



UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO

DIPARTIMENTO DI INFORMATICA

CORSO DI LAUREA IN INFORMATICA E TECNOLOGIE PER LA PRODUZIONE DEL SOFTWARE

---

TESI DI LAUREA IN  
SISTEMI MULTIMEDIALI

PIPELINE INNOVATIVA  
PER LA FOTOIDENTIFICAZIONE DEL GRAMPUS GRISEUS

**Relatore:**

Chiar.mo Prof. Giovanni Dimauro

**Correlatore:**

Chiar.ma Dott.sa Rosalia Maglietta

**Laureando:**

Emanuele Seller

---

Anno Accademico 2018-2019

## Sommario

1	Abstract.....	4
2	Introduzione .....	5
3	Metodologia .....	9
3.1	Metodologia manuale .....	9
3.2	Metodologia automatica.....	11
3.2.1	Il ritaglio automatico.....	14
3.2.2	L'estrazione delle features .....	22
3.2.3	La foto-identificazione automatica.....	31
4	Esperimenti e risultati .....	37
4.1	Acquisizione .....	37
4.2	I dati dell'esperimento .....	37
4.2.1	I modelli .....	37
4.2.2	Il dataset di valutazione.....	38
4.3	Risultati sperimentali .....	38
4.4	Presentazione dell'Output .....	40
5	Conclusioni e sviluppi futuri .....	41
6	Appendice: la descrizione del contorno .....	42
6.1	Divisione tra pinna e background .....	42
6.2	Rilevamento della punta della pinna e distinzione tra pinna destra e sinistra .....	44
6.3	Il contorno di interesse .....	47
6.4	I descrittori del contorno di interesse.....	49
6.4.1	Validità dell'approccio .....	52
7	Bibliografia .....	54

8	Ringraziamenti.....	<b>Error! Bookmark not defined.</b>
---	---------------------	-------------------------------------

## 1 Abstract

La foto-identificazione di esemplari di animali nel loro ambiente naturale è un obiettivo importante per la descrizione dell'ambiente stesso, per gli studi relativi alla distribuzione delle specie e sulla conservazione ed uso dell'habitat. L'analisi manuale delle foto collezionate da parte dei ricercatori e biologi è generalmente molto costosa in termini temporali e richiede una particolare esperienza. SPIR (*Smart Photo-Identification of the Risso's dolphin*) è un tool che consente di eseguire la foto-identificazione del delfino di Risso (*Grampus griseus*), una specie di particolare interesse in quanto poco conosciuta, attraverso il confronto delle pinne dorsali. La versione di SPIR proposta in questo lavoro di tesi implementa una serie di operazioni innovative che rendono il tool molto performante e applicabile a dataset di dimensioni molto elevate. SPIR è stato applicato in modo molto proficuo nell'analisi dei dati acquisiti nel 2019 nel Golfo di Taranto che includevano immagini scattate in quasi 50 giornate di avvistamento.

È stato inoltre avviato uno studio preliminare sul calcolo automatico del contorno della pinna dorsale, passaggio fondamentale per l'estensione dell'applicazione di SPIR ad altre specie di cetacei tra cui ad esempio il Tursiope (*Tursiops truncatus*).

## 2 Introduzione

In un mondo in cui le conseguenze dell'evoluzione dell'uomo e delle sue attività minacciano costantemente gli ecosistemi marini, lo studio della distribuzione dei cetacei e della sua mutazione nel tempo può essere un elemento indicativo e determinante nella comprensione dell'alterazione di tali ecosistemi.

In tal senso, dati dettagliati su gruppi di cetacei in determinati ambienti sono di grande importanza. L'identificazione dei singoli individui permette grandi passi avanti nello studio del gruppo e dell'ambiente in cui vive. Metodi comuni in passato per l'identificazione di singoli individui animali prevedevano l'applicazione di elementi distintivi sul loro corpo. Tuttavia, tale applicazione poteva alterare il naturale corso della vita dell'animale, come anche il suo comportamento, creando un *bias* negli studi e problemi di natura etica [1]–[3].

L'incremento di interesse verso tecniche non invasive (quali la foto-identificazione mediante segni naturali unici sul corpo dell'individuo) è la conseguenza di quest'ultima considerazione.

Tra i cetacei, in particolare, vi è una specie notevolmente adatta a questo tipo di identificazione: il delfino di Risso, o grampo (*Grampus griseus*), figura 2.1 a pagina seguente.

Il delfino di Risso può avere una lunghezza compresa tra 2,5 e 4 metri ed un peso di 500-600kg. Il capo si presenta senza rostro ("becco"); la pinna dorsale, posta circa a metà del corpo è particolarmente alta e appuntita, e presenta una curva piuttosto accentuata.

La sua livrea unica è ciò che probabilmente lo distingue maggiormente da altri cetacei e lo rende, come già detto, particolarmente adatto all'identificazione: sebbene nei primi anni di età il colore sia scuro e uniforme, con il passare degli anni il corpo viene man mano ricoperto da numerosissimi ed estesi segni, graffi e cicatrici estremamente chiari; gli esemplari più anziani ne vantano così tanti da risultare praticamente bianchi.



Figura 2.1 *Grampus griseus*

Questi graffi sono causati sia dalle bocche dei calamari (la loro maggiore preda) che da interazioni sociali con altri individui. Ogni esemplare presenta graffi e cicatrici pressoché unici, paragonabili alle impronte digitali per l'uomo: tale caratteristica rende possibile l'identificazione di singoli individui.

Il delfino di Risso è una specie largamente diffusa, soprattutto in acque tropicali e temperate ad alta profondità. Nonostante la sua diffusione però, le informazioni e le stime sulla sua presenza, così come anche i dettagli della sua distribuzione sono scarsi e insufficienti, tanto che il suo rango attribuito dalla IUCN Red List of Threatened Species fino al 2007 è stato quello di "Data Deficient" [4].

Valutata dunque l'importanza dello sforzo di identificazione dei singoli esemplari di delfino di Risso all'interno dei loro ambienti e delle comunità che essi creano, ed evidenziata la

possibilità di identificarli attraverso un metodo non invasivo quale la foto-identificazione, il maggiore ostacolo rimaneva la realistica esecuzione di tale attività.

In effetti, la foto-identificazione di grandi quantità di dati è un'operazione dal costo (temporale e pratico) estremamente alto: ciò ha suggerito l'introduzione di tecniche automatizzate a supporto dell'esecuzione manuale.

Scopo di questa tesi è di apportare diversi importanti miglioramenti agli strumenti automatici a supporto della creazione di informazioni circa l'identificazione degli esemplari attraverso fotografie digitali acquisite in missioni al largo del golfo di Taranto, e di descrivere i notevoli benefici ottenuti.

L'attuale stato dell'arte per il riconoscimento dell'individuo sconosciuto può essere individuato in due tool differenti.

- Il primo (nonché storico e largamente citato in letteratura), di nome DARWIN [5], prevede l'utilizzo di tecniche di foto-identificazione assistite. Richiede quindi una indispensabile e continua interazione uomo-macchina, incentrata sull'analisi del bordo della pinna dorsale mediante tecniche di *digital image processing*.

Il processo di identificazione è costituito da più fasi: una prima fase permette la conversione di una immagine di pinna dorsale in un contorno confrontabile con altri, la fase seguente ne gestisce il salvataggio in un database, infine l'ultima si occupa del confronto dei contorni per la restituzione della lista ordinata (*matching*); il tutto è accessibile con l'utilizzo di una interfaccia grafica.

Il tool è semi-automatico, in quanto la prima fase richiede che venga indicato il punto di inizio della pinna e il punto di fine, così come, in alcuni casi, il tracciamento da parte dell'utente del bordo, in maniera manuale.

- Il secondo, SPIR (*Smart Photo Identification of the Risso's dolphin*) [6], si presenta invece come tool automatico. Grazie a degli algoritmi di *feature extraction* e di *feature matching*, SPIR permette l'identificazione di esemplari non noti basandosi su un dataset di esemplari precedentemente identificati, senza alcun intervento da parte dell'utente.

L'identificazione senza intervento dell'utente rende SPIR potenzialmente più adatto a processare grandi quantità di dati. Per questo motivo, è stato scelto come punto di partenza per il lavoro di tesi.

Diversi sono stati i miglioramenti apportati al programma originale (d'ora in avanti definito nel lavoro di tesi come *SPIR v1*), tutti focalizzati verso una più efficace ed efficiente elaborazione e classificazione dei dati:

- la creazione della funzionalità del ritaglio automatico
- la sostituzione della libreria per il calcolo delle feature in modo da migliorare l'usabilità, la facilità del setup iniziale sulle macchine e le prestazioni
- l'introduzione della capacità di utilizzare più immagini dello stesso modello per l'identificazione di un esemplare non noto
- la ristrutturazione dell'output del tool, che ora consente di accedere facilmente ai dati già identificati e riordinati per la verifica da parte dei biologi e permette di ottenere report riassuntivi sull'identificazione condotta

A queste modifiche, inoltre, si è aggiunto uno studio preliminare sul calcolo e la descrizione automatici del contorno della pinna dorsale, passaggio fondamentale per l'estensione dell'applicazione di SPIR ad altre specie di cetacei tra cui ad esempio il Tursiope (*Tursiops truncatus*).

Il risultato di questi miglioramenti è *SPIR v2*, un tool estremamente performante e applicabile a dataset di dimensioni molto elevate, utilizzato per svolgere gli esperimenti descritti nel capitolo 4, riguardanti l'identificazione automatica di delfini di Risso in 44 giornate di avvistamenti nel golfo di Taranto durante il corso dell'estate del 2019.



## 3 Metodologia

### 3.1 Metodologia manuale

Gli sforzi sostenuti nel corso del lavoro di tesi si sono concentrati maggiormente su attività del workflow particolarmente costose in termini di tempo ed impegno: il ritaglio e l'identificazione.

La figura 3.1 a pagina seguente descrive il workflow intrapreso in una situazione esclusivamente manuale e priva di supporti automatici o semiautomatici. Si tenga presente che la parola “modello” indica, qui e nel corso dell'intera tesi, un'immagine di un delfino di Risso già identificato e catalogato in precedenza.

Ogni missione di successo al largo si conclude generalmente con una certa quantità di foto di esemplari da identificare: per l'operatore ciò vuol dire recuperare le immagini da identificare dalla macchina fotografica e recuperare dagli archivi le immagini di delfini già identificati. A questo punto, foto per foto, si tenta di identificare tramite un confronto visivo i delfini di Risso presenti nelle foto ottenute in missione. Le foto vanno anche ritagliate per ottimizzare lo spazio e per poterle riutilizzare come modelli in seguito, aggiornando l'archivio degli esemplari noti ed inserendo eventuali nuovi delfini trovati. Al termine di queste operazioni è possibile stilare un eventuale report contenente informazioni di interesse da consegnare ai biologi, assieme eventualmente alle foto identificate e ritagliate.

## Flowchart per l'identificazione manuale degli esemplari



Figura 3.1

Flowchart per l'identificazione manuale degli esemplari.

Le attività contrassegnate dai colori caldi richiedono uno sforzo manuale di elaborazione delle immagini: nello specifico si è scelto di intervenire sui passaggi contrassegnati in arancione, ovvero i passaggi di identificazione e ritaglio, in quanto estremamente onerosi.

### 3.2 Metodologia automatica

Come già accennato nel capitolo 1, il software oggetto di questo lavoro di tesi è chiamato SPIR (acronimo di *Smart Photo Identification of the Risso's dolphin*). La versione originale di SPIR (SPIR v1) consente l'identificazione degli esemplari del delfino di Risso attraverso l'analisi e l'estrazione di informazioni circa alcune caratteristiche della pinna dorsale, quali i punti chiave identificativi e caratteristici generati dalla presenza delle cicatrici e dalla loro intersezione (altresì chiamati *feature*). In SPIR v2, risultato delle modifiche apportate nel corso di questo lavoro di tesi, si migliorano le attività di identificazione e si aggiunge anche la funzionalità di ritaglio automatico delle pinne dorsali. Si veda il workflow dell'identificazione automatica in figura 3.2 a pagina seguente.

