

ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA
SCUOLA DI SCIENZE
CORSO DI LAUREA IN INGEGNERIA E SCIENZE
INFORMATICHE

ESTRAZIONE DI MINUZIE IN IMPRONTE DIGITALI DI BASSA
QUALITÀ

Relazione finale in

Fondamenti di Elaborazione di Immagini

Relatore:
Raffaele Cappelli

Presentata da:
Filippo Pronti

Co - relatore:
Matteo Ferrara

Sessione: Dicembre 2018

Anno Accademico: 2017/2018

Indice

1	Introduzione.....	5
2	Il riconoscimento delle impronte digitali.....	9
2.1	Cenni storici.....	9
2.2	Caratteristiche	11
2.3	Estrazione delle minuzie.....	13
2.4	Il confronto	17
2.5	Problemi nell'estrazione di minuzie.....	19
3	I database utilizzati.....	23
3.1	Database con Ground Truth	23
3.1.1	Il database utilizzato.....	23
3.1.2	Analisi dell'algoritmo di estrazione minuzie	25
3.1.3	Analisi dei problemi individuati	29
3.2	Database di bassa qualità.....	31
3.2.1	Il database utilizzato.....	31
3.2.2	Analisi di algoritmi di estrazione minuzie.....	33
3.2.3	Analisi dei problemi individuati	35
4	MQF	37
4.1	Le prove effettuate.....	37
4.2	Risultati ottenuti	43
5	Algoritmo ibrido.....	45
5.1	Le prove effettuate.....	45
5.2	Risultati ottenuti	50
6	Ottimizzazione parametri algoritmo di estrazione.....	55
6.1	I parametri considerati.....	55
6.2	Le prove effettuate.....	63
6.3	I risultati ottenuti.....	66
7	Conclusioni.....	69

1 Introduzione

In un mondo che si sta declinando in direzione di una sempre crescente virtualizzazione e in cui un numero di dati perennemente in aumento affolla dispositivi elettronici e meandri di Internet, trovare soluzioni per garantire la riservatezza di queste informazioni è diventato oggi un presupposto inderogabile. Si provi solamente a pensare a tutti gli sforzi messi in atto negli ultimi anni da grandi multinazionali e da colossi del settore: a questo proposito, paradigmatico è il clamore suscitato recentemente da FaceID, il meccanismo di sblocco schermo basato su riconoscimento facciale studiato e sviluppato da Apple.

Il bisogno di restringere l'accesso a informazioni sensibili e di poter identificare univocamente un individuo non si esaurisce qui però: un altro settore fortemente connotato da esigenze analoghe è quello della scienza forense. Questo ambito è forse ancora più presente nell'immaginario collettivo, in quanto la produzione letteraria e cinematografica intorno all'argomento è stata molto prolifica negli anni, riscuotendo da sempre molto interesse nel pubblico. A questo proposito si pensi allo sviluppo investigativo di un *crime drama*: poter riconoscere una persona attraverso le tracce lasciate sulla scena del crimine costituisce solitamente una prerogativa per poter risalire al colpevole. È in questo scenario costellato da richieste che uniscono più sfere cognitive che si inseriscono gli argomenti principi di questa trattazione, le impronte digitali e tutta la tecnologia sviluppata negli anni per rendere il loro riconoscimento sempre più accurato. Sul perché già a partire dal 1892 venne utilizzata questa tecnica di riconoscimento, vi sono due caratteristiche che rendono questi tratti fisiologici i candidati perfetti per gli utilizzi sopra citati: l'immutabilità - si pensi che esse si formano a partire dall'ottavo mese di gravidanza e non subiscono più modificazioni per l'intero corso della vita di un uomo - e l'individualità che, anche se non matematicamente provata, costituisce sul piano empirico un dato di fatto.

In passato, sin dai primi studi dell'antropologo britannico Francis Galton, probabilmente il più grande teorico e sostenitore della dattiloscopia, il riconoscimento delle impronte digitali era un'operazione che si svolgeva prettamente a mano: l'unica applicazione a fine del XIX secolo è quella nella lotta alla criminalità e ogni operazione sulle impronte viene effettuata sulla base di osservazioni ad occhio nudo. La grande svolta all'interno del settore si ha con

l'arrivo dei sistemi informatici che introducono i concetti di digitalizzazione delle informazioni e che ampliano le possibilità di acquisizione e di *matching* fra i campioni. Con quest'ultimo termine si fa riferimento al processo attraverso cui due impronte vengono messe a confronto per stabilire se appartenenti ad uno stesso dito. Ora, poiché l'approccio al mondo del riconoscimento delle impronte è di tipo sperimentale e poiché il numero di casistiche diverse è rapportabile al numero di individui che popolano il pianeta, spesso proprio in fase di confronto si verificano anomalie e situazioni inaspettate. Può accadere ad esempio che tentando di sbloccare il proprio *smartphone* apponendo il polpastrello sul piccolo sensore per le impronte, qualche volta l'identificazione fallisca e l'accesso al dispositivo venga negato. Portando ad estreme conseguenze il problema, di ben più grave entità è il caso in cui una persona non autorizzata riesca ad avere accesso ad un ambiente protetto: pensiamo ad esempio ad un locale ad accesso fisico autorizzato in cui vengano effettuate attività pericolose o in cui siano custoditi dati ad alto livello di segretezza. Onde evitare situazioni di questo tipo, è indispensabile progettare sistemi che tengano conto delle necessità finali e che adottino strategie più o meno restrittive: permettere un accesso indesiderato su un numero di mille tentativi potrebbe non essere un problema così grave in un sistema di sblocco schermo, mentre potrebbe avere ripercussioni di non trascurabile importanza nel caso di una transizione bancaria. L'introduzione di sistemi biometrici automatizzati risale ormai a circa quarant'anni fa e sin da subito l'individuazione di problematiche ed errori è il filo conduttore negli studi in questo settore. Il tentativo di arginare queste situazioni critiche ha portato nel corso degli anni alla formulazione di algoritmi sempre più raffinati. Ogni fase del processo di riconoscimento, dall'acquisizione alla digitalizzazione fino alla successiva attività di *matching* presenta criticità e negli anni si è cercato di intervenire con soluzioni che spaziassero su più fronti.

In questa trattazione l'accento viene posto sugli errori commessi in fase di estrazione delle minuzie: lo scopo perseguito nell'attività di sperimentazione si concretizza nel tentativo di arginare il numero di singolarità ignorate e di artefatti introdotti in fase di estrazione di un *template* di minuzie. Gli argomenti affrontati in questa dissertazione sono presentati secondo la seguente scansione: il Capitolo 2 fornirà un inquadramento generale sulle nozioni, sulle procedure e sugli strumenti utilizzati durante il percorso, con un'attenzione

particolare rivolta alle procedure di estrazione delle minuzie. Nel Capitolo 3 verranno invece affrontate le problematiche rilevate sui *database* utilizzati: i risultati ottenuti in questo capitolo possono considerarsi come punto di partenza per le successive fasi di sperimentazione. Nel Capitolo 4 e nel Capitolo 5 verranno illustrati ed implementati gli algoritmi documentati rispettivamente negli articoli [1] e [2]; successivamente questi ultimi verranno testati e gli esiti conseguiti saranno opportunamente documentati. Nel Capitolo 6, invece, l'attenzione si focalizzerà principalmente sull'estrattore del laboratorio BioLab: si analizzeranno le problematiche riscontrate in fase di estrazione delle minuzie e verrà condotto uno studio sui parametri utilizzati dall'algoritmo al fine di migliorarne le prestazioni.

