti coloro che hanno letto completamente o parzialmente questa tesi, per avermi fornito preziosi consigli.