## Flowchart per l'identificazione automatica degli esemplari

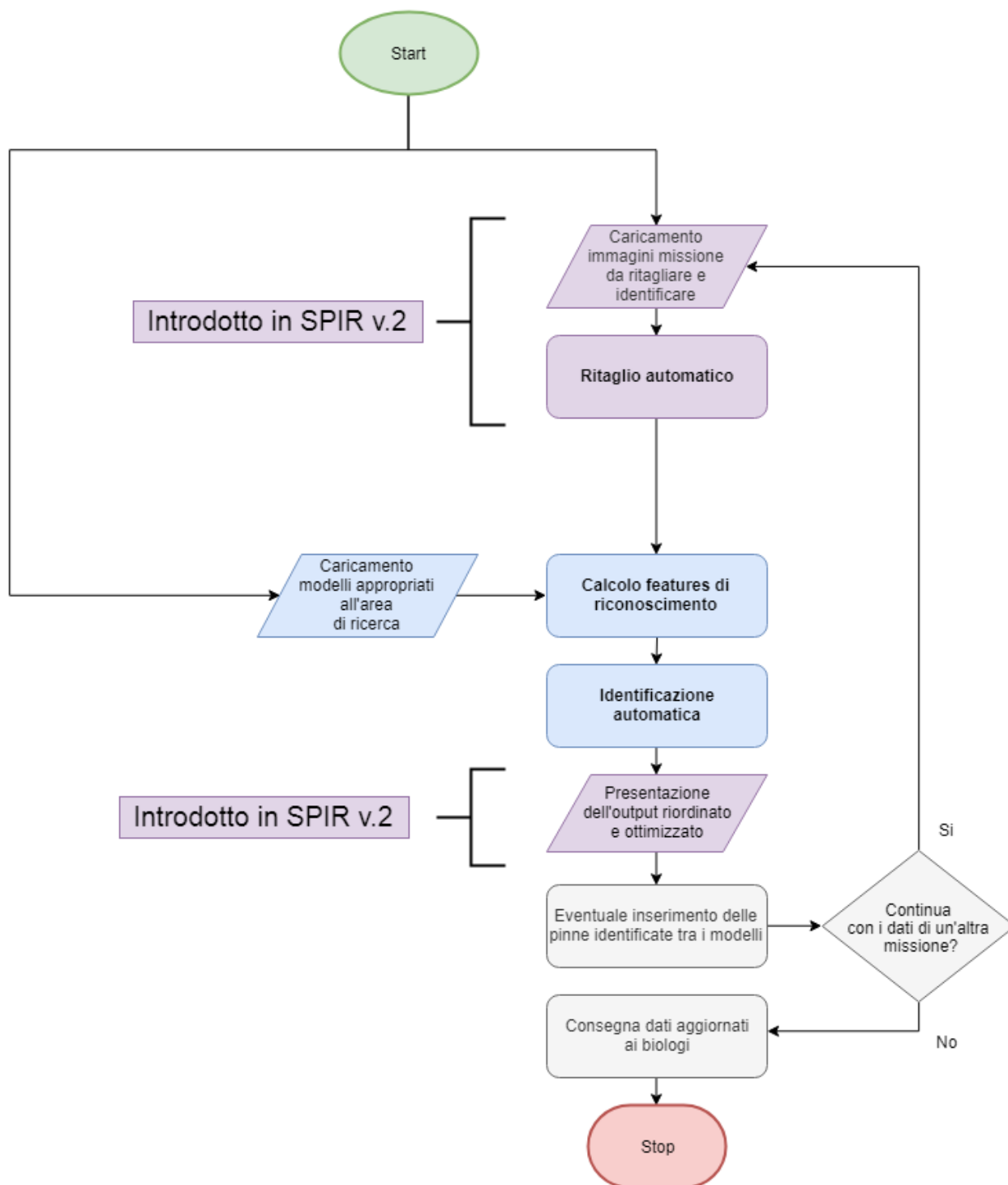


Figura 3.2

Flowchart per l'identificazione automatica degli esemplari

I riquadri in azzurro del flowchart rappresentano le attività svolte da SPIR v1

SPIR v2 apporta miglioramenti in quelle attività e introduce le attività rappresentate nei riquadri viola

In questo scenario automatico, l'utente sottopone al software direttamente le immagini ottenute durante la missione. In SPIR v1 queste immagini dovevano includere esclusivamente pinne dorsali già ritagliate dalle foto originali acquisite in missione; il ritaglio doveva essere fatto manualmente. SPIR v2 invece analizza le immagini e crea automaticamente dei ritagli, distinguendoli in tre categorie: pinne, non pinne e incerti. Ogni ritaglio valutato correttamente come pinna contiene un'unica pinna. Per maggiori informazioni sul ritaglio automatico si veda la sezione 3.2.1.

Una volta raccolti tutti i ritagli di pinne dorsali, questi possono essere usati come input della parte di identificazione automatica di SPIR. Analizzando quei ritagli, così come anche i ritagli delle pinne già identificate, vengono estratte le feature, ovvero i tratti univoci necessari all'identificazione dell'individuo. Questi tratti vengono poi usati nel passaggio seguente compiuto sempre dal programma in modo automatico: l'identificazione vera e propria. Per ogni individuo non noto viene effettuato un confronto con quelli noti, attraverso una funzione di *matching*, che ricerca quello noto più simile. Se questo match rispetta alcuni parametri di controllo, viene considerato valido ed il ritaglio viene identificato con successo.

Anche per l'identificazione sono stati introdotti miglioramenti: la modifica più importante tra SPIR v1 e SPIR v2 è la possibilità da parte di SPIR v2 di utilizzare più modelli, ovvero più immagini dello stesso delfino noto. Al contrario, in SPIR v1, ogni modello doveva essere identificato da una sola immagine. Maggiori dettagli sul calcolo delle feature e sul processo di confronto e match possono essere trovati rispettivamente nella sezione 3.2.2 e 3.2.3.

SPIR v1 era sviluppato in MATLAB 2015a, utilizzando il Computer Vision System Toolbox. Tuttavia, la libreria OpenCV 2.4.10 utilizzata per le funzionalità di estrazione delle feature era scritta in C e C++ e richiedeva un'ulteriore libreria di supporto, mexopencv 2.4, nonché il compilatore incluso nel pacchetto Microsoft Visual C++ 2013. Tutto questo rendeva particolarmente complicata la procedura di setup e influiva sulla portabilità del sistema.

SPIR v2 risolve questo tipo di problema. Il software è stato sviluppato in MATLAB 2019a, utilizzando il Computer Vision System Toolbox e Deep Learning toolbox. Per le funzionalità

di estrazione delle feature viene ora utilizzata la libreria VLfeat 0.9.21, scritta anch'essa utilizzando MATLAB. Non vi è più alcuna necessità di effettuare un setup o operazioni preliminari.

### **3.2.1 Il ritaglio automatico**

Il lavoro descritto in questa sezione è, come già accennato, di grande importanza per tutti i passaggi seguenti, ed è stato svolto in collaborazione con un collega laureando del Politecnico di Bari, Gianvito Losapio, che ha svolto la sua tesi approfondendolo ulteriormente e presentando i risultati degli esperimenti. In questo lavoro di tesi, invece, si procede a considerarlo come uno dei passaggi all'interno di un processo automatico più grande e omnicomprensivo.

Per poter eseguire l'estrazione delle feature e l'identificazione di nuovi esemplari in maniera efficiente, si è reso necessario il ritaglio delle pinne dorsali dalle immagini ottenute nelle missioni. Questo ha naturalmente richiesto un riconoscimento automatico delle pinne all'interno delle varie immagini. La funzionalità implementata segue tre passaggi:

- La segmentazione dell'immagine basata sui colori
- Il filtraggio delle regioni individuate e il ritaglio adattivo sulla base di caratteristiche geometriche
- La classificazione binaria di ogni ritaglio: pinna/no pinna

Nelle sezioni successive si descrivono i singoli passaggi.

#### ***3.2.1.1 Segmentazione basata sui colori***

Per rendere possibile la divisione iniziale delle potenziali pinne dal background è stato necessario convertire ogni immagine in CIE  $L^*a^*b^*$  e applicare una binarizzazione tramite la soglia secondo il metodo di Otsu. Di seguito i dettagli.

### *Conversione in CIE $L^*a^*b^*$*

L'immagine ottenuta dalla macchina fotografica utilizza uno specifico spazio colore (di solito l'sRGB). Ciò significa che le informazioni sul colore di ogni singolo pixel dell'immagine vengono fornite da tre valori, che rappresentano la quantità di rosso, di verde e di blu.

Per poterla processare con la Soglia di Otsu, nel passaggio successivo, è necessario prima effettuare una conversione nello spazio di colore CIE  $L^*a^*b^*$ . In questo spazio di colore le informazioni sono comunque dettate da tre valori ( $L^*$ ,  $a^*$  e  $b^*$ ), ma esse rappresentano qualcosa di diverso. Il valore  $L^*$  rappresenta la luminanza (variazione di luminosità), mentre  $a^*$  (scala verde-rosso) e  $b^*$  (scala blu-giallo) rappresentano la cromaticità (variazione di colore). È necessaria fare questa conversione perché tale spazio di colore riesce a descrivere meglio la differenza tra il foreground dell'immagine (la pinna) e il background (l'acqua).

### *Soglia di Otsu*

Lo scopo di questo passaggio è di distinguere l'area del ritaglio che contiene una pinna dall'area che contiene invece l'acqua del mare. È necessario compiere questo passaggio per evitare di rilevare features in zone esterne alla pinna, collegate quindi alle onde e agli schizzi.

L'operazione di sogliatura, come suggerisce lo stesso nome, non è altro che la distinzione dei pixel dell'immagine in classi, basata sul superamento da parte dei valori con cui i pixel sono descritti, di una determinata soglia. Nel caso le classi da individuare siano solo due (pinna e non pinna, appunto), un buon metodo per determinare la soglia è quello di Otsu, poiché questo metodo massimizza la varianza interclasse [7].

### **Calcolo della soglia**

Consideriamo un'immagine avente dimensione  $d = M \times N$  con  $L$  livelli distinti di intensità e sia  $n_i$  il numero di pixel di intensità  $i$ . Il suo istogramma normalizzato ha componenti  $p_i = \frac{n_i}{d}$ . Supponiamo di selezionare una soglia  $T(k) = k$  con  $0 < k < L - 1$ , e di dividere in base ad essa l'immagine in due classi,  $C_1$  (contenente tutti i pixel con intensità compresa nell'intervallo  $[0, k]$ ) e  $C_2$  (contenente tutti i pixel con intensità

compresa nell'intervallo  $[k + 1, L - 1]$ ). Con tale soglia la probabilità che un pixel sia assegnato alla classe  $C_1$  è:

$$P_1(k) = \sum_{i=0}^k p_i$$

Per la classe  $C_2$  si ha:

$$P_2(k) = \sum_{i=k+1}^{L-1} 1 - P_1(k)$$

Mentre il valore medio di intensità dei pixel appartenenti alla classe  $C_1$  è:

$$m_1(k) = \sum_{i=0}^k iP(i/C_1) = \sum_{i=0}^k iP \frac{P(C_1/i)P(i)}{P(C_1)} = \frac{1}{P_1(k)} \sum_{i=0}^k ip_i$$

In maniera simile, si ricava per la seconda classe:

$$m_2(k) = \sum_{i=k+1}^{L-1} iP(i/C_2) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} ip_i$$

Si definisce la media cumulativa fino al livello  $k$ :

$$m(k) = \sum_{i=0}^k ip_i$$

La media delle intensità dell'intera immagine è data da:

$$m_G(k) = \sum_{i=0}^{L-1} ip_i$$

Per valutare la bontà della soglia  $k$  si utilizza la metrica normalizzata:

$$\eta = \frac{\sigma_B^2}{\sigma_G^2}$$

Con  $\sigma_G^2$  varianza globale dei pixel dell'immagine:

$$\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 p_i$$

e  $\sigma_B^2$  varianza interclasse:

$$\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2$$



La varianza interclasse può essere riscritta come:

$$\sigma_B^2 = P_1 P_2 (m_1 - m_2)^2 = \frac{(m_G P_1)^2}{P_1 (1 - P_1)}$$

Più distanti sono le due medie, più la varianza interclasse è elevata. Questa formulazione della varianza interclasse permette una più efficiente implementazione. La soglia ottima viene dunque calcolata:

$$\sigma_B^2(k^*) = \max_{0 \leq k \leq L-1} \sigma_B^2(k)$$

Allo scopo di distinguere la pinna dall'acqua, la soglia di Otsu è stata calcolata sul canale  $L^*$ , che descrive una variazione di luminosità spesso presente tra pinna e sfondo (specie quando questa è in controluce) e sul canale  $b^*$  che rappresentava al meglio la distinzione (variazione di colore blu-giallo).

In figura 3.3 a pagina seguente viene mostrato il risultato ottenuto mediante l'utilizzo della soglia.



**Figura 3.3**  
In alto: l'immagine originale  
In basso: l'immagine convertita in CIE L\*a\*b\* e sottoposta a soglia di Otsu

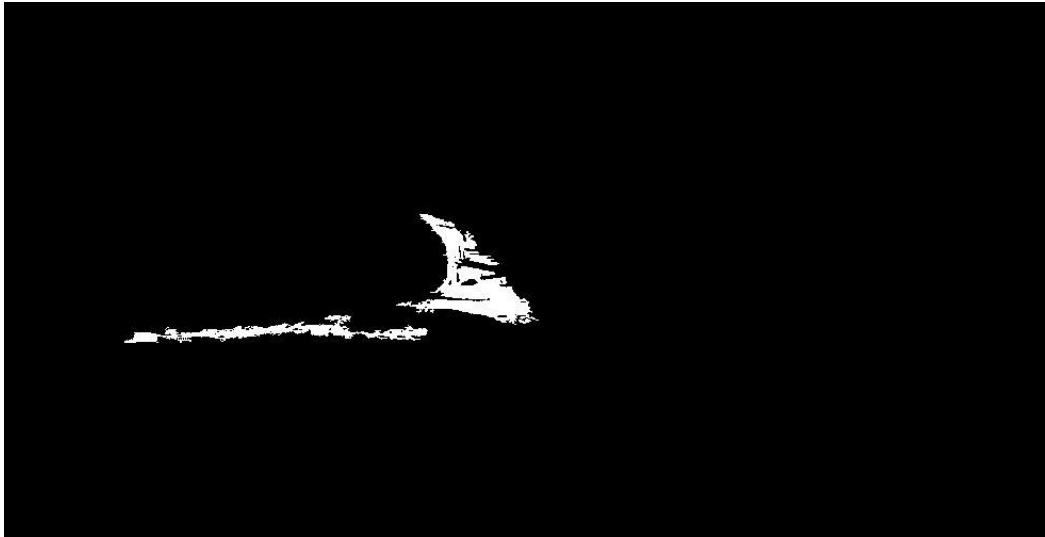
L'applicazione di questa soglia sull'intera immagine porta all'estrazione di una serie di aree di pixel che superano la soglia. Queste aree potenzialmente individuano le pinne all'interno dell'immagine, tuttavia sono necessari ulteriori passaggi per ridurre eventuali errori prima del ritaglio.