2 Il riconoscimento delle impronte digitali

2.1 Cenni storici

L'introduzione della dattiloscopia (dal greco δάκτυλος, dito e σκοπέιν, guardare), ovvero il riconoscimento di persone attraverso l'esame delle loro impronte digitali, è stato per lungo tempo l'unico strumento realmente valido ai fini dell'identificazione di criminali. L'anno zero delle impronte digitali coincide dunque con l'insorgere di questa esigenza, il bisogno cioè di trovare un metodo di identificazione univoco che non possa essere raggirato dai malviventi. Testimonianze storiche dell'utilizzo di impronte sono rintracciabili già a partire dal III a. C. e provengono dall'antico impero cinese: pare infatti che l'imperatore Qin Shi Hang le utilizzasse per siglare documenti ufficiali [3]. Ora, benché l'intero corso della storia sia disseminato di episodi simili, anche in periodi ed epoche molto distanti fra loro, il primo caso risolto per via del riconoscimento di impronte risale al 1892 [4]. L'episodio è ambientato in Argentina e, nonostante i risvolti macabri dell'avvicendamento, costituisce un momento di svolta nella storia della dattiloscopia: il 19 giugno di quell'anno, in periferia di Necochea, vengono rinvenuti i corpi senza vita di due bambini, figli illegittimi della ventiseienne Francisca Rojas. Dalle prime prove raccolte il principale indiziato pare essere un vecchio servitore che, secondo le testimonianze della donna, da tempo la importuna; a pochi giorni dall'inizio delle indagini viene trovato però un alibi per l'uomo e nel mentre si viene a scoprire che la madre delle vittime ha un amante. Quest'ultimo le ha intimato di sbarazzarsi dei figli in quanto d'intralcio al loro matrimonio e da questo momento tutti i sospetti si concentrano sulla donna. Ad inchiodarla è proprio un'impronta intrisa di sangue rinvenuta sul luogo del delitto, prelevata dall'ispettore di polizia inviato sul luogo e comparata successivamente con quella della signora: la corrispondenza non lascia spazio ad esitazioni e la donna in seguito confessa il figlicidio. Importante è notare come il ricorso a questa tecnica di riconoscimento avvenga anche sulla scia dell'interesse suscitato dai recenti studi di Galton, che in quell'anno ha già pubblicato il libro "Fingerprints" e che ha scritto diversi articoli in cui rivendica il primato della dattiloscopia sulla già in uso antropometria. Da questo momento l'interesse e gli sforzi di ricerca si intensificano. Un primo tentativo in direzione di un'omologazione degli strumenti di identificazione si ha con

l'incontro tenutosi nel Principato di Monaco nel 1914 [5], incontro a cui prendono parte diversi Stati: è proprio in questa occasione che viene per la prima volta proposta l'adozione del riconoscimento di impronte come procedura di identificazione uniforme, ma questo tentativo viene interrotto dallo scoppio del primo conflitto mondiale. È invece proprio la Seconda Guerra Mondiale a determinare un nuovo incremento significativo dell'attività dattiloscopica: in questi anni, infatti, si intensificano le operazioni di schedatura nei confronti non solo di criminali, ma anche di tanti civili appartenenti a diverse fasce della popolazione e l'FBI arriva ad accumulare archivi dell'ordine di migliaia di campioni. In seguito, terminata la guerra, negli anni '60 nasce l'idea e al tempo stesso l'esigenza di automatizzare tutto il processo di acquisizione che fino a quel momento viene effettuato manualmente. È proprio la NIST (National Institute of Standards and Technology) a investire in termini di ricerca in questa direzione, affiancata da imprese di non trascurabile importanza, quali ad esempio Rockwell e Calspan. L'intento è quello di realizzare ciò che viene definito come "minutia-based fingerprint identification system", uno strumento in grado di automatizzare i processi di scansione, riconoscimento del disegno e classificazione. Nel 1969 è l'FBI stessa a porre sotto contratto la NIST [6] per la realizzazione di un sistema automatizzato: i costi in termini di tempo e di capitale umano impiegato in fase di acquisizione e catalogazione stanno infatti divenendo insostenibili. La prima svolta si ha nel 1975 quando viene messo in funzione un primo prototipo di lettore di impronte basato su tecniche conoscitive: questo è in grado però di catalogare solamente pochi dati biografici e informazioni relative alle sole minuzie in quanto memorizzare informazioni digitali relative all'impronta comporta per l'epoca il sostenimento di costi troppo elevati. Le direzioni di ricerca percorse dalla NIST negli anni a venire sono la digitalizzazione di impronte ad inchiostro e lo studio della compressione di immagini, dell'estrazione di minuzie e delle tecniche di *matching*. D'ora in avanti un numero costantemente in crescita di industrie inizia a sviluppare tecnologie analoghe: nel 1978 viene fondata la Printrak Computerized Fingerprint Identification User's Group e l'azienda diverrà nel corso dei due decenni successivi *leader* indiscussa all'interno del settore. Ciò che avviene poi ci riconduce ai giorni nostri e ci interessa da vicino: con un occhio rivolto alla realtà locale, nel 1993 viene costituito il BioLab, il laboratorio di sistemi biometrici dell'università di Bologna, in

cui l'attività di ricerca nei primi anni si concentra sullo sviluppo di tecnologie informatiche per il riconoscimento di impronte digitali. A questa si affiancano, nel corso degli anni, studi su altri aspetti dei sistemi biometrici, fra cui il riconoscimento dei volti, la generazione di impronte artificiali e la valutazione delle prestazioni.

2.2 Caratteristiche

Requisito fondamentale per poter parlare di riconoscimento di impronte è la definizione di sistema biometrico, ovvero un sistema basato sull'utilizzo di *pattern* in grado di riconoscere un individuo basandosi sull'unicità di un suo connotato fisico o comportamentale. A seconda degli scopi che si vogliono perseguire, si distingue poi fra sistema di verifica e sistema di identificazione. Nel primo un campione già precedentemente acquisito viene comparato con un altro contingente per verificare una possibile corrispondenza, mentre con il secondo si attua un raffronto di tipo molti a molti, al fine di identificare un singolo di cui si ignora l'identità. Nella pratica il funzionamento tipo di un sistema di riconoscimento di impronte digitali può essere schematizzato nelle seguenti tre fasi:

- *Enrollment*: in un primo momento, utilizzando uno *scanner* di impronte digitali, viene acquisita l'immagine dell'impronta; solitamente, dato l'elevato numero di fattori che possono incidere negativamente sulla qualità dell'immagine (sostanze oleose sul polpastrello, tagli di diversa entità), questa operazione viene ripetuta più volte per arrivare ad avere campioni di qualità soddisfacente. Successivamente si procede all'archiviazione delle impronte all'interno di un *database*: queste ultime non vengono però memorizzate sotto forma di immagine, ma a partire da essa viene creata una relativa rappresentazione digitale. Questo formato compresso, che prende il nome di *template*, incapsula un *set* di informazioni che identificano in modo univoco il campione ed è fra questi dati che figurano anche le minuzie, sulle quali si basano tutte le successive operazioni di *matching*.
- *Verification*: è ciò di cui si dibatteva precedentemente. Nello specifico, emblematico è il caso di un sistema in cui si voglia vincolare l'accesso solamente a determinate utenze. In questo caso, al verificarsi di un tentativo d'accesso, l'individuo richiedente dovrà fornire una propria identificazione e sottoporsi ad una scansione

dell'impronta. A questo punto essa verrà comparata con uno specifico campione precedentemente memorizzato ed il sistema verificherà che chi sta tentando l'accesso sia di fatto l'avente diritto e non un cosiddetto impostore.

- *Identification*: è analoga all'operazione precedente, ma il procedimento viene reiterato un numero N di volte. Per avere in mente un possibile campo di applicazione, si pensi ad esempio alle impronte rilevate su una scena del crimine: in questo caso l'obiettivo è l'identificazione di un individuo di cui si ignora l'identità. Il sistema metterà in atto una serie di comparazioni con altri campioni, prelevati ad esempio all'interno di un *database*, con lo scopo di trovare una possibile corrispondenza e dare quindi un volto ad un soggetto sconosciuto.

Soffermiamoci ora su alcuni aspetti meramente didascalici che ci serviranno in seguito per affrontare la trattazione. Facendoci aiutare dalla scienza della medicina legale, possiamo definire l'impronta digitale o dermatoglifi come “l'insieme di sollevamenti e depressioni, articolati in specifici disegni, che le creste papillari cutanee formano sulla superficie dei polpastrelli delle dita” [7]. Questi sollevamenti, definiti anche creste (o *ridges*, dall'inglese), seguono solitamente un andamento curvilineo continuo, ma a tratti possono assumere comportamenti improvvisi che ne modificano il normale flusso; queste discontinuità prendono il nome di minuzie (anche conosciute come dettagli di Galton). Solitamente in letteratura vengono identificati sette tipi di minuzie più comuni, mostrate in Figura 2.1.




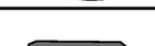



	Termination
	Bifurcation
	Lake
	Independent ridge
	Point or island
	Spur
	Crossover

Figura 2.1: Esempi di minuzie più comuni

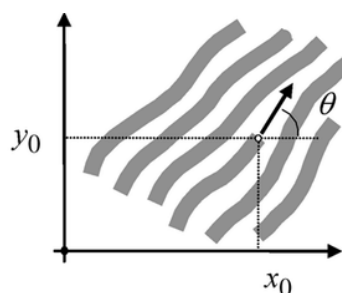


Figura 2.2: Terminazione

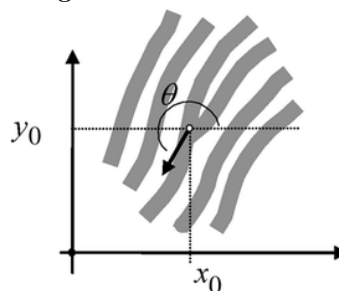


Figura 2.3: Biforcazione

Nell'ottica di un approccio più scientifico al problema, si utilizzerà qui la distinzione operata dall' FBI, che restringe la classificazione a sole due tipologie: terminazioni [Figura 2.2] e biforcazioni [Figura 2.3]. Per convenzione e al fine di evitare ambiguità, le minuzie vengono sempre rilevate sulle creste. Come specificato nel paragrafo precedente, l'individuazione di queste singolarità viene realizzata su un *template*, ossia su una rappresentazione digitale dell'impronta. Di conseguenza, più accurata sarà la creazione di questa riproduzione digitale e maggiori saranno le possibilità di non incorrere in errori in fase di riconoscimento.

2.3 Estrazione delle minuzie

Continuando nel nostro percorso conoscitivo, presentiamo qui una sezione dal contenuto già più tecnico e strettamente legata al problema in esame, ricorrendo a qualche richiamo al mondo dell'elaborazione di immagini e traendo qualche esempio dal volume [8]. Mantenendoci ad un livello di astrazione adeguato e osservando il problema ad alto livello, possiamo considerare l'estrazione di minuzie come il prodotto risultante dall'applicazione in sequenza di più algoritmi, ognuno dei quali determina esiti intermedi che fungono da *input* per gli *step* successivi. Esponiamo di seguito per sommi capi alcune delle operazioni più importanti: calcolo delle orientazioni, *enhancement*, estrazione dello scheletro e localizzazione delle minuzie con *crossing number*.

Orientazioni

Se si considera un *pixel* generico $[x, y]$ appartenente all'immagine di un'impronta, possiamo definire l'orientazione locale come l'angolo ϑ che la *ridge* forma, attraversando un intorno arbitrariamente piccolo centrato in $[x, y]$, con l'asse x. Nella maggior parte dei sistemi in circolazione questo valore non viene calcolato per ogni singolo *pixel*, ma si preferisce un approccio a intervalli discreti: i vantaggi che ne derivano sono sicuramente un minore dispendio di risorse dal punto di vista computazionale e la possibilità di ottenere successive stime sfruttando l'interpolazione. Il risultato di questa operazione è una matrice quadrata [Figura 2.4] in cui ogni valore $\vartheta[i, j]$ rappresenta un'approssimazione delle orientazioni locali in un intorno di $[x_i, x_j]$. A questo valore locale viene inoltre associato un altro

parametro r (coerenza) che indica il grado di affidabilità della stima sull'orientazione: quest'ultimo è basso in zone danneggiate o caratterizzate da molto rumore, alto in aree di elevata qualità.

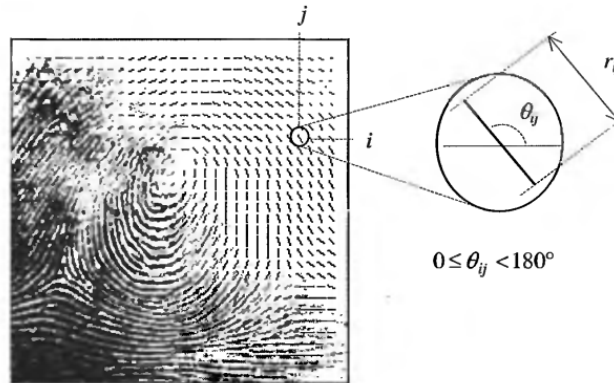


Figura 2.4: Immagine direzionale relativa a un'impronta e dettaglio; ogni elemento di questa griglia rappresenta una orientazione locale e la sua lunghezza è direttamente proporzionale al valore r di *reliability*.

Per ricavare l'immagine direzionale ci si serve del calcolo del gradiente in quanto per definizione questo indica la direzione di maggior variazione di intensità nel punto in cui viene calcolato. Per ragioni legate a diversi fattori, fra cui discontinuità che sorgono in corrispondenza di 0° e 90° , eccessiva sensibilità al rumore e situazioni di ambiguità in casi particolari di media fra le orientazioni, l'utilizzo di operatori classici come quelli di Prewitt e Sobel non è praticabile. Negli anni sono state proposte svariate soluzioni e fra tutte ne viene qui citata una, la più emblematica, suggerita da Kass e Witkin nel 1968 [9]: la loro idea, semplice quanto efficace, consiste nel raddoppiare gli angoli ed offre una valida soluzione al calcolo della media delle orientazioni in un intorno arbitrario $N \times N$.

Enhancement

Come sottolineato a più riprese, le prestazioni di un sistema di riconoscimento risentono fortemente della qualità dei campioni acquisiti. Muovendo dal presupposto che una percentuale consistente di immagini (intorno al 10%) possiede una qualità molto bassa a causa di problematiche contingenti o per stessa natura di determinate tipologie di impronte, sorge il bisogno di uno strumento in grado di migliorare la qualità del campione. È proprio questo lo scopo che viene perseguito in fase di *enhancement*. L'immagine utilizzata in ingresso è solitamente a scala di grigi, mentre l'*output* prodotto dal

procedimento è in scala di grigi o, come nel caso dell'estrattore del laboratorio BioLab, un'immagine binaria. Le normali tecniche di miglioramento della qualità di immagini generiche hanno dimostrato di non essere efficaci a questo proposito, nonostante possano fornire comunque un supporto per successive fasi di affinamento. Una possibile soluzione è ad esempio una normalizzazione effettuata secondo la formula mostrata di seguito:

$$\mathbf{I}[x, y] = \begin{cases} m_0 + \sqrt{(\mathbf{I}[x, y] - m)^2 \cdot v_0 / v} & \text{if } \mathbf{I}[x, y] > m \\ m_0 - \sqrt{(\mathbf{I}[x, y] - m)^2 \cdot v_0 / v} & \text{otherwise,} \end{cases}$$

Questa tecnica possiede però delle limitazioni: non riesce ad arginare piccoli tagli che si formano sulle *ridges*, non è in grado di colmare i cosiddetti *lake*, piccole fenditure che si creano all'interno delle creste e non riesce a separare *ridges* parallele che presentano zone di contatto.