#### ***3.2.1.2 Filtraggio e ritaglio adattivo***

Il filtraggio si occupa di ridurre eventuali errori effettuati dalla soglia nel riconoscimento delle pinne, prima del ritaglio effettivo. Il filtro di assicura che ogni area di pixel identificata dalla soglia di Otsu rispetti le seguenti caratteristiche:

- Sia tra le prime 15 in ordine decrescente di dimensione, in numero di pixel
- Il numero di pixel che la costituisce sia compreso tra i 1600 e i 40000
- Il rapporto tra il numero di pixel che la costituisce e il numero di pixel del rettangolo minimo che la contiene sia tra  $-\infty$  e 0.55

Se l'area non rispetta queste caratteristiche, il filtro la elimina dalla selezione (figura 3.4)



**Figura 3.4**  
L'immagine nella figura precedente dopo essere stata sottoposta a filtraggio

Le aree di pixel contigue rimanenti dopo il filtraggio vengono sottoposte a ritaglio adattivo, basato su specifici punti geometrici.

I passaggi che permettono il ritaglio adattivo per ogni area contigua ottenuta sono i seguenti:

- Si sottopone la regione binaria al riempimento dei cosiddetti *holes*. I pixel non inclusi nell'area che però vengono circoscritti da essa, vengono inclusi.
- Si individuano quattro punti di interesse: il più in alto  $A$ , il punto medio tra il centroide dell'area e il punto  $A$ , chiamato  $B$ , e i punti di estrema sinistra ( $C$ ) e destra ( $D$ ) all'altezza del punto medio  $B$

- Si individua un ulteriore punto  $E$  di coordinate  $(x, y)$ , costituito dal punto medio tra il punto  $A$  e il punto più lontano da  $A$  (ovvero  $C$  oppure  $D$  a seconda dell'area). Si veda figura 3.5.
- Si individua un rettangolo che abbia centro in  $E$  ed uno dei vertici in  $A$ . L'altezza di questo rettangolo viene chiamata  $h$ , la base viene chiamata  $w$  (figura 3.6 a pagina seguente).
- Si individua un punto  $F$ . Questo punto ha coordinate  $(x - w, y - 0.65 \times h)$  ed è il vertice superiore sinistro del ritaglio finale. Il ritaglio finale ha la base definita dal valore  $2w + \frac{w}{2}$  e l'altezza definita da  $2h + \frac{h}{2}$  (figura 3.7 a pagina seguente).

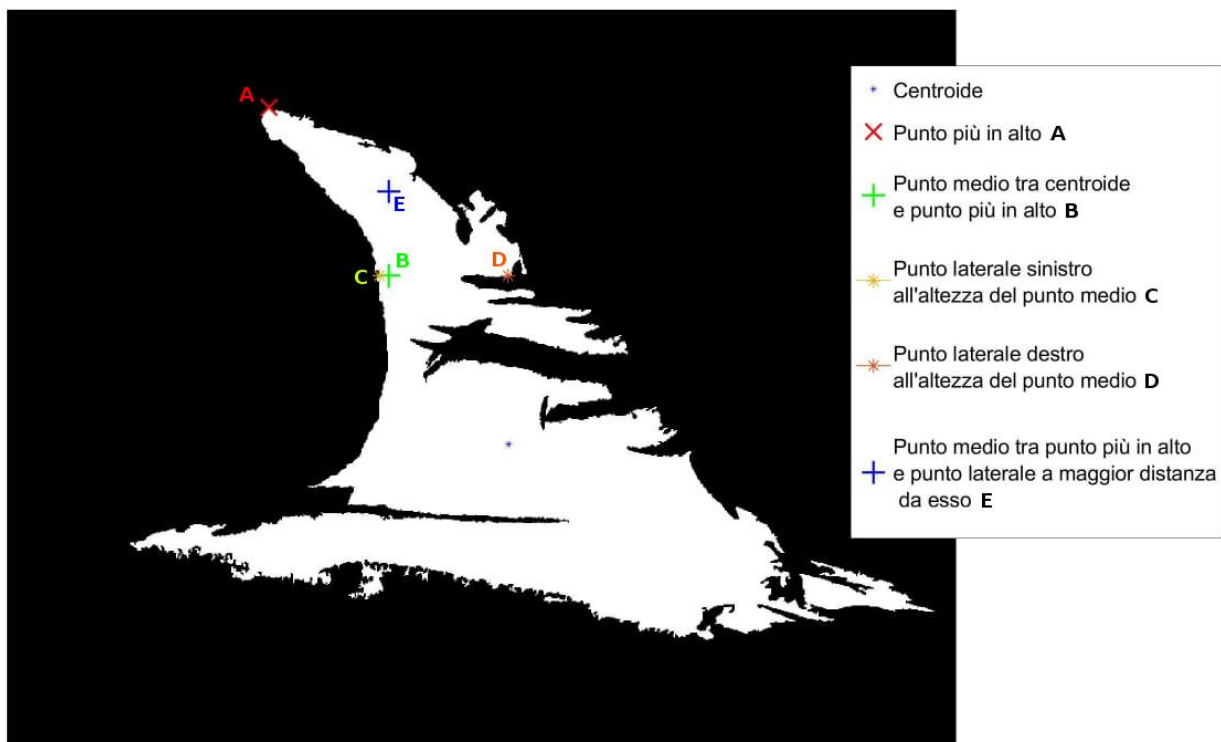


Figura 3.5  
Individuazione dei punti A, B, C, D ed E

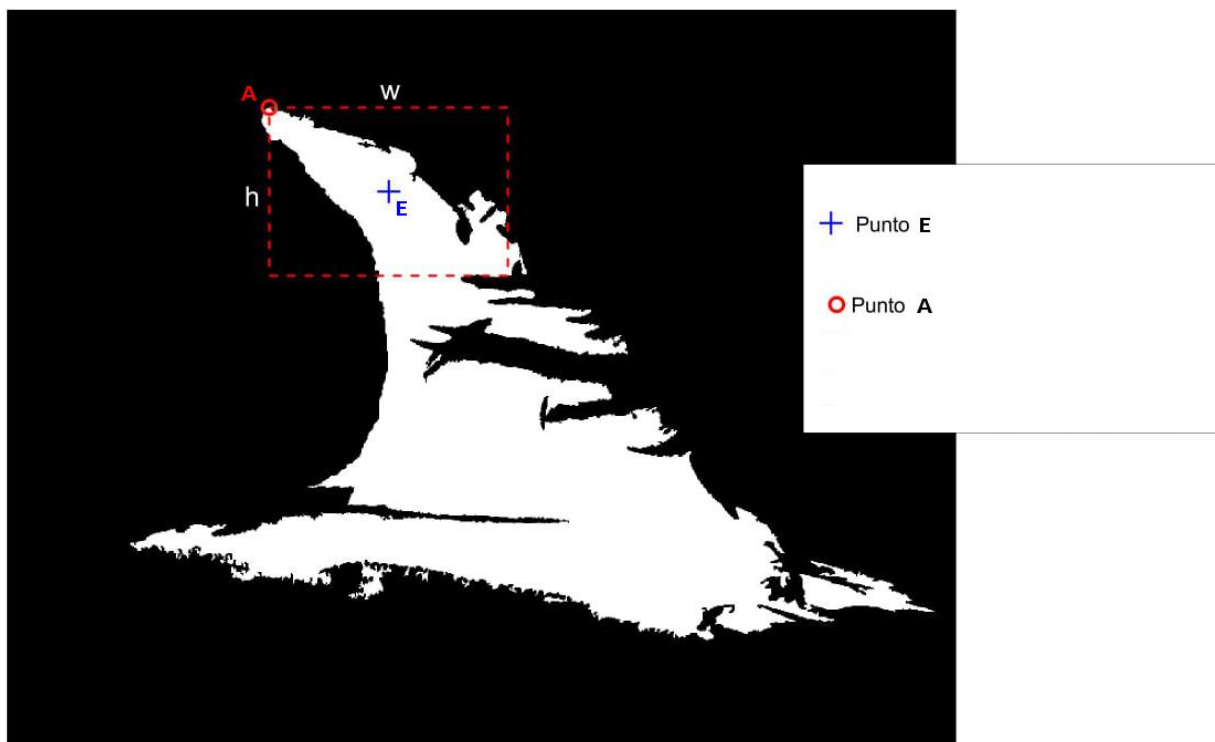


Figura 3.6  
Identificazione del rettangolo di altezza  $h$  e base  $w$

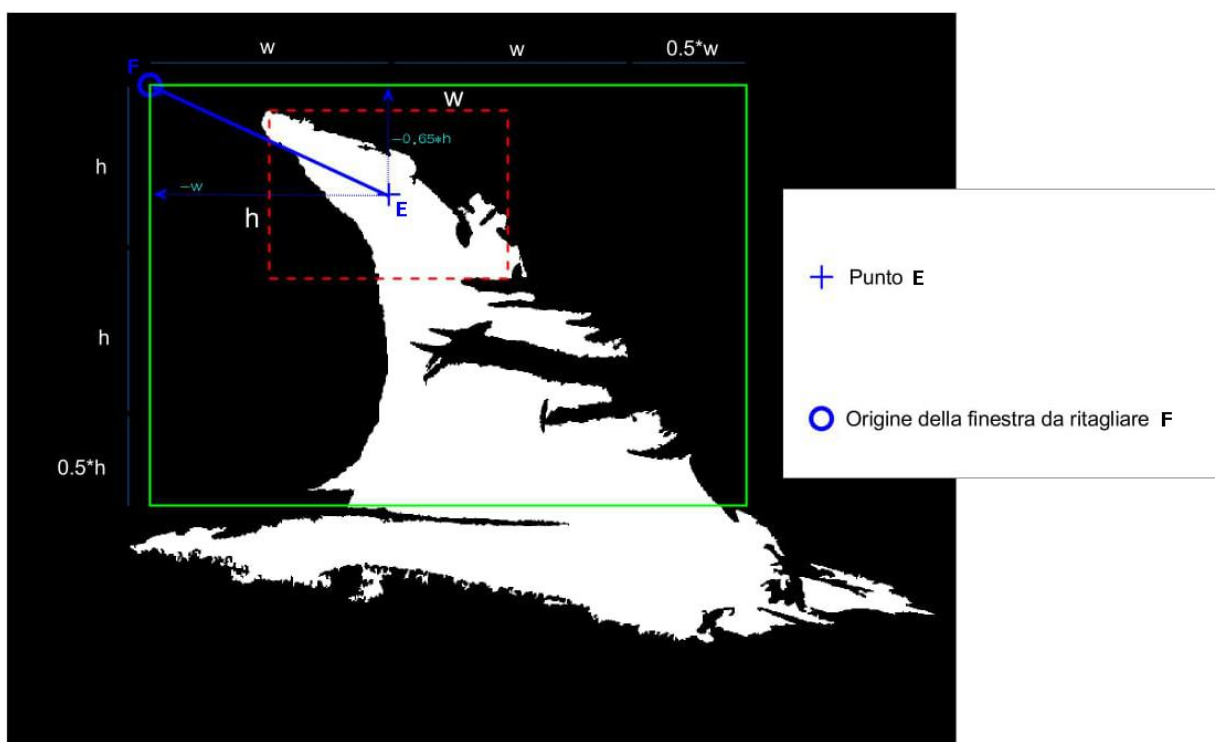


Figura 3.7  
Creazione della finestra di ritaglio basata sulle coordinate del punto  $F$

Sebbene macchinoso, questo metodo riesce ad estrarre alcuni ritagli che contengono già delle pinne pronte all'identificazione.

Non tutti i ritagli ottenuti però contengono delle pinne: alcune delle aree filtrate contengono solo particolari artefatti nella foto, oggetti estranei quali barche e altra fauna, o semplicemente, il riflesso delle pinne nell'acqua. Per ottenere soltanto effettivi ritagli di pinne, i ritagli vengono processati nell'ultimo passaggio, ovvero la classificazione binaria.

### ***3.2.1.3 Classificazione binaria pinna/non pinna***

Il classificatore utilizzato in questo procedimento è costituito da 5 differenti reti convoluzionali, addestrate secondo una procedura di 5-Cross Fold Validation sul medesimo dataset. Queste reti eseguono la propria classificazione binaria per ciascun ritaglio. L'esito definitivo è stabilito secondo la seguente strategia di tipo major voting:

- se le classificazioni Pinna sono maggiori o uguali a 4, allora l'esito definitivo è Pinna
- se le classificazioni Pinna sono pari a 2 o 3, allora l'esito definitivo è Incerto
- se le classificazioni Pinna sono minori o uguali a 1, allora l'esito definitivo è No Pinna.

Sulla base della classificazione finale, i ritagli vengono salvati in corrispondenti directory.

### **3.2.2 L'estrazione delle features**

Una volta ottenuti i ritagli delle pinne dorsali, il passaggio successivo del software è il calcolo delle features, ovvero dei punti chiave che permettono di descrivere ogni pinna in maniera univoca. Queste features vengono calcolate sia per le immagini di delfini di Risso noti che per quelle di delfini di Risso non noti: in seguito verranno confrontate per ottenere una identificazione.

Poiché questa operazione può rivelarsi costosa in termini di tempo e di calcolo, il software è stato programmato per salvare le features calcolate in un file nella stessa cartella delle immagini in input: può caricarle rapidamente senza ricalcolarle nel caso venga eseguito nuovamente con le stesse immagini.