Un approccio migliore prevede l'introduzione di filtri contestuali: questi, a differenza di quelli classici studiati in sede di elaborazione di immagini, applicano filtri con *set* di valori diversi per ogni area dell'immagine. I valori da utilizzare variano in questo caso in relazione alla stima delle orientazioni locali e delle frequenze. I filtri proposti in letteratura sono anche in questo caso molteplici: qui si espone la soluzione adottata all'interno dell'algoritmo di estrazione di BioLab, ossia il ricorso ai filtri di Gabor. La formula di applicazione è la seguente:

$$g(x, y; \theta, f) = \exp\left\{-\frac{1}{2}\left[\frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2}\right]\right\} \cdot \cos(2\pi f \cdot x_\theta)$$

Nel caso di immagini di impronte, f è il valore delle frequenze locali, ϑ la stima delle orientazioni locali e σ_x, σ_y devono essere impostati tenendo conto che valori alti rendono il filtro più robusto in situazioni di rumore nell'immagine, ma espongono al rischio di creare artefatti per quanto riguarda *ridges* e *valleys*. Dunque l'obiettivo è quello di trovare parametri che garantiscano il raggiungimento di un equilibrio.

Scheletro

In generale, per scheletro di un'immagine si intende un'immagine binaria che preserva le caratteristiche topologiche dell'oggetto: in termini formali, lo scheletro è il luogo dei centri dei cerchi, tangenti in almeno due punti, che sono completamente contenuti nel *foreground*.

Questa operazione è tanto semplice quanto importante in quanto è sull'immagine ottenuta al termine di questa procedura che, nel caso dell'algoritmo di estrazione di BioLab, verrà eseguita la successiva localizzazione delle minuzie. L'immagine utilizzata in ingresso è quella ricavata in fase di *enhancement* e su questa viene applicato un algoritmo di *thinning*. Quest'ultimo opera un assottigliamento delle *ridges* portandole ad avere spessore 1. Gli algoritmi più efficienti sono quelli che adottano un approccio iterativo e uno dei più noti è sicuramente quello di Hilditch. Ne illustriamo brevemente il comportamento.

n ₃	n ₂	n ₁
n ₄	p	n ₀
n ₅	n ₆	n ₇

Considerando il *pixel* p , indichiamo con $A(p)$ il numero di transizioni da *background* a *foreground* lungo la sequenza ordinata $n_0, n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_0$ e con $B(p)$ il numero di *pixel* di *foreground* nell'intorno di p . Ad ogni passo l'algoritmo verifica queste quattro condizioni:

- $2 \leq B(p) \leq 4$
- $A(p) = 1$
- $n_2 = background \vee n_0 = background \vee n_4 = background \vee A(n_2) <> 1$
- $n_2 = background \vee n_0 = background \vee n_6 = background \vee A(n_0) <> 1$

Se il *pixel* considerato rispetta queste quattro condizioni, viene cancellato. L'algoritmo termina nel momento in cui non è più possibile operare alcuna cancellazione.

Estrazione di minuzie con *crossing number*

Avviene in questa fase l'estrazione vera e propria delle minuzie. La loro localizzazione si basa sull'analisi del *crossing number*. La formula utilizzata è la seguente:

$$cn(\mathbf{p}) = \frac{1}{2} \sum_{i=1..8} |val(\mathbf{p}_{i \bmod 8}) - val(\mathbf{p}_{i-1})|$$

Analizzandola, $p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7$ sono i *pixel* appartenenti all'intorno di p e

$val(p) \in \{0,1\}$ è il valore posseduto da p . Dato p avente $val(p) = 1$, si individuano le seguenti casistiche:

- $cn(p) = 2$: p è interno alla *ridge* [Figura 2.5];
- $cn(p) = 1$: p corrisponde a una terminazione [Figura 2.6];
- $cn(p) = 3$: p corrisponde a una biforcazione [Figura 2.7];
- $cn(p) > 3$ p è parte di una struttura singolare complessa.

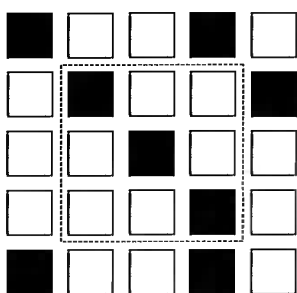


Figura 2.5: $cn(p) = 2$

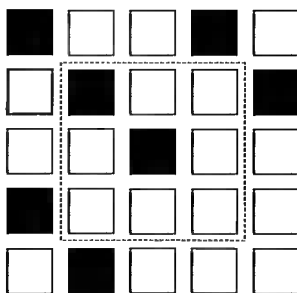


Figura 2.6: $cn(p) = 1$

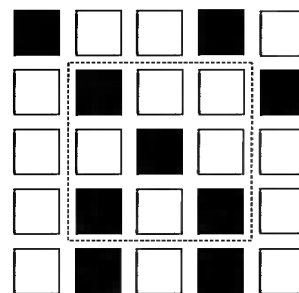


Figura 2.7: $cn(p) = 3$

2.4 Il confronto

Per verificare l'identità di un individuo o per rintracciare una corrispondenza all'interno di un *database* ci si serve di algoritmi cosiddetti di *matching*. Questi ultimi pongono a confronto due impronte digitali e, dopo averle comparate, determinano un grado di similarità fra le due. Ciò a cui si è interessati è sapere in definitiva se le due impronte appartengono allo stesso dito oppure no. Questa operazione è fra tutte quella che presenta più problematiche in quanto i fattori che incidono sulle diversità fra le acquisizioni di una medesima impronta sono molteplici ed occorre trovare approcci che ne tengano conto. Cercando di fare una sintesi, le principali problematiche sono:

- diversa disposizione del dito al momento della scansione;
- rotazione: il dito viene appoggiato sullo *scanner* con angolazioni sempre diverse nel corso di acquisizioni successive;
- distorsione non lineare: il problema nasce nel momento in cui, in fase di rilevamento, la forma tridimensionale del dermatoglifo viene trasformata in un'immagine che si sviluppa in sole due dimensioni. Questa complicazione rappresenta il motivo di fallimento degli approcci basati su utilizzo di *pattern* rigidi;

- pressione e condizioni della pelle al momento della scansione;
- rumore: quest'ultimo può essere ad esempio introdotto da tracce residue rimaste sul vetro dello *scanner* da scansioni precedenti;
- errori in fase di estrazione delle minuzie: si tratta di problematiche che sorgono in fase di applicazione dei procedimenti illustrati precedentemente (ne discuteremo nel corso della prossima sezione, in quanto questi errori rappresentano il nucleo di sviluppo di tutta la trattazione).