### 3.2.2.1 Operazioni preliminari

Prima del calcolo delle features, ogni immagine viene preparata, eseguendo le seguenti operazioni.

#### *Conversione in CIE $L^*a^*b^*$*

Questa operazione è eseguita in modo analogo a quello presentato nella sezione 3.2.1.1, nel paragrafo “Conversione in CIE  $L^*a^*b^*$ ”.

#### *Soglia di Otsu*

Questa operazione è eseguita in modo analogo a quello presentato nella sezione 3.2.1.1, nel paragrafo “Soglia di Otsu”. Tuttavia in questo caso è sufficiente calcolare la soglia unicamente sul canale  $b^*$ .

#### *Operazioni morfologiche*

La distinzione tra pinna e acqua effettuata dalla soglia di Otsu non è perfetta: essa può restituire più di una singola area distinta e può includere del rumore al suo interno quali onde o schizzi. Per pulire maggiormente il risultato, vengono utilizzate delle funzioni di MATLAB quali *bwmorph* [8] (per effettuare la dilatazione e l'erosione dell'immagine ed in questo modo definire in maniera più pulita le aree ottenute con la soglia di Otsu) e *regionprops* [9], che permette di selezionare e ritagliare l'immagine considerando esclusivamente l'area più grande ottenuta con la soglia (escludendo quindi schizzi, onde o comunque elementi più piccoli della pinna).

Queste operazioni di preparazione non permettono di ottenere il contorno perfetto della pinna, né di escludere completamente schizzi o onde, ma riducono il rumore nel calcolo delle features, evitando che vengano inclusi alcuni degli elementi di disturbo. Il risultato può essere visionato nella sezione 3.2.2.2 nella figura 3.11, dove la maschera realizzata in questi passaggi viene utilizzata.

### 3.2.2.2 L'estrazione

Una volta preparata l'immagine, vengono calcolate le features.

I punti di interesse sono individuati dal detector in locazioni particolari dell'immagine come gli angoli, incroci a T o delle macchie (blobs). Una delle proprietà più importanti del detector è la ripetibilità, ovvero data una immagine o foto di un soggetto il sistema trova gli stessi punti indipendentemente dalle trasformazioni geometriche o angolazioni di scatto. In seguito, viene descritto l'intorno del punto tramite un vettore. Questo descrittore deve essere distintivo e, allo stesso tempo, robusto al rumore e deformazioni geometriche e fotografiche.

La tecnica scelta per il calcolo è quella SIFT [10]. Queste feature sono invarianti alla scala e rotazione dell'immagine ed, in parte, a trasformazioni affini, cambio del punto di vista (in un piano 3D), aggiunta di disturbi e cambio di illuminazione. Sono anche molto distintive, quindi possono essere correttamente abbinate all'interno di un grande database di features ricavato da molte immagini.

### *Detector dei punti di interesse*

Per individuare i punti di interesse si analizza l'immagine in diverse scale di grandezza alla ricerca di punti candidati. In seguito si analizzano più a fondo i punti candidati per determinare se sia un valido punto di interesse e per aumentarne la precisione.

La ricerca dei punti candidati si basa sulle proprietà della Laplaciana della Gaussiana (indicata come  $\sigma \nabla^2 G$ , dove  $G$  è l'operatore di convoluzione tra l'immagine e una curva Gaussiana). In particolare i punti di massimo e minimo producono dei punti di interesse stabili. La Laplaciana della Gaussiana viene approssimata con delle differenze della Gaussiana (DOG). Sia  $G(x, y, \sigma)$  un filtro Gaussiano centrato nel punto dell'immagine  $(x, y)$  e con scala  $\sigma$ , DOG viene calcolata convolvendo l'immagine,  $I(x, y)$ , con la differenza di due filtri gaussiani centrati nel punto dell'immagine  $(x, y)$  con scala differente,  $\sigma$  e  $k\sigma$ . Quindi risulta che:

$$DOG(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y)$$

Dove  $*$  è l'operatore di convoluzione e  $DOG(x, y, \sigma)$  è la differenza delle Gaussiane nel punto  $(x, y)$  e con scala  $\sigma$ .



Se consideriamo:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

Allora

$$DOG(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma)$$

Essendo l'operatore  $*$  un operatore lineare.  $L(x, y, \sigma)$  è l'immagine gaussiana di scala  $\sigma$ .

La funzione DOG, quindi, può essere calcolata in ogni punto come differenze due immagini Gaussiane con scale diverse. Ciò viene utilizzato per un calcolo efficiente della funzione a diverse scale. Le scale sono organizzate in ottave, dove ogni ottava copre un fattore di scala pari a due. Per ogni ottava viene calcolata  $L(x, y, \sigma)$  in punti campionati dell'immagine e a diverse scale, incrementate di un fattore pari a  $k$ . Le immagini Gaussiane adiacenti vengono sottratte tra loro per ricavare DOG. Una volta completata un'ottava, si campiona l'immagine Gaussiana con scala doppia rispetto al valore di scala iniziale dell'ottava precedente, prendendo un punto ogni due pixel in verticale e orizzontale. Dall'immagine campionata si ripete il processo fino al completamento di tutte le ottave necessarie. In figura 3.8 a pagina seguente è mostrato un esempio dell'utilizzo delle immagini gaussiane per ricavare le differenze della gaussiana in diverse ottave.

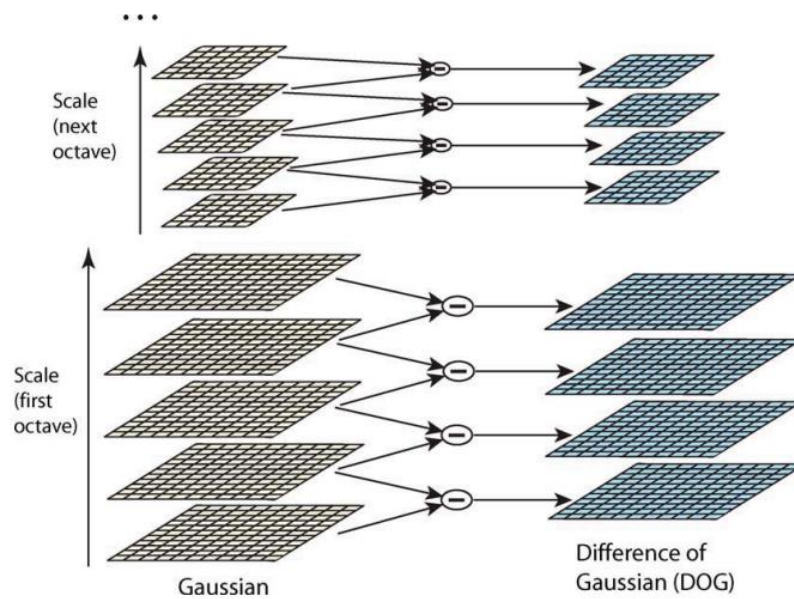


Figura 3.8

A sinistra: le immagini gaussiane calcolate a diverse scale per ogni ottava.  
A destra: la differenza tra quelle adiacenti per ricavare la differenza delle gaussiane (DOG).

### Individuazione dei punti di interesse candidati

Per ricavare i punti di massimo e minimo all'interno di  $DOG(x, y, \sigma)$  a diverse scale, ogni punto è comparato con il suo intorno di 8 elementi nella stessa scala e con i 9 punti corrispondenti delle scale adiacenti come in figura 3.9. È selezionato solo se è il più grande, o il più piccolo, tra tutti loro.

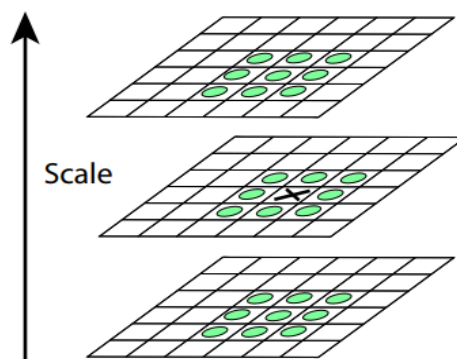


Figura 3.9

La croce nel livello intermedio individua il punto candidato che si sta analizzando, i cerchi individuano i punti che formano il suo intorno (8 nel livello corrente e 9 per ogni scala adiacente).

### **Miglioramento dell'insieme dei punti di interesse candidati individuati**

Per evitare che vengano utilizzati punti di interesse non localizzati correttamente oppure poco rappresentativi, tutti i punti candidati vengono analizzati e rigettati in caso abbiano un contrasto basso oppure siano localizzati sui bordi e non siano abbastanza stabili.

Per ogni punto viene determinata la posizione interpolata del punto di interesse analizzando una approssimazione della derivata della DOG. Una volta ricavato il punto interpolato viene calcolato il valore della DOG nel nuovo punto per analizzarne il contrasto: se è inferiore ad una certa soglia viene rigettato.

Dato che la differenza delle Gaussiane ha risposte forti sui bordi, vengono eliminati i punti che hanno una curvatura principale attraverso il bordo molto più elevata rispetto a quella in direzione perpendicolare. Per confrontare il rapporto tra le due curvature viene utilizzata la matrice Hessiana di dimensione  $2 \times 2$ . Se il rapporto tra il quadrato della traccia e il determinante della matrice è inferiore ad una soglia allora il punto viene scartato.

### *Descrittore dei punti di interesse*

Il descrittore è ciò che rappresenta la distribuzione di intensità di un intorno del punto di interesse.

### **Orientamento**

Assegnando un orientamento consistente ad ogni punto di interesse, il descrittore può essere rappresentato in relazione a questo orientamento e quindi essere invariante alla rotazione. Viene selezionata l'immagine Gaussiana (calcolata in precedenza) con la scala più simile a quella del punto di interesse. Per ogni campione dell'immagine,  $L(x, y)$ , calcolato in questa scala, ricaviamo le informazioni sul suo gradiente (vettore con orientamento  $\theta(x, y)$  e lunghezza  $m(x, y)$ ).

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2}$$

$$\theta(x, y) = \tan^{-1}\left(\frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)}\right)$$

Viene costruito un istogramma degli orientamenti con 36 contenitori che coprono 360°. Ogni punto campionato viene aggiunto all'istogramma e pesato in base a  $m$  e il valore di una Gaussiana nella posizione del punto campionato, centrata nel punto di interesse con scala pari a 1.5 volte la scala in cui è stato trovato il punto di interesse. Picchi nell'istogramma rappresentano le direzioni predominanti. Il picco maggiore determina l'orientamento. Se il secondo picco maggiore ha un valore superiore all'80% del picco massimo allora viene creato un nuovo punto di interesse con coordinate e scala pari a quelle del punto che lo ha generato, ma con il nuovo orientamento.

### **Vettore descrittore**

Le operazioni precedenti individuano un'area di interesse che è centrata nelle coordinate del punto di interesse, ha una scala e un orientamento. Il vettore descrittore deve descrivere questa area in modo distintivo, invariante all'illuminazione e il punto di vista e robusto al rumore.

Vengono ricavati i gradienti in dei punti campionati intorno al punto di interesse all'interno della scala più simile a quella del punto di interesse. Per essere invariante alla rotazione, l'orientamento dei gradienti viene normalizzato rispetto all'orientamento del punto di interesse. La lunghezza dei vettori gradiente è pesata con una funzione Gaussiana con  $\sigma$  pari alla scala del punto di interesse.

Viene creato un istogramma degli orientamenti, come in precedenza, con 8 contenitori utilizzando 4x4 punti campionati (e i relativi gradienti). L'istogramma rappresenta l'orientamento con 8 direzioni, la cui intensità è influenzata dai gradienti dei punti campionati (figura 3.10 a pagina seguente). Ogni gradiente può influenzare fino a 4 direzioni in base alla sua lunghezza e orientamento.

In un esempio reale si può utilizzare un'area di 4x4 in cui campionare 4x4 punti. Da ogni area ricaviamo i valori dell'istogramma nelle 8 direzioni, risultando quindi in un vettore che descrive l'area in 128 valori.

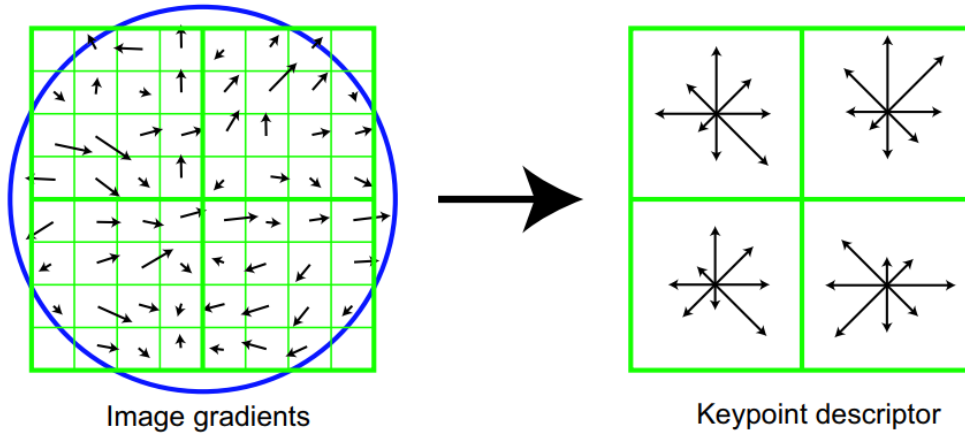


Figura 3.10

A sinistra: 2x2 aree in cui sono campionati 4x4 punti da cui si ricava il gradiente.  
A destra: i gradienti convertiti in istogrammi a 8 direzioni.

*Il risultato del calcolo delle feature e l'eliminazione delle feature all'esterno della pinna*

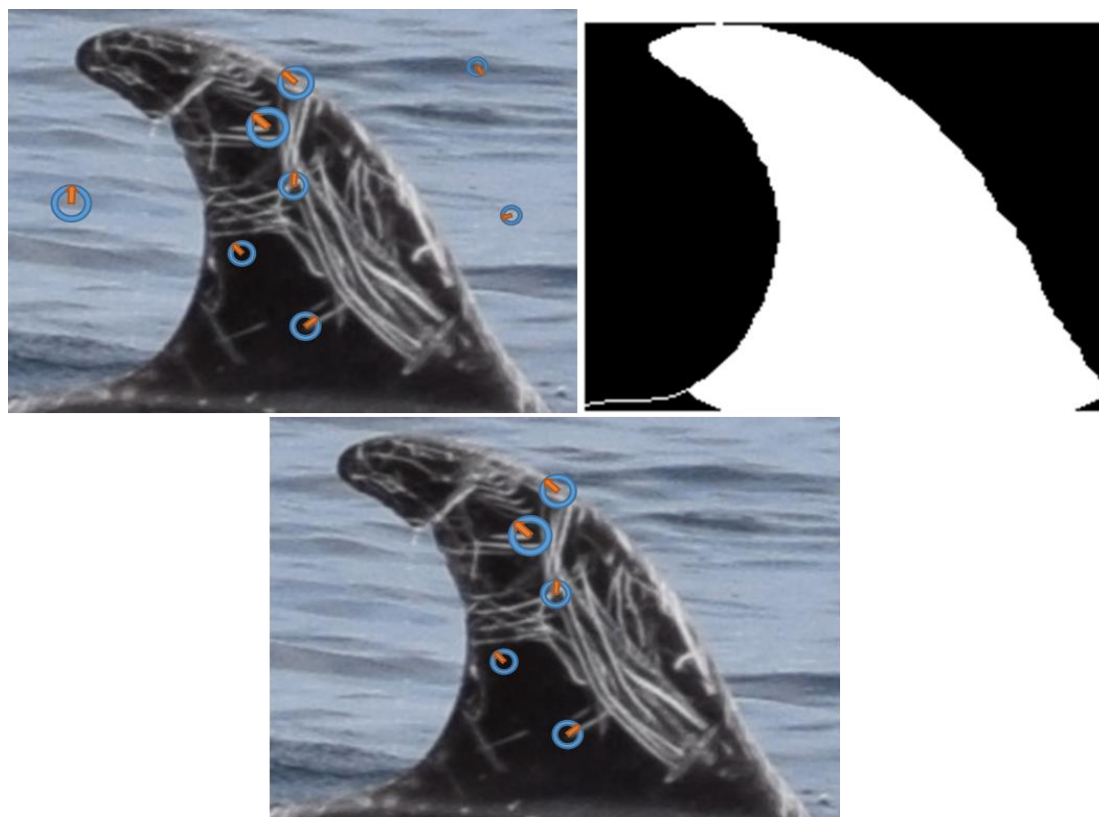
Il risultato finale del procedimento è un insieme di feature  $f$ , ciascuna così strutturata:

$$f = \begin{bmatrix} pt = (x, y) \\ ori \\ s \\ desc \end{bmatrix}$$

$pt$  indica le coordinate del punto di interesse indicato dalla feature;  $desc$  è il vettore che descrive l'intorno del punto di interesse;  $ori$  indica l'orientamento dell'intorno descritto dal descrittore e  $s$  la sua scala.

Le feature estratte nei passaggi precedenti e la maschera ottenuta dalla divisione tra pinna e background realizzata nella sezione 3.2.2.1 sono l'input della funzione di filtraggio. Questa funzione scarta tutte le feature la cui posizione  $pt$  è all'esterno della pinna. Quindi, per ogni feature, ponendo  $pt = (x, y)$  e  $Mask$  la maschera dell'immagine sotto forma di matrice che rappresenta con 1 i pixel appartenenti all'area della pinna e con 0 i pixel al di fuori dell'area della pinna, se  $Mask(x, y)$  è uguale a 0, allora la feature viene scartata.

La figura 3.11 presenta un esempio grafico dell'input e dell'output di quest'ultima funzione.



**Figura 3.11**

**In alto: l'input assegnato alla funzione di filtraggio. Le feature trovate dal feature extractor sono rappresentate dai cerchi celesti, mentre la maschera divide la pinna dal mare.  
In basso: l'output restituito dalla funzione di filtraggio. Le feature al di fuori della pinna sono state rimosse.**

Al termine di quest'ultima operazione di filtraggio, si procede al calcolo delle feature della prossima immagine. Una volta calcolate tutte le feature per tutte le immagini, queste vengono salvate e si procede all'identificazione del delfino sconosciuto mediante il loro confronto.

SPIR v2 tiene conto anche della possibilità che il feature extractor non riesca a trovare alcun punto di interesse in un'immagine: in quel caso il software è addestrato a scartare automaticamente l'immagine e inserirla in una specifica directory di output.

### 3.2.3 La foto-identificazione automatica

La fase di identificazione prevede il confronto di ogni ritaglio di pinna sconosciuta con tutti i ritagli di pinna nota (modelli), in modo da restituire l'identità predetta di ogni esemplare sconosciuto presentato.

Il confronto viene eseguito attraverso le feature SIFT calcolate nella sezione precedente.

Consideriamo l'insieme di tutti i ritagli dei modelli  $M = (m_1 \dots m_n)$ , con  $n$  numero di ritagli di tutti i modelli, e consideriamo l'insieme di tutti i ritagli di pinne sconosciute  $S = (s_1 \dots s_k)$ , con  $k$  numero di ritagli di pinne sconosciute.

Per ogni ritaglio di modello  $m_i$  e per ogni ritaglio sconosciuto  $s_j$  vengono considerati due insiemi:

- $f_{Ai}$ , ovvero l'insieme delle feature estratte per il ritaglio di modello  $m_i$
- $f_{Bj}$ , ovvero l'insieme delle feature estratte per il ritaglio sconosciuto  $s_j$

Questi insiemi vengono confrontati e associati tramite la funzione *matchFeatures* (sezione 3.2.3.1). Il risultato per ogni coppia  $m_i$  e  $s_j$  è l'insieme  $matches_{ij}$ , i cui elementi sono coppie di feature. In tal modo l'insieme  $matches_{ij}$  rappresenta quindi le associazioni tra l'insieme  $f_{Ai}$  e l'insieme  $f_{Bj}$ .

In seguito, le coppie dell'insieme  $matches_{ij}$  vengono sottoposte a operazioni di filtraggio (sezione 3.2.3.2), in modo da escludere alcune associazioni non corrette.

Il risultato di queste operazioni di filtraggio è dato da  $filtered\_matches_{ij}$ .

Una volta che ogni ritaglio di pinna sconosciuta viene confrontato con ogni ritaglio di modello, si ottengono  $n \times k$  insiemi  $filtered\_matches$ . Questi insiemi vengono utilizzati per l'operazione conclusiva dell'identificazione, ovvero la scelta del modello migliore (sezione 3.2.3.3).

In SPIR v2 si introduce anche la risoluzione delle ambiguità che si generano nel caso vengano utilizzati più ritagli di pinne di uno stesso esemplare noto. Il software trova più

immagini che corrispondono in egual maniera al ritaglio di pinna sconosciuta e gestisce questo caso secondo un algoritmo definito in sezione 3.2.3.4.

### 3.2.3.1 *matchFeatures*

Presi due insiemi di feature,  $f_A$  e  $f_B$  così descritti

$$f_A = \{f_A(1), f_A(2), f_A(3), \dots, f_A(N)\}$$

$$f_B = \{f_B(1), f_B(2), f_B(3), \dots, f_B(M)\}$$

con  $N$  cardinalità del primo insieme (numero di feature dell'immagine da identificare) e  $M$  cardinalità del secondo insieme (numero di feature dell'immagine modello), i generici  $f_A(i)$  e  $f_B(j)$  sono composti da quattro elementi: la posizione del punto di interesse, l'orientamento predominante, la scala del suo intorno ed infine un vettore descrittore che descrive la distribuzione di intensità dell'intorno.

$$f_A(i) = \begin{bmatrix} pt_i = (x_i, y_i) \\ ori_i \\ s_i \\ d_{Ai} \end{bmatrix} \text{ per } i = 1, 2, \dots, N$$

$$f_B(j) = \begin{bmatrix} pt_j = (x_j, y_j) \\ ori_j \\ s_j \\ d_{Bj} \end{bmatrix} \text{ per } j = 1, 2, \dots, M$$

$d_{Ai}$  e  $d_{Bj}$  sono quindi vettori descrittori composti da 128 numeri reali e ricavati durante l'analisi dell'intorno centrato nel punto di interesse, secondo le modalità descritte nella sezione 3.2.2.2, all'interno del paragrafo "Descrittore dei punti d'interesse".

Il primo passaggio compiuto è quello della normalizzazione, in modo tale che i vettori abbiano media pari a 0 e norma pari a 1. Siano  $dn_{Ai}$  e  $dn_{Bj}$  i descrittori normalizzati di  $d_{Ai}$  e  $d_{Bj}$ , rispettivamente, risulta:



$$dn_{Ai} = \frac{d_{Ai} - \mu(d_{Ai})}{\|d_{Ai}\|_2}$$

$$dn_{Bj} = \frac{d_{Bj} - \mu(d_{Bj})}{\|d_{Bj}\|_2}$$

Dove  $\mu(\cdot)$  e  $\|\cdot\|_2$  sono gli operatori che calcolano la media e la norma 2 rispettivamente.

Una volta effettuata la normalizzazione è possibile calcolare la distanza tra i due vettori, ovvero la somma delle differenze al quadrato dei rispettivi elementi dei vettori o ancora, secondo [11]:

$$\Delta(dn_{Ai}, dn_{Bj}) = (dn_{Ai} - dn_{Bj})^T (dn_{Ai} - dn_{Bj})$$

Ritroviamo questa distanza nella formula che permette il calcolo dell'insieme *matches* delle coppie ordinate che costituisce l'output di *matchFeatures*:

$$(f_A(i), f_B(j)) \in matches \leftrightarrow$$

$$\leftrightarrow bestMatchInd(i) = j \text{ AND } \Delta(dn_{Ai}, dn_{Bj}) < thresholdValue$$

Dove

$$bestMatchInd(i) = \underset{j=1,2,\dots,M}{argmin} \left( \Delta(dn_{Ai}, dn_{Bj}) \right) \text{ con } 1 \leq i \leq N$$

Quindi la coppia  $(f_A(i), f_B(j))$  è elemento di *matches* se e solo se:

- la distanza tra i descrittori normalizzati delle due feature  $\Delta(dn_{Ai}, dn_{Bj})$  è la minima tra tutte le possibili distanze calcolate tra il descrittore normalizzato dell'i-esima feature del set  $f_A$  e tutti i descrittori normalizzati delle feature del set  $f_B$
- la distanza tra i vettori normalizzati delle due feature associate  $\Delta(dn_{Ai}, dn_{Bj})$  è inferiore a *thresholdValue*.

*thresholdValue* è un parametro di questa funzione di matching, definito secondo lo studio pubblicato in [6], e ha valore 5.

Un esempio di matching è mostrato in figura 3.12 a pagina seguente.

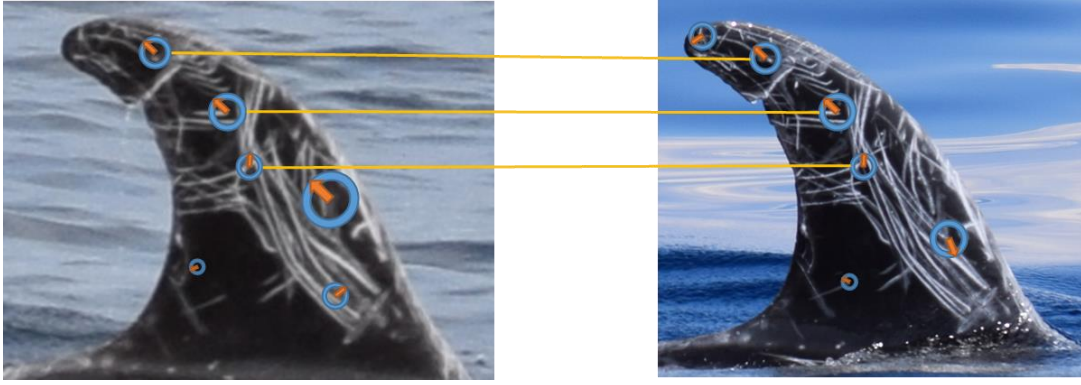


Figura 3.12

Esempio di match effettuati da `matchFeatures`. I cerchi rappresentano le feature trovate per entrambe le pinne, mentre le linee gialle rappresentano i match ottenuti.

Le coppie presenti nell'insieme *matches* così come vengono prodotte da *matchFeatures* non sono ancora perfettamente adeguate all'identificazione: è necessario effettuare alcune operazioni di filtraggio per escluderne alcune che potrebbero generare errori.