Si illustrano ora i passi effettuati a livello generale da un qualsiasi sistema di *matching*: per quanto concerne poi l'effettiva implementazione, le soluzioni proposte sono le più svariate e fondate anche su esigenze di diversa natura: i parametri considerati sono solitamente affidabilità del risultato e minimizzazione della complessità computazionale. Esaminiamo qui solo il caso di *matching* basato sul confronto di minuzie [Figura 2.8].



Figura 2.8: Esempio di *templates* fra cui è stata trovata una corrispondenza sfruttando tecniche di *matching* basato sul confronto di minuzie

Possiamo immaginare un *template* come una lista di informazioni inerenti alle singolarità: per semplicità, consideriamo le informazioni relative a una minuzia come una terna costituita da posizioni x e y e relativa orientazione. In questo caso due minuzie vengono accoppiate se la loro distanza spaziale e le loro orientazioni si trovano al di sotto di una determinata soglia prefissata. Dall'elenco delle problematiche stilato in precedenza, nasce però la necessità di effettuare delle operazioni di allineamento prima di procedere con questa operazione di *pairing*: anche in questo caso esistono diversi approcci utilizzati e tutti fanno ricorso ad attività di traslazione, rotazione o di ulteriori trasformazioni geometriche. Nell'ambito del *matching* basato su minuzie vi è un aspetto che ci preme

sottolineare in quanto rappresenta un'idea che verrà ripresa al Capitolo 3. Per definire il grado di affidabilità di una minuzia viene spesso utilizzato, come nel caso dell'estrattore BioLab, un indice di qualità locale: ad ogni singolarità viene cioè assegnato un punteggio (*score*) che ne indica in un certo senso in che misura tale minuzia contribuisce all'operazione di *matching*.

2.5 Problemi nell'estrazione di minuzie

Le problematiche in cui ci si può imbattere in fase di estrazione delle singolarità sono molteplici e non è ovviamente possibile catalogarle in un numero finito di casistiche. A questo proposito si pensi solamente al fatto che nel corso della storia si sono andate identificando un numero di circa 150 minuzie di diversa natura [10]. In certe situazioni - ne è una testimonianza anche il lavoro portato avanti nella seconda parte di questa trattazione - è difficoltoso anche all'occhio umano capire se una minuzia sia realmente tale o se si tratti invece di un artefatto. Situazioni contingenti dovute alla presenza di rumore nell'immagine, a sostanze sul polpastrello o ancora a caratteristiche proprie dello strato cutaneo di alcuni individui, creano spazi di ambiguità in cui intuire l'andamento delle *ridges* non è affatto un'operazione immediata. Lo scopo di queste righe è quello di fornire un'introduzione al problema, cercando di mostrare alcuni esempi significativi che tendono a riproporsi più frequentemente. Volendo proporre una classificazione a livello globale, si individuano in letteratura le seguenti macro problematiche:

- presenza di piccoli tagli che causano il tipico andamento frammentato nella *ridge*;
- *ridges* parallele che non sono del tutto separate e che sembrano anzi avere punti di contatto a causa dell'elevato grado di rumore o di pressione del dito sul sensore;
- lacerazioni o altre discontinuità sulla superficie cutanea.



Figura 2.9: Campione di buona qualità



Figura 2.10: Campione di media qualità



Figura 2.11: Campione di pessima qualità

Le immagini di cui sopra esibiscono esempi concreti dei fenomeni elencati [Figura 2.9] [Figura 2.10] [Figura 2.11].

A fronte di queste situazioni compromesse, vengono compiuti in fase di estrazione numerosi errori, classificabili come segue:

- estrazione di minuzie spurie (tradizionalmente denominate *false minutiae*);
- mancata rilevazione di minuzie *genuine* esistenti (le cosiddette *missed minutiae*).

È proprio sul calcolo del numero di *false* e *missed* rilevate in fase di *matching* che viene misurato il livello di efficienza di un algoritmo di estrazione delle minuzie. Nei seguenti capitoli ci serviremo a più riprese di questo strumento per verificare l'efficienza delle soluzioni proposte.

Muovendo dal presupposto che l'efficienza di un sistema di riconoscimento delle impronte è direttamente influenzata dalla qualità delle stesse, sorge l'esigenza di poterle classificare e stabilire una gerarchia di utilizzabilità dei campioni. È per questo motivo che nel 2004 il NIST mette a punto uno strumento di valutazione qualitativa delle impronte denominato NFIQ, rendendolo pubblico e facendolo diventare uno *standard de facto*. Ad ogni singolo campione acquisito viene associato un numero intero compreso fra 1 e 5: il valore 1 è indice di una qualità elevata, mentre man mano che ci si avvicina a 5 si ha un progressivo peggioramento. Questa tecnologia fornisce una valutazione di tipo predittivo sulle *performance* del *matcher*, cercando di prevedere il contributo positivo o negativo di una determinata immagine sulle prestazioni del sistema nella sua globalità. Di seguito vengono

mostrati alcuni esempi di immagini aventi NFIQ 5 riportati sul sito del NIST, in cui si vuole porre l'accento sulla diversità delle problematiche in cui si può incorrere in fase di acquisizione [Figura 2.12] [Figura 2.13] [Figura 2.14].



Figura 2.12: Presenza di cicatrici sulla pelle



Figura 2.13: Deformazioni avvenute in fase di acquisizione



Figura 2.14: Campione apparentemente buono ad un'analisi visiva, ma problematico per il *matcher*

Successivamente, a fronte di esigenze legate ad un miglioramento delle prestazioni e al bisogno di avere una gamma di valori di più ampio respiro, nel 2011 viene dato avvio allo sviluppo di un nuovo *standard*, ribattezzato col nome di NFIQ 2.0. Il progetto coinvolge più soggetti, vi prendono parte il NIST, il BSI (Federal Office for Information Security) e il BKA (Federal Criminal Police Office) che a loro volta lavorano fianco a fianco con enti di ricerca quali MITRE, Fraunhofer IGD, Hochschule Darmstadt (HAD) e Secunet. Il risultato di tali sforzi è un nuovo *standard* in cui le immagini vengono classificate in una scala da 1 a 100 e dove vengono considerati anche valori in virgola mobile: i principali benefici si concretizzano in una maggiore accuratezza, in una minore complessità computazionale e in una integrazione dello strumento con le piattaforme mobili.

3 I database utilizzati

3.1 Database con Ground Truth

3.1.1 Il database utilizzato

In questa sezione vengono discusse le proprietà dei *database* utilizzati per il *test* della metrica di qualità MQF, di cui verrà fornita in seguito un'ampia descrizione. Inizialmente la scelta è ricaduta su FVC2004 DB1_A [11], un *dataset* costituito perlopiù da immagini di buona qualità. La decisione è stata motivata anche dal fatto che per quest'ultimo sono disponibili i *templates* contenenti le corrette posizioni e orientazioni di ogni minuzia (Ground Truth), ottenute individuando manualmente le singolarità di ogni impronta del *dataset*. La bontà di questo lavoro consiste proprio nell'essere il frutto di un'osservazione attiva ed intelligente: qui l'esaminazione di ogni minuzia avviene infatti per mezzo di un'analisi empirica e non per via di strategie automatizzate e sottoposte a vincoli deterministici. Con l'avanzare delle prove ci si è poi resi conto che i campioni contenuti in FVC2004 DB1_A non costituivano una popolazione sufficientemente ampia affinché i risultati conseguiti potessero avere carattere generale. È proprio a partire da questa constatazione che si è scelto di utilizzare per i *test* successivi un *database* di dimensioni più consistenti. Un grande problema di FVC2004 DB1_A è inoltre quello di avere un ampio assortimento di immagini con NFIQ 1, 2, 3 e un numero di campioni pressoché irrisorio con NFIQ 4 e 5. Il *database* che si va ora ad esaminare è il DL02: questo, a differenza del primo, presenta una distribuzione più equilibrata delle immagini fra le varie classi di NFIQ. L'archivio in esame raccoglie 140 impronte, ognuna delle quali acquisita 12 volte ed ha una peculiarità rispetto a tutti quelli che verranno illustrati in questa trattazione: i campioni catalogati sono sintetici e non appartengono a persone fisiche. Lo strumento utilizzato per la generazione delle impronte si chiama SFINGE (Synthetic Fingerprint Generation) ed è stato sviluppato all'interno di BioLab [12]. Quest'ultimo restituisce inoltre, per ogni immagine generata, il relativo *template* di Ground Truth, cosa che si è rivelata ancora una volta molto utile ai fini delle operazioni di *matching* per il calcolo delle prestazioni.