### 3.2.3.2 Operazioni di filtraggio

Le feature SIFT vengono filtrate in base al loro orientamento. In particolare:

$$\text{sia } (f_i, f_j) \in \text{matches}, f_i = \begin{bmatrix} pt_i = (x, y) \\ ori_i \\ s_i \\ desc_i \end{bmatrix} \text{ e } f_j = \begin{bmatrix} pt_j = (x, y) \\ ori_j \\ s_j \\ desc_j \end{bmatrix}$$

Se  $(-tolerance \leq (ori_i - ori_j) \leq tolerance)$

Allora  $(f_i, f_j) \in \text{filtered\_matches}$

Altrimenti  $(f_i, f_j) \notin \text{filtered\_matches}$

Gli elementi non inclusi in *filtered\_matches*, quindi, sono quei match le cui feature associate hanno un orientamento che dista tra loro più di *tolerance*. In questo modo le feature orientate in maniera evidentemente errata vengono scartate (si veda figura 3.13 a pagina seguente). Il valore di *tolerance* per gli esperimenti (ovvero 25 gradi) è stato definito secondo lo studio pubblicato in [6].

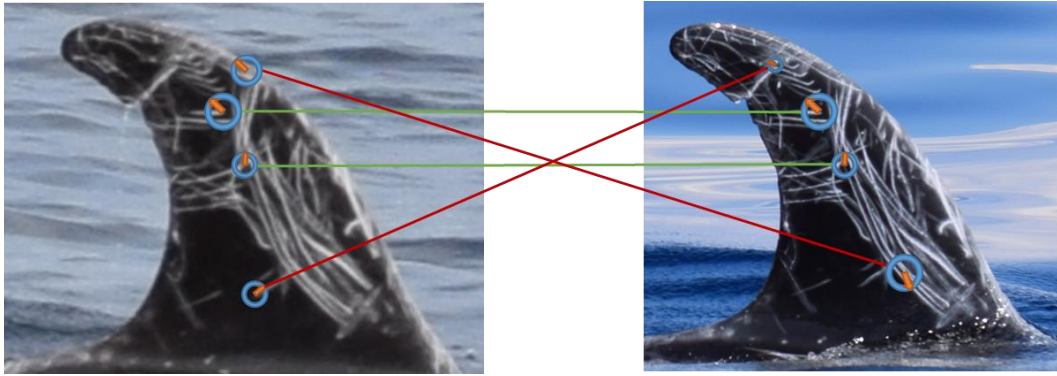


Figura 3.13

Esempio di match scartati e mantenuti. I cerchi rappresentano le feature trovate per entrambe le pinne, le linee verdi i match mantenuti e quelle rosse i match scartati in seguito all'operazione di filtraggio.

### 3.2.3.3 Scelta del modello migliore

Una volta ottenuti tutti gli insiemi di match correttamente orientati tra il ritaglio di pinna da identificare e tutti i ritagli di modelli, il sistema sceglie il modello in cui ci sono più corrispondenze, ovvero il caso in cui la cardinalità di *filtered\_matches* è massima. Per determinare se la scelta effettuata è accurata vi è un numero minimo di match necessari. In altre parole: se l'insieme *filtered\_matches* dove la cardinalità è massima ha una cardinalità minore di *nfmin*, allora la scelta non è accurata. Altrimenti è accurata. Il valore di *nfmin* per gli esperimenti (ovvero 7) è stato definito secondo lo studio pubblicato in [6].

### 3.2.3.4 Risoluzione delle ambiguità

È possibile trovarsi, specie se si utilizzano più modelli dello stesso delfino nel dataset, nella situazione in cui per uno stesso ritaglio di pinna sconosciuta due o più insiemi *filtered\_matches* abbiano la stessa cardinalità massima, maggiore di *nfmin*. In questo caso, SPIR v1 semplicemente scartava il ritaglio di pinna sconosciuta. SPIR v2, invece, analizza i nomi dei file dei vari modelli corrispondenti (contenenti appunto i nomi dei modelli) e segue questo algoritmo:

- se tra i nomi dei modelli trovati ce n'è uno che prevale sugli altri (ovvero che è presente in più della metà di tutti i nomi trovati) allora identifica con quel modello la pinna sconosciuta

- *altrimenti* considera la selezione non accurata, ma segnala all'utente l'immagine per ulteriori analisi manuali.

## 4 Esperimenti e risultati

### 4.1 Acquisizione

SPIR v2 è stato utilizzato per eseguire con successo la foto-identificazione di circa 30000 immagini raccolte in 44 missioni dal 10 aprile al 28 agosto 2019, utilizzando circa 2000 immagini di modelli acquisite e identificate tra il 2013 e il 2018. L'acquisizione delle immagini è stata effettuata dalla Jonian Dolphin Conservation, un'associazione di ricerca scientifica privata finalizzata allo studio dei cetacei nel Golfo di Taranto nel Mar Ionio Settentrionale. Tutte le fotografie sono state acquisite nel golfo di Taranto, in un'area di circa 14000 km<sup>2</sup>, utilizzando di volta in volta due o tre imbarcazioni e seguendo una rotta prestabilita a zig-zag per massimizzare gli avvistamenti. Le missioni durano circa 5 ore al giorno, percorrendo all'incirca 35 miglia nautiche al giorno. La velocità è mantenuta tra 7 e 8 nodi e le uscite sono effettuate solo in buone condizioni meteo (scala Douglas  $\leq 3$  e scala Beaufort  $\leq 4$ ). La data, condizioni meteo, distanza, orario del primo contatto e dimensione del gruppo vengono registrate durante gli avvistamenti. Vengono utilizzate diverse macchine fotografiche professionali con lenti diverse, e le condizioni di luce possono essere molto differenti tra loro.

### 4.2 I dati dell'esperimento

#### 4.2.1 I modelli

Come già accennato, per la foto-identificazione automatica sono state utilizzate 1914 immagini note, acquisite e identificate nel corso degli anni in 23 missioni dal 2013 al 2018. Gli esemplari unici sono 93 e sono così distribuiti:

- 45 esemplari unici identificati tra il 2013 e il 2016
- 11 esemplari unici identificati nel 2017
- 37 esemplari unici identificati nel 2018

I 45 esemplari del primo gruppo sono anche stati utilizzati per la valutazione di SPIR v1 in [6].

### 4.2.2 Il dataset di valutazione

Per la valutazione delle performance software in questa tesi, sono state scelte e verificate manualmente le immagini foto-identificate appartenenti a sei giornate di avvistamenti:

- il 2 giugno 2019
- il 7 giugno 2019
- il 2 luglio 2019
- il 3 luglio 2019
- il primo agosto 2019
- il 4 agosto 2019

Queste date campione corrispondono ai primi due avvistamenti di ogni mese di giugno, luglio e agosto 2019. Data la mole immensa dei dati analizzati, la verifica dei rimanenti dati ottenuti dal software verrà eseguita dilazionandola nel tempo da un team di ricercatori e biologi che si occuperà anche di analizzare i dati raccolti e trarne conclusioni sulla distribuzione della specie nell'area di ricerca.

## 4.3 Risultati sperimentali

La tabella 4.1 riassume i risultati rilevati a seguito della verifica manuale dell'output di SPIR v2.

Data avvistamento	Immagini totali	Immagini scartate	Identificazioni errate
<b>02/06/2019</b>	315	48,89%	13,66 %
<b>07/06/2019</b>	372	49,19%	6,35 %
<b>02/07/2019</b>	303	59,41%	44,71 %
<b>03/07/2019</b>	445	54,38%	2,96 %
<b>01/08/2019</b>	488	59,63%	36,04 %
<b>04/08/2019</b>	140	67,14%	15,21 %

Tabella 4.1 Tabella riassuntiva dei risultati

Un rapido esame dei dati ottenuti sembrerebbe indicare due grossi problemi:

- La percentuale di immagini scartate è alta (56,44% in media)

- Le performance dell'algoritmo sono apparentemente non stabili, infatti la percentuale di errore passa da valori vicini al 3% nel caso del 3 luglio fino a valori del 45% nel caso del 2 luglio

In realtà, nessuno dei due punti sopra elencati rappresenta un problema, in quanto entrambi hanno cause precise ed effetti limitati.

Nel caso del primo punto, esso è una costante di tutti e 44 gli avvistamenti. Non si tratta di un indicatore negativo: bisogna tener presente che, durante le missioni, i ricercatori addetti all'acquisizione tendono a scattare molte foto anche nell'arco di pochi secondi, per timore di perdere il momento in cui la pinna dell'animale è completamente e perfettamente visibile. Il risultato è una grande ridondanza nei dati, la quale è poi anche la causa dell'immensa mole di dati disponibile. Questo, aggiunto ad eventuali errori della routine di ritaglio automatico e alla scarsa qualità di alcune foto, porta allo scarto di numerosi ritagli da parte della routine addetta l'identificazione automatica; tuttavia, quasi mai ciò porta alla perdita di informazione su uno degli esemplari presenti in quell'avvistamento. Lo scopo ultimo infatti non è quello di identificare ogni immagine sottoposta al software, ma di riuscire ad identificare tutti e soli gli individui presenti durante ogni missione.

C'è anche un'altra motivazione che spiega un possibile aumento nella quantità di ritagli scartati, ed è la stessa che spiega il secondo punto presentato sopra, ovvero l'irregolarità nella percentuale degli errori. Nelle missioni del 2 luglio e del primo agosto, infatti, sono stati individuati due possibili nuovi individui, mai identificati. In questo caso il software reagisce sia scartando alcuni ritagli che non riesce ad identificare, sia (in misura molto maggiore) identificando in maniera errata l'individuo non noto con numerosi modelli, tutti diversi.

Sebbene SPIR v2 non sia stato pensato per riconoscere nuovi individui mai identificati, questo tipo di risultato può essere considerato come una sorta di "campanello d'allarme" che segnala l'eventuale presenza di nuovi esemplari.

Considerando quindi esclusivamente i 4 risultati che non contenevano individui mai identificati, la media dell'errore commesso nell'identificazione da parte di SPIR v2 si attesta intorno al 10%, contro il 12% di SPIR v1 [6].

#### 4.4 Presentazione dell'Output

Per ogni missione, il software ha organizzato i file di output secondo la seguente struttura:

- *comparison\_folder* è la cartella che contiene, in modo ordinato, le coppie modello-esemplare sconosciuto. In questo modo la verifica dell'identificazione può essere compiuta in maniera semplice e rapida. I modelli sono caratterizzati dal carattere \$ all'interno nel nome, mentre le pinne degli esemplari identificati corrispondenti nel proprio nome contengono il carattere @.
- *discarded\_images* è la cartella che contiene le immagini scartate dal software perché di scarsa qualità o prive di match attendibili. Contiene anche una sottocartella *hq\_nomatch*, dove vengono smistate le immagini che hanno matchato in modo valido con più modelli, senza che nessuno di essi prevalessesse nell'identificazione.
- *images\_with\_no\_features* è la cartella che contiene eventuali immagini per le quali non è stato possibile calcolare nemmeno una feature.



## 5 Conclusioni e sviluppi futuri

Obiettivo del presente lavoro di tesi è stato lo sviluppo di alcune modifiche ad un sistema automatico (SPIR v1) per la foto-identificazione di esemplari appartenenti alla specie *Grampus griseus* (delfino di Risso). Lo sviluppo di queste modifiche si è concretizzato in una versione più avanzata ed efficace del software (SPIR v2). Per identificare tali delfini si analizzano i graffi bianchi presenti sulla pinna dorsale che si formano naturalmente durante la vita e perdurano nel tempo.

La metodologia proposta prevede il ritaglio automatico delle immagini ottenute durante le missioni, l'estrazione delle feature SIFT dalle pinne dorsali e il confronto delle stesse con quelle di altri esemplari noti.

Gli esperimenti hanno evidenziato una percentuale di errore che si attesta intorno al 10%, e le modifiche apportate in SPIR v2 hanno permesso il lavoro di identificazione in pochi giorni su circa 30000 immagini senza intervento da parte dell'utente, risultato mai raggiunto da nessun sistema automatico in tale contesto. In questo senso, SPIR v2 migliora notevolmente lo stato dell'arte, rappresentato da DARWIN e SPIR v1.

Il sistema è aperto a sviluppi futuri, in particolare:

- Un miglioramento della routine per il ritaglio automatico, con una ristrutturazione che equilibri e armonizzi il lavoro svolto tra le CNN e le tecniche di computer vision
- L'introduzione della capacità di identificare un nuovo individuo mai riconosciuto
- L'estensione delle capacità di identificazione del software attraverso l'analisi del profilo della pinna, la quale lo renderebbe idoneo all'utilizzo anche per altre specie di cetacei. In tal senso, si veda il lavoro svolto nell'appendice sull'estrazione del contorno di interesse e sulla sua rappresentazione tramite descrittori invarianti alla posa dell'esemplare.

## 6 Appendice: la descrizione del contorno

Esiste un'ulteriore proprietà che può avere rilevanza nell'identificazione degli individui: il contorno e le sue frastagliature uniche. Sebbene nel delfino di Risso esse assumano decisamente meno importanza rispetto all'evidente superiorità distintiva dei graffi, le irregolarità nel contorno più concavo (si veda si veda figura 6.4 nel paragrafo 6.3) della pinna possono comunque potenzialmente perfezionare l'identificazione, e possono essere utilizzate anche per l'identificazione di altri esemplari, quali i tursiopi (*Tursiops truncatus*).

Sembra pertanto una naturale continuazione del lavoro finora descritto l'estrazione e la descrizione del contorno delle pinne dorsali, in vista di una futura integrazione con l'identificazione basata esclusivamente sui graffi.

### 6.1 Divisione tra pinna e background

Questo è il punto di partenza per l'estrazione del contorno. Sebbene questo argomento sia stato affrontato già nella sezione 3.2.1.1, in questo caso la divisione tra pinna e background è un processo di vitale importanza, poiché un errata divisione porta ad avere un contorno che non rispecchia effettivamente la forma della pinna, e che altera i punti in cui la pinna è frastagliata o viceversa i punti in cui non presenta alcuna irregolarità. Poiché l'analisi del contorno si baserà sulla valutazione di anche piccolissime irregolarità nel bordo della pinna, è necessario ridurre al minimo possibili errori nell'estrazione. In tale contesto la procedura vista nella sezione 3.2.1.1 risulta decisamente troppo sommaria e richiede una considerevole ristrutturazione.