I seguenti esempi [Figura 3.1] [Figura 3.2] ritraggono situazioni particolarmente compromesse e forniscono una prima panoramica su quelle che saranno le immagini su cui ci focalizzeremo in seguito.



Figura 3.1: Esempi di immagini tratte da FVC2004 DB1_A



Figura 3.2: Esempi di immagini tratte da DL02

- Nbis è stato sviluppato dal NIST per FBI e DHS (Department of Homeland Security). Il *software* è distribuito gratuitamente e non sottoposto ad alcun tipo di licenza [14];
- VeriFinger 10.0 è un prodotto commerciale che nasce già nel 1998, sviluppato dall'azienda Neurotechnology e negli anni ha visto il rilascio di numerose versioni con apporto di continui miglioramenti.

Nelle tabelle seguenti [Figura 3.4] [Figura 3.5] vengono mostrate le prestazioni raggiunte sui due *database* dai vari estrattori qui menzionati: partendo da sinistra figurano la categoria di NFIQ considerata, la percentuale di immagini che l'estrattore non è stato in grado di analizzare (Not Enrolled) e le *false* e *missed* generate assumendo come *templates* di riferimento quelli di Ground Truth.

Prestazioni raggiunte su FVC2004 DB1_A

Category	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
ALL	0.00 %	18.21 %	24.57 %	18.44 %	28.43 %
NFIQ1	0.00 %	17.46 %	20.44 %	17.70 %	22.42 %
NFIQ2	0.00 %	19.13 %	20.33 %	18.72 %	20.73 %
NFIQ3	0.00 %	22.80 %	66.02 %	22.43 %	70.16 %
NFIQ4	0.00 %	22.14 %	176.43 %	21.35 %	183.32 %
NFIQ5	0.00 %	22.83 %	217.39 %	23.13 %	218.87 %

(a) BioLab

Category	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
ALL	0.00 %	22.49 %	21.54 %	23.00 %	23.81 %
NFIQ1	0.00 %	21.45 %	20.10 %	21.87 %	21.92 %
NFIQ2	0.00 %	23.33 %	17.94 %	23.06 %	18.33 %
NFIQ3	0.00 %	29.67 %	40.38 %	29.37 %	42.54 %
NFIQ4	0.00 %	32.86 %	110.71 %	32.35 %	112.59 %
NFIQ5	0.00 %	36.96 %	98.91 %	36.39 %	97.80 %

(b) BstFVCEXtr

Category	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
ALL	0.75 %	19.48 %	51.56 %	19.26 %	61.20 %
NFIQ1	0.98 %	19.15 %	42.99 %	18.98 %	48.06 %
NFIQ2	0.00 %	19.20 %	53.30 %	18.66 %	56.13 %
NFIQ3	1.43 %	23.05 %	122.14 %	22.07 %	143.40 %
NFIQ4	0.00 %	30.71 %	125.00 %	29.13 %	132.05 %
NFIQ5	0.00 %	26.09 %	353.26 %	25.35 %	354.98 %

(c) FingerJet

Category	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
ALL	0.00 %	25.97 %	33.57 %	26.27 %	38.28 %
NFIQ1	0.00 %	25.38 %	30.45 %	25.62 %	34.01 %
NFIQ2	0.00 %	25.78 %	29.79 %	25.43 %	31.35 %
NFIQ3	0.00 %	31.33 %	69.05 %	30.97 %	74.09 %
NFIQ4	0.00 %	39.29 %	118.57 %	39.29 %	117.59 %
NFIQ5	0.00 %	47.83 %	173.91 %	48.07 %	171.92 %

(d) FX

Category	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
ALL	0.00 %	93.41 %	144.21 %	93.32 %	154.40 %
NFIQ1	0.00 %	93.24 %	139.73 %	93.07 %	147.20 %
NFIQ2	0.00 %	94.86 %	136.76 %	94.97 %	141.93 %
NFIQ3	0.00 %	91.85 %	190.28 %	92.06 %	204.67 %
NFIQ4	0.00 %	80.71 %	367.86 %	81.89 %	378.40 %
NFIQ5	0.00 %	79.35 %	463.04 %	80.16 %	464.90 %

(e) Nbis

Category	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
ALL	0.00 %	15.93 %	24.23 %	15.70 %	28.89 %
NFIQ1	0.00 %	15.66 %	20.15 %	15.50 %	23.07 %
NFIQ2	0.00 %	15.66 %	24.84 %	15.08 %	26.58 %
NFIQ3	0.00 %	18.63 %	54.27 %	17.21 %	60.87 %
NFIQ4	0.00 %	23.57 %	120.71 %	22.05 %	119.63 %
NFIQ5	0.00 %	32.61 %	171.74 %	32.37 %	172.29 %

(f) VeriFinger 10.0

Figura 3.4: Prestazioni ottenute dai vari estrattori su FVC2004 DB1_A

Ad una prima analisi si può notare come i risultati ottenuti da BioLab per immagini di NFIQ 1 e 2 siano molto positivi, riportando talvolta esiti migliori rispetto a quelli ottenuti dagli altri estrattori. I problemi sorgono invece in corrispondenza di NFIQ 3, 4, 5 (in

questo caso in relazione ad esempio a due estrattori più robusti come BstFVCEXtr e VeriFinger 10.0). Mano a mano che la qualità degrada, si notano infatti peggioramenti consistenti, soprattutto se considerato l'ingente numero di *false* introdotte.

Prestazioni raggiunte su DL02

Extractor	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
ALL	0.00 %	19.64 %	34.72 %	19.52 %	35.54 %
NFIQ1	0.00 %	10.05 %	3.67 %	10.12 %	3.84 %
NFIQ2	0.00 %	10.54 %	4.90 %	10.10 %	5.03 %
NFIQ3	0.00 %	17.38 %	24.92 %	17.35 %	25.88 %
NFIQ4	0.00 %	26.62 %	49.35 %	26.68 %	50.35 %
NFIQ5	0.00 %	40.48 %	109.80 %	41.06 %	115.63 %

(a) BioLab

Category	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
ALL	8.21 %	26.69 %	38.17 %	25.99 %	39.74 %
NFIQ1	0.35 %	11.32 %	13.83 %	11.30 %	14.17 %
NFIQ2	0.00 %	17.54 %	15.84 %	16.45 %	16.01 %
NFIQ3	1.89 %	27.87 %	55.55 %	27.36 %	58.83 %
NFIQ4	3.13 %	45.56 %	67.13 %	45.59 %	71.08 %
NFIQ5	38.32 %	58.39 %	72.18 %	58.20 %	75.66 %

(b) FingerJet

Category	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
ALL	0.00 %	93.75 %	121.23 %	93.88 %	122.28 %
NFIQ1	0.00 %	94.75 %	107.85 %	94.79 %	108.55 %
NFIQ2	0.00 %	96.80 %	109.11 %	96.93 %	110.50 %
NFIQ3	0.00 %	94.81 %	120.49 %	94.96 %	122.13 %
NFIQ4	0.00 %	93.01 %	130.24 %	93.01 %	131.06 %
NFIQ5	0.00 %	89.54 %	147.90 %	89.43 %	149.89 %

(c) Nbis

Category	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
ALL	0.00 %	16.89 %	20.79 %	16.68 %	21.45 %
NFIQ1	0.00 %	8.75 %	11.21 %	8.78 %	11.43 %
NFIQ2	0.00 %	8.64 %	13.04 %	8.22 %	13.42 %
NFIQ3	0.00 %	14.97 %	21.63 %	14.66 %	22.47 %
NFIQ4	0.00 %	21.22 %	25.81 %	20.84 %	26.45 %
NFIQ5	0.00 %	35.85 %	38.84 %	36.52 %	40.74 %

(d) VeriFinger 10.0

Figura 3.5: Prestazioni ottenute dai vari estrattori su DL02

Come per il *database* precedente, risultati molto positivi si hanno in corrispondenza di valori di NFIQ 1 e 2. Diversamente si può dire invece per immagini di NFIQ 4 e 5, dove in questo caso l'attenzione va focalizzata sul confronto delle percentuali ottenute da BioLab e VeriFinger 10.0. Il numero di *false* introdotte dal primo è quasi il doppio di quelle rinvenute dal concorrente.

Vediamo nella seguente sezione alcuni esempi significativi che pongono in luce le problematiche principali alla base di questi esiti.

3.1.3 Analisi dei problemi individuati

Le immagini riportate di seguito sono tutte prelevate dal *database* FVC2004 DB1_A, con un'attenzione particolare rivolta a situazioni altamente problematiche.

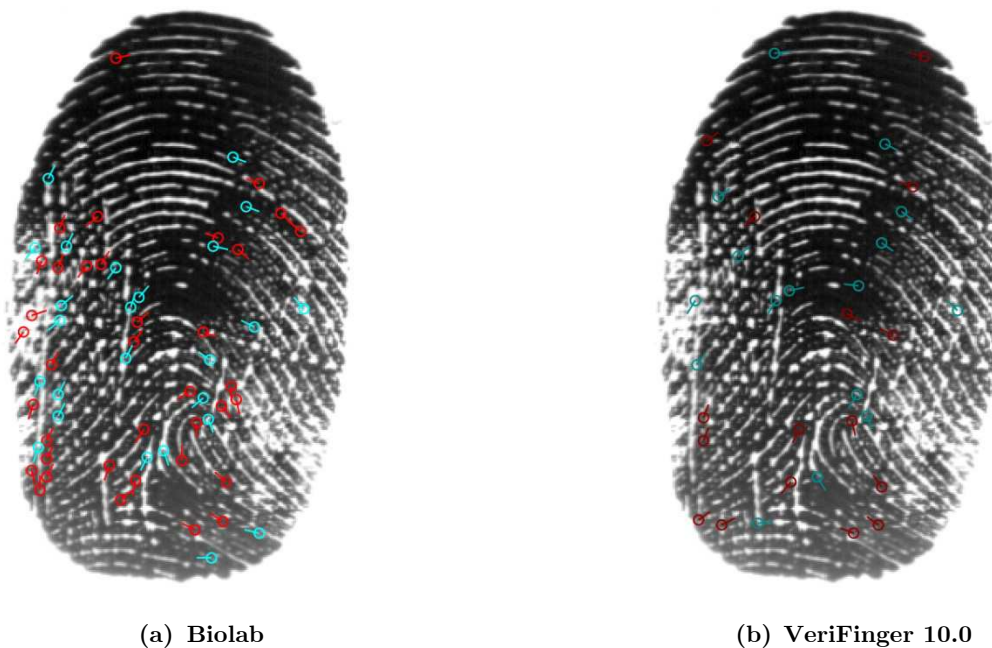


Figura 3.6: Minuzie estratte in un'immagine di NFIQ 5

In Figura 3.6 vengono mostrate le minuzie estratte rispettivamente da BioLab e da VeriFinger 10.0 da un'immagine classificata come NFIQ 5. In rosso sono marcate le biforcazioni, mentre in azzurro le terminazioni, convenzione che adotteremo sempre d'ora in avanti nell'analisi di successivi campioni. Ciò che salta immediatamente all'occhio è l'ingente numero di minuzie estratte da BioLab - se rapportate a quelle che figurano nel *template* di VeriFinger 10.0 - in corrispondenza di zone più compromesse. Nel caso

dell'esempio qui riportato, si può notare all'estremità sinistra una zona di bassa qualità. È proprio in corrispondenza di questa situazione che compaiono agglomerati di minuzie situate a distanza molto ravvicinata l'una dall'altra.

Una circostanza analoga è quella riportata nell'esempio successivo [Figura 3.7].

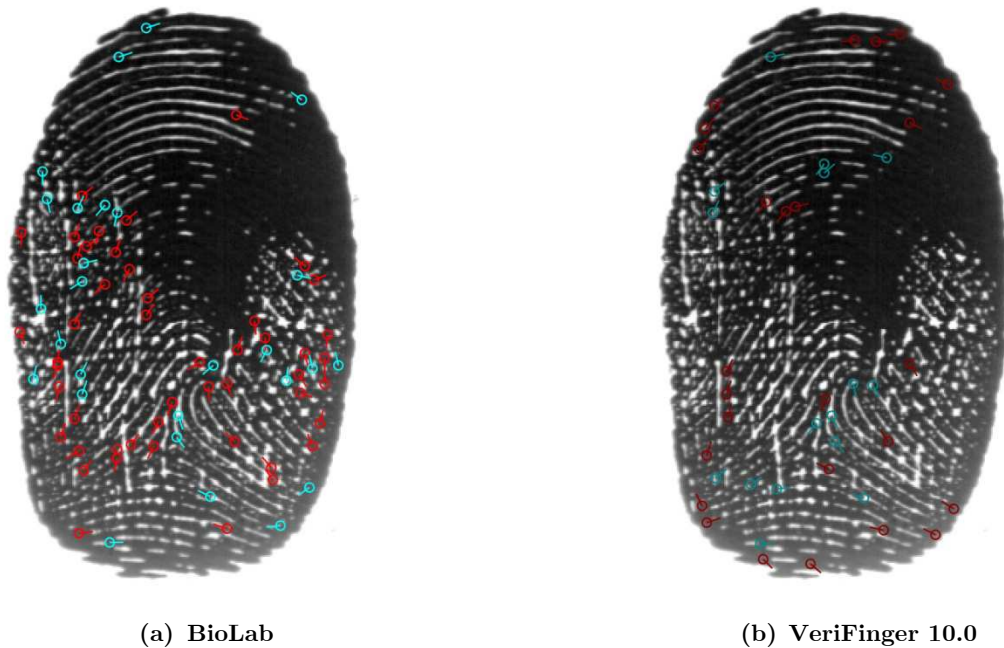


Figura 3.7: Minuzie estratte in un'immagine di NFIQ 5

Anche in questo caso il numero di minuzie estratte da BioLab è altissimo. Analogamente all'esempio precedente, anche qui *cluster* di minuzie si sono formati prevalentemente in zone di scarsa qualità e ad alto livello di rumore.