Il primo passaggio è la creazione di una maschera utilizzando la conversione in CIE  $L^*a^*b^*$  (sezione 3.2.1.1, paragrafo "Conversione in CIE  $L^*a^*b^*$ ") e il calcolo della Soglia di Otsu (sezione 3.2.1.1, paragrafo "Soglia di Otsu"). Questa maschera verrà utilizzata in un passaggio successivo e viene per il momento tenuta da parte.

Il passaggio seguente prevede la rimozione dei canali G e B dell'immagine di partenza della pinna (quella in RGB). Anche in questo caso, come per la conversione in CIE  $L^*a^*b^*$ , si sfrutta l'organizzazione dello spazio colore per distinguere tra la pinna e il mare: il canale R

(ovvero il rosso) è quello che meglio registra la distinzione tra il grigio della pinna e il blu del mare.

A questo punto, l'immagine costituita unicamente dal canale rosso diviene l'input di un algoritmo che ne calcola il gradient magnitude attraverso il metodo sobel [12].

Questo algoritmo prende in input un'immagine e restituisce un'altra immagine che evidenzia i punti in cui l'immagine di partenza cambia più o meno drasticamente i suoi colori (si veda figura 6.1)



Figura 6.1

A sinistra: l'immagine privata dei livelli G e B e trasformata in toni di grigio.

A destra: il gradient magnitude calcolato evidenzia i pixel dove il cambiamento di colore è più drastico.

Il passaggio finale per ottenere la maschera definitiva che verrà utilizzata per distinguere la pinna dal mare e per estrarre il contorno è svolto dall'algoritmo *activecontour* [13].

Questo algoritmo prende in input la maschera creata con la Soglia di Otsu al primo passaggio e l'immagine creata con l'algoritmo del gradient magnitude appena presentata e, in un procedimento iterativo, adatta la prima alla seconda, correggendola e refinendola

in modo che essa ricalchi perfettamente la pinna. Il numero di iterazioni è stato preso in modo proporzionale alla dimensione dell'immagine, in modo empirico, cercando di ottenere il miglior risultato possibile.

La maschera così ottenuta viene sottoposta ad alcune operazioni morfologiche di perfezionamento: viene eseguita la funzione *regionprops* [9], che permette di selezionare e ritagliare l'immagine considerando esclusivamente l'area più grande ottenuta con la soglia (escludendo quindi schizzi, onde o comunque elementi più piccoli della pinna), nonché la funzione *bwmorph* [8] che sottopone la maschera ad operazioni di dilatazione ed erosione.

## 6.2 Rilevamento della punta della pinna e distinzione tra pinna destra e sinistra

L'attività successiva è quella di ritrovare le coordinate, all'interno dell'immagine, della punta della pinna. Questa informazione è utile sia per determinare se la pinna è destra o sinistra (e quindi catalogare ulteriormente le pinne per facilitarne l'identificazione) sia per estrarre il contorno di interesse finale, ovvero quello più concavo (si veda figura 6.4 nel paragrafo 6.3) compreso tra la punta e la fine della pinna.

La funzione *FinTip* svolge esattamente questo compito: calcola la punta della pinna restituendo le sue coordinate, prendendo in input il contorno della pinna sotto forma di maschera.

La maschera deve essere correttamente orientata, in modo tale che la punta della pinna sia in alto e la base della pinna in basso e deve delimitare un'unica regione di spazio. È anche necessario che la punta della pinna sia nella metà superiore dell'immagine, cosa che si verifica in tutte le immagini correttamente ritagliate. Naturalmente, la pinna può essere sia destra che sinistra.

*FinTip* procede dividendo la maschera in due parti uguali: la metà superiore e la metà inferiore. Successivamente, lavorerà esclusivamente sulla parte superiore, che contiene certamente la punta della pinna.

Un'importante funzione fornita da MATLAB e chiamata da *FinTip*, *bwboundaries* [14], calcola i punti che costituiscono il bordo della regione di spazio delimitata dalla maschera e

restituisce quindi un array di coordinate  $(y, x)$ . L'ordine in cui i punti vengono restituiti può variare a seconda della forma della maschera, pertanto sono stati riordinati in modo tale che, definito  $P$  l'insieme dei punti  $p_{1...n}$  (dove  $n$  è il numero dei punti ricavati da *bwboundaries*):

- $p_1$  è sempre il punto alla base della pinna, sul lato sinistro
- $p_2$  è il punto più vicino a  $p_1$  (secondo la formula della distanza tra due punti)
- $\forall i > 1, p_{i+1}$  è il punto più vicino a  $p_i$  (secondo la formula della distanza tra due punti), escludendo dall'insieme i punti  $p_1 \dots p_{i-1}$

Ovvero, i punti del contorno della pinna sono in ordine di prossimità e il primo punto è sempre quello in basso a sinistra.

Riordinati i punti del contorno, *FinTip* prosegue prendendo per ogni punto A del contorno appena riordinato, altri due punti B e C in posizione più avanzata all'interno dell'elenco riordinato dei punti trovati da *bwboundaries*. Ad esempio, alla prima iterazione prende le coordinate del primo punto nell'elenco dei punti trovati da *bwboundaries* e riordinati (e lo chiama A), le coordinate del settimo punto nell'elenco (e lo chiama B) e le coordinate del quattordicesimo punto nell'elenco (e lo chiama C). La misura dell'avanzamento nell'elenco (nell'esempio 7) dipende dalla quantità di punti trovati da *bwboundaries* (e quindi è direttamente proporzionale alla dimensione della maschera), ma una volta stabilita rimane la stessa per tutte le iterazioni.

In figura 6.2 a pagina seguente è possibile vedere i punti A, B e C alla prima iterazione del procedimento. Il punto A è quello verde, il punto B è quello giallo e il punto C è quello rosso.

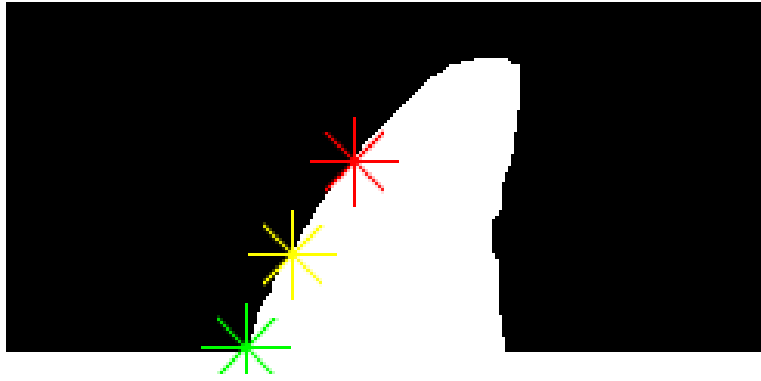


Figura 6.2

I punti A, B e C alla prima iterazione del procedimento. Il punto A è quello verde, il punto B è quello giallo e il punto C è quello rosso.

Alla prima iterazione *FinTip* calcola la distanza  $AB$ ,  $BC$  e  $AC$  e registra il valore della formula  $AC - AB + AC - BC$ . Per tutte le iterazioni seguenti, man mano che i punti A, B e C avanzano lungo il contorno della pinna, registra il valore della formula  $AC - AB + AC - BC$  solo nel caso in cui sia inferiore rispetto a quello calcolato nell'iterazione precedente. In questo specifico caso, registra anche le coordinate del punto B.

In questa maniera, al termine delle iterazioni, le coordinate registrate del punto B corrisponderanno al momento in cui la distanza  $AC$  era molto bassa rispetto alle distanze  $AB$  e  $BC$ , il che indica la presenza della punta della pinna proprio sul punto B (si veda figura 6.3).

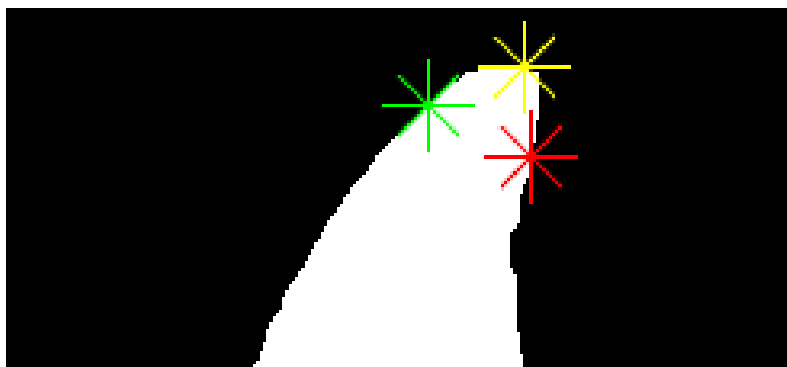


Figura 6.3

La ridotta distanza tra il punto A (in verde) e il punto C (in rosso) rispetto alle distanze tra A e B e tra B e C indica la presenza della punta nel punto B.

Ottenute le coordinate della punta della pinna, è banale riscontrare che nella stragrande maggioranza dei casi:

- Se la punta si trova nella metà destra della maschera, il lato osservato della pinna è il sinistro
- Se la punta si trova nella metà sinistra della maschera, il lato osservato della pinna è il destro

Questo controllo viene eseguito dalla funzione *isALeftFin*, che utilizza *FinTip* per determinare le coordinate della punta.

### 6.3 Il contorno di interesse

Una volta ottenute le coordinate della punta della pinna, è possibile calcolare il contorno di interesse. Il contorno di interesse è la parte più concava della pinna dorsale, dove si concentrano le irregolarità del profilo (si veda figura 6.4 a pagina seguente).



Figura 6.4

Il contorno più concavo della pinna, delimitato dalla punta (in giallo) e la fine della pinna (in verde) è ciò che costituisce il contorno di interesse (in rosso). È questa parte del contorno ad essere maggiormente significativa per il riconoscimento.

La funzione *FinElements* si occupa di estrarre il contorno di interesse. Si serve di *FinTip* per ottenere le coordinate della punta della pinna, di *bwboundaries* per ottenere le coordinate di tutti i punti che appartengono al contorno e di *isALeftFin* per determinare quale lato della pinna si sta osservando, se il sinistro o il destro. Procedo innanzitutto riordinando i punti del contorno ottenuti con *bwboundaries*, utilizzando esattamente lo stesso algoritmo utilizzato in precedenza nella sezione 6.2. In seguito, calcolo la posizione della pinna con *FinTip* e valuto il risultato di *isALeftFin*, determinando se il contorno di interesse si trova a sinistra o a destra della punta: se il lato della pinna osservato è il destro, il contorno di interesse è quello a sinistra della punta, mentre se il lato della pinna osservato è il sinistro, il contorno di interesse è quello a destra della punta. Poiché, ricordiamo, i punti ottenuti attraverso *bwboundaries* e in seguito riordinati procedono da sinistra verso destra, *FinElements* determina il contorno di interesse considerando esclusivamente i punti con



posizione precedente (nel caso il lato della pinna sia il destro) o successiva (nel caso il lato della pinna sia il sinistro) a quella in cui è collocata la punta.

*FinElements* termina restituendo quindi:

- le coordinate della punta
- l'array contenente le coordinate ordinate di tutti i punti facenti parte del contorno di interesse
- l'informazione che descrive se il lato della pinna osservato è sinistro o destro.

#### 6.4 I descrittori del contorno di interesse

Sebbene il risultato ottenuto, ovvero le coordinate di ogni punto appartenente al contorno di interesse, sia potenzialmente in grado di descrivere le irregolarità del contorno di una pinna allo scopo di poter effettuare un confronto con altri individui, è necessario elaborarlo per poterlo sfruttare a livello pratico. Particolarmente desiderabile per questo scopo diviene la rappresentazione mediante descrittori invarianti al punto di vista e alla posa assunta dal delfino al momento dello scatto. La metodologia utilizzata in questa tesi per ottenere questi descrittori è simile a quella presentata in [15]. L'implementazione svolta e di seguito esposta, tuttavia, presenta alcune importanti differenze, facilmente individuabili.

Dato il contorno di interesse  $C$ , rappresentato come un insieme ordinato di coordinate  $\{(x_1, y_1) \dots (x_n, y_n)\}$  con  $n$  numero di punti all'interno dell'insieme, viene considerato un sottoinsieme  $C_1$  ordinato identificato in tal modo:

- il primo punto  $(x_i, y_i)$  corrisponde al primo punto di  $C$  la cui distanza da  $(x_1, y_1)$  è pari o superiore ad un valore  $r$
- l'ultimo punto  $(x_j, y_j)$  corrisponde all'ultimo punto di  $C$  la cui distanza da  $(x_n, y_n)$  è pari o superiore allo stesso valore  $r$
- i punti tra  $(x_i, y_i)$  e  $(x_j, y_j)$  appartengono tutti al sottoinsieme  $C_1$

Nell'implementazione utilizzata, il valore  $r$  viene definito secondo la formula  $r = 3 \frac{n}{20}$ .

Per ogni punto  $(x_k, y_k)$  appartenente a  $C_1$  viene tracciata una circonferenza con raggio  $r$  e centro proprio in  $(x_k, y_k)$ . Per ogni circonferenza, viene valutato il rapporto tra l'area della pinna (in pixel) all'interno della circonferenza e l'area della circonferenza stessa. Ciò spiega il perché si rende necessario utilizzare un sottoinsieme di  $C$ : i primi e gli ultimi punti del contorno non permettono una divisione chiara dell'area desiderata (si veda figura 6.5).



**Figura 6.5**  
Nei punti più all'estremità del contorno di interesse non è possibile calcolare l'area della pinna all'interno della circonferenza

Questo rapporto rappresenta il descrittore della curvatura della pinna per quel punto  $(x_k, y_k)$ . Per ottenere informazioni più locali e meno globali, è possibile tracciare per ogni punto altre  $m$  circonferenze con raggi inferiori a  $r$  (figura 6.6 a pagina seguente).

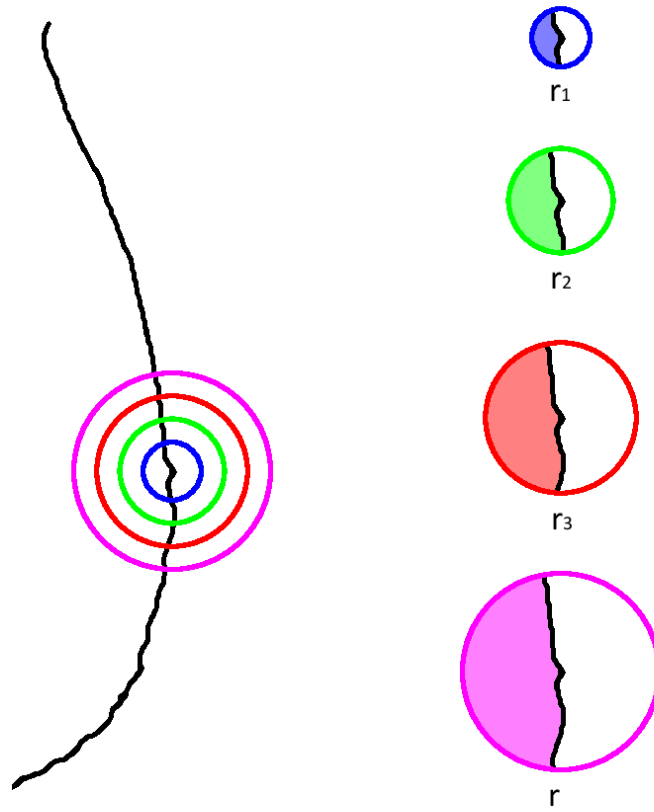


Figura 6.6  
Utilizzo di più circonferenze per descrivere il contorno più globalmente o più localmente

Nell'implementazione proposta abbiamo utilizzato altre tre circonferenze di raggio inferiore a  $r$ :

- $r_1 = \frac{n}{20}$
- $r_2 = \frac{n}{20} \cdot \frac{2}{3} r_1$
- $r_3 = \frac{n}{20} \cdot \frac{4}{3} r_1$

Il risultato finale è una matrice  $D$  di dimensioni  $n_1 \times m$ , dove  $m$  è il numero di valori assegnati ai raggi (e quindi di circonferenze tracciate in un punto), e  $n_1$  è il numero di elementi nell'insieme  $C_1$ . Lo scalare  $D_{ij} \in [0,1]$  calcolato attraverso il rapporto delle aree è il descrittore della curvatura per l' $i$ -esimo valore di  $r$  al  $j$ -esimo punto del contorno  $C_1$ .

Questi descrittori sono invarianti al punto di vista e alla posa assunta dal delfino nel momento dello scatto, e potranno essere utilizzati come dati di input in un matcher di natura simile a quello descritto per l'identificazione nella sezione 3.2.3.

#### 6.4.1 Validità dell'approccio

L'approccio utilizzato in [15] utilizza, al posto delle circonferenze per ogni punto del contorno, i quadrati circoscritti alle circonferenze (si veda figura 6.7). In tal modo, viene calcolato l'integrale del trapezoide ottenuto dai lati del quadrato e dal contorno di interesse e viene utilizzato quello per il calcolo del rapporto.

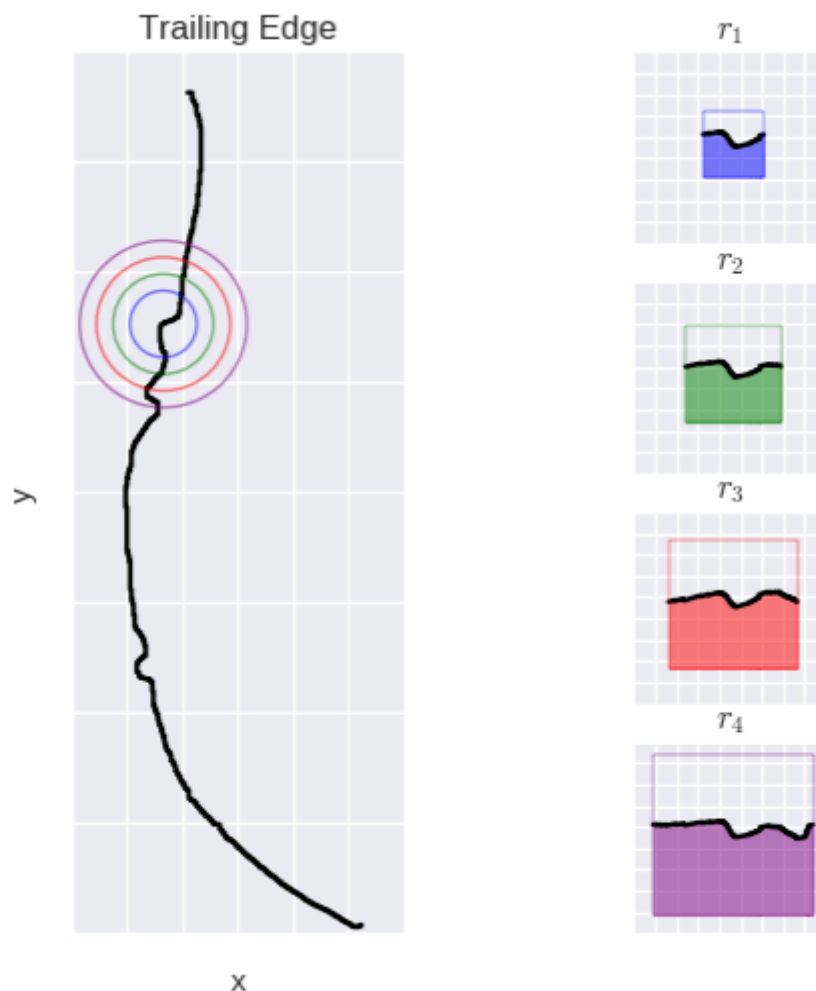
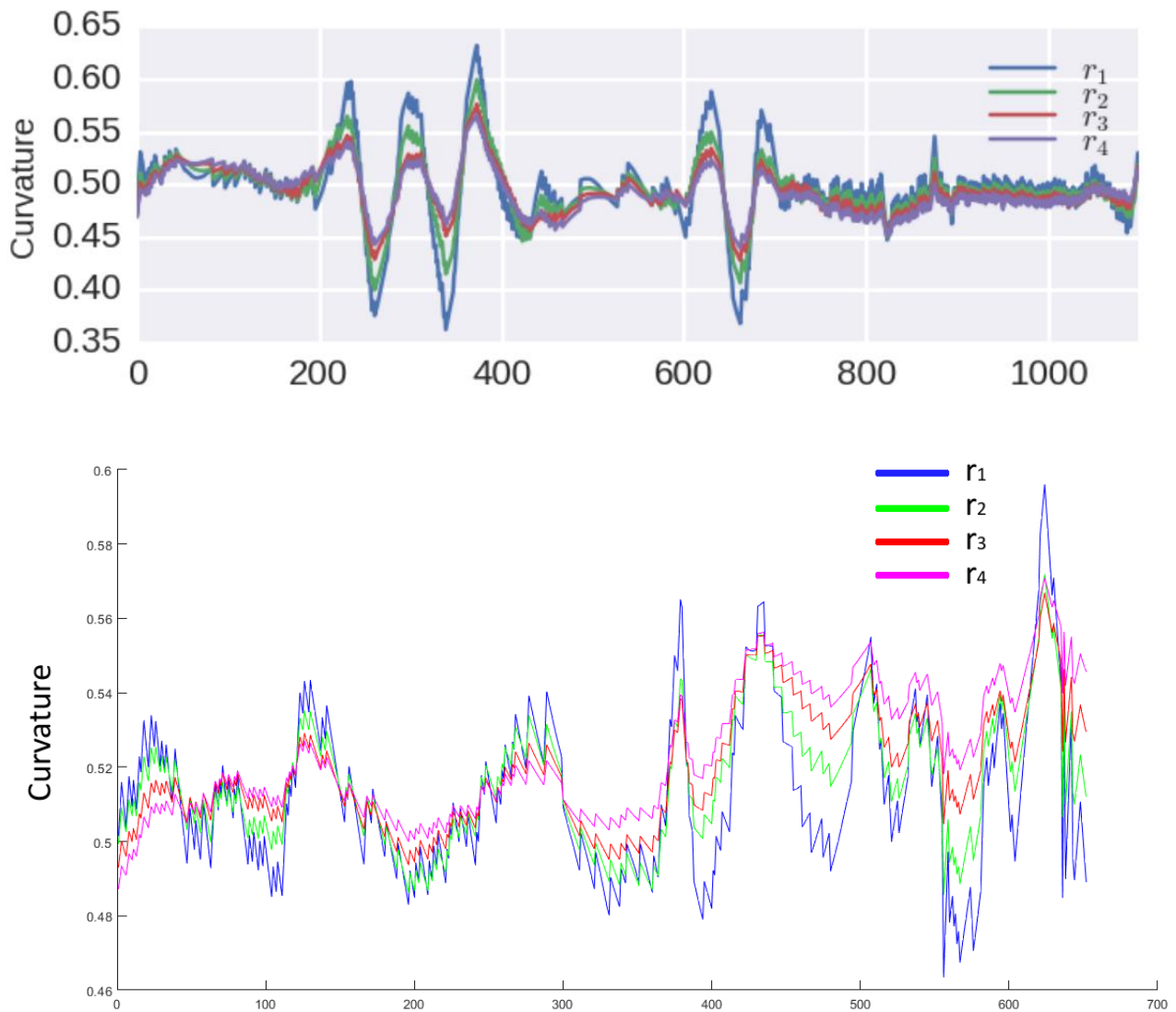


Figura 6.7  
Utilizzo di quadrati circoscritti alla circonferenza in [15]

Sebbene dal punto di vista matematico quel modo di procedere possa sembrare più intuitivo o fattibile, lavorando invece in termini di immagini digitali e considerando le

esigenze implementative, il metodo utilizzato in questa sede semplifica di molto i calcoli, riducendo le operazioni da svolgere e l'approssimazione, ottenendo comunque la stessa descrizione della curvatura del contorno (figura 6.8).



**Figura 6.8**

**In alto: rappresentazione dei descrittori ottenuti con il metodo definito in [15]**

**In basso: rappresentazione dei descrittori ottenuti con il metodo realizzato in questa sede**

**N.B. I contorni rappresentati non appartengono allo stesso delfino.**

## 7 Bibliografia

- [1] M. A. McCarthy and K. M. Parris, “Clarifying the effect of toe clipping on frogs with Bayesian statistics,” *Appl. Ecol.*, vol. 41, no. 4, pp. 780–786, 2004.
- [2] S. L. Petersen, G. M. Branch, D. Ainley, P. Boersma, J. Cooper, and E. Woehler, “Is Flipper banding of penguins a problem?,” *Mar. Ornithol.*, vol. 33, pp. 75–79, 2005.
- [3] R. Wilson and C. McMahon, “Measuring devices on wild animals: What constitutes acceptable practice?,” *Front. Ecol. Environ.*, vol. 4, pp. 147–154, 2006.
- [4] J. Kiszka, G. Braulik, R. Reeves, and L. Lowry, “Grampus griseus, IUCN Red List of Threatened Species, Assessment Information in detail.” [Online]. Available: <https://www.iucnredlist.org/species/9461/50356660>.
- [5] J. Stewman, R. Stanley, and M. Allen, “DARWIN: A System to Identify Dolphins from Fin Profiles in Digital Images,” in *8th Annual Florida Artificial Intelligence Research Symposium*, 1995.
- [6] V. Renò *et al.*, “A SIFT-based software system for the photo-identification of the Risso’s dolphin,” *Ecol. Inform.*, vol. 50, 2019.
- [7] N. Otsu, “A Threshold Selection Method from Gray-Level Histograms,” *IEEE Trans. Syst. Man. Cybern.*, vol. 9, no. 1, pp. 62–66, 1979.
- [8] “Morphological Operations on Binary Image (bwmorph).” [Online]. Available: <https://it.mathworks.com/help/images/ref/bwmorph.html>.
- [9] “Measure properties of image regions (regionprops).” [Online]. Available: <https://it.mathworks.com/help/images/ref/regionprops.html>.
- [10] D. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
- [11] “Pairwise distance between pairs of observations (pdist).” [Online]. Available: <https://it.mathworks.com/help/stats/pdist.html>.

- [12] "Gradient magnitude and direction of an image (imgradient)." [Online]. Available: <https://it.mathworks.com/help/images/ref/imgradient.html>.
- [13] "Segment image into foreground and background using active contours (activecontour)." [Online]. Available: <https://it.mathworks.com/help/images/ref/activecontour.html>.
- [14] "Trace region boundaries in binary image (bwboundaries)." [Online]. Available: <https://it.mathworks.com/help/images/ref/bwboundaries.html>.
- [15] H. J. Weideman *et al.*, "Integral Curvature Representation and Matching Algorithms for Identification of Dolphins and Whales," 2017, pp. 2831–2839.