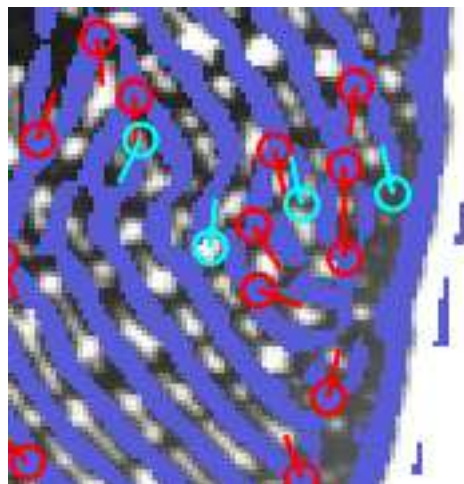


Figura 3.8: Dettaglio con sovrapposizione di immagine *enhanced*

Notiamo infatti come, a fronte di una situazione problematica, venga generata un'immagine *enhanced* [Figura 3.8] gravemente compromessa. Ciò che è interessante osservare è che tutti e due gli esempi precedentemente riportati appartengono all'impronta dello stesso dito: questo è indice del fatto che in acquisizioni differenti, i problemi che possono verificarsi sono di natura diversa, come già anticipato nel Capitolo 2. Partendo da tutte le considerazioni fino ad ora effettuate, l'operazione di ricerca si sviluppa su due fronti:

- da un lato, l'idea di trovare una metrica che, a prescindere dagli errori intercorsi e tralasciando l'analisi dell'immagine in sé, si focalizzi unicamente sulla disposizione spaziale delle minuzie all'interno del *template*;
- dall'altro, un approccio più classico basato sull'individuazione di *patterns* d'errore ricorrenti, con conseguente eliminazione delle minuzie coinvolte in queste situazioni.

3.2 Database di bassa qualità

3.2.1 Il database utilizzato

In una seconda fase di sperimentazione, incentrata sul miglioramento delle prestazioni di BioLab in presenza di immagini gravemente compromesse, anche le esigenze legate ai *database* sono cambiate. Le motivazioni per cui si è abbandonato l'impiego di DL02 sono di duplice natura:

- questo *database* possiede nel complesso una buona percentuale di immagini di alta qualità, mentre qui ci si vuole concentrare esclusivamente su immagini aventi NFIQ 3, 4 e 5.
- il Ground Truth di questo *database* è stato ricavato attraverso l'applicazione meccanica di regole prestabilite. Infatti, per limitare il più possibile errori in fase di identificazione delle minuzie corrette, le regole di scrematura seguono uno *standard* ben preciso (ISO/IEC 19794-2) e sono molto restrittive.

Abbandonato DL02, si è scelto di effettuare le sperimentazioni sui seguenti *database*: FVC2000 Db3_a e Db3_b [15], FVC2002 Db1_a, Db1_b, Db2_a e Db2_b [16] ed FVC2006 DB2_A e DB2_B [17]. Come nel caso di FVC2004 DB1_A, i campioni catalogati sono di natura biologica, ma ciò che li contraddistingue da quelli del primo è una

cospicua presenza di immagini di scarsa qualità. Infatti su un totale di 4400 acquisizioni, 1069 sono classificate con NFIQ 3, 4 e 5, mentre in FVC2004 DB1_A erano solamente 80. Nell'andare ad effettuare le prove ci si concentrerà solamente su queste ultime, tralasciando le impronte con NFIQ 1 e 2; inoltre i risultati riportati saranno relativi a tutte le 1069 immagini considerate d'ora in avanti come parte di un unico archivio.

Di seguito, le tabelle riassuntive sulle caratteristiche dei vari *dataset* [Figura 3.9].

FVC2000 Db3_a	
N. Impronte	100
Samples (per impr.)	8
Larghezza/altezza	448 x 478
DPI	500
NFIQ1 (%)	9.38
NFIQ2 (%)	31.12
NFIQ3 (%)	30.50
NFIQ4 (%)	18.25
NFIQ5 (%)	10.75

(a) FVC2000 Db3_a

FVC2000 Db3_b	
N. Impronte	10
Samples (per impr.)	8
Larghezza/altezza	448 x 478
DPI	500
NFIQ1 (%)	3.75
NFIQ2 (%)	10.00
NFIQ3 (%)	33.75
NFIQ4 (%)	35.00
NFIQ5 (%)	17.50

(b) FVC2000 Db3_b

FVC2002 Db1_a	
N. Impronte	100
Samples (per impr.)	8
Larghezza/altezza	388 x 374
DPI	500
NFIQ1 (%)	64,00
NFIQ2 (%)	26,00
NFIQ3 (%)	8,75
NFIQ4 (%)	0,50
NFIQ5 (%)	0,75

(c) FVC2002 Db1_a

FVC2002 Db1_b	
N. Impronte	10
Samples (per impr.)	8
Larghezza/altezza	388 x 374
DPI	500
NFIQ1 (%)	59.88
NFIQ2 (%)	34.88
NFIQ3 (%)	5.24
NFIQ4 (%)	0.00
NFIQ5 (%)	0.00

(d) FVC2000 Db1_b

FVC2002 Db2_a	
N. Impronte	100
Samples (per impr.)	8
Larghezza/altezza	262 x 492
DPI	500
NFIQ1 (%)	64,00
NFIQ2 (%)	26,00
NFIQ3 (%)	8,75
NFIQ4 (%)	0,50
NFIQ5 (%)	0,75

(e) FVC2002 Db2_a

FVC2002 Db2_b	
N. Impronte	10
Samples (per impr.)	8
Larghezza/altezza	262 x 492
DPI	500
NFIQ1 (%)	52.62
NFIQ2 (%)	40.88
NFIQ3 (%)	6.12
NFIQ4 (%)	0.38
NFIQ5 (%)	0.00

(f) FVC2002 Db2_b

FVC2006 DB2_A	
N. Impronte	140
Samples (per impr.)	12
Larghezza/altezza	351 x 492
DPI	500
NFIQ1 (%)	20.06
NFIQ2 (%)	55.65
NFIQ3 (%)	20.95
NFIQ4 (%)	2.32
NFIQ5 (%)	1.02

(g) FVC2006 DB2_A

FVC2006 DB2_B	
N. Impronte	10
Samples (per impr.)	12
Larghezza/altezza	351 x 492
DPI	500
NFIQ1 (%)	18.33
NFIQ2 (%)	52.50
NFIQ3 (%)	21.67
NFIQ4 (%)	2.50
NFIQ5 (%)	5.00

(h) FVC2006 DB2_B

Figura 3.9: Caratteristiche riassuntive degli otto *database*

3.2.2 Analisi di algoritmi di estrazione minuzie

Il primo problema che ci si è trovati a fronteggiare in fase di analisi è consistito nel non avere a disposizione dei *templates* di GroundTruth. Nel caso di FVC2004 DB1_A il lavoro di classificazione era infatti frutto di un'attività manuale, mentre per DL02 si applicavano

regole prestabilite fissate da uno *standard*. Non avendo a disposizione questi strumenti per le banche dati qui utilizzate, si è deciso di scegliere arbitrariamente come *templates* di riferimento quelli ricavati dagli estrattori BstFVCEXtr e VeriFinger 10.0. Questi ultimi, come apprezzabile dall’osservazione delle prove precedenti, sono ritenuti estrattori molto robusti e riportano risultati soddisfacenti anche in presenza di immagini di scarsa qualità. Poiché in questo particolare caso i *templates* di Ground Truth erano dunque più di uno (uno per ogni estrattore di riferimento), si è scelto di utilizzare il seguente schema per eseguire l’individuazione di *false* e *missed*.

BioLab	BstFVCEXtr	VeriFinger 10.0	
✓	✗	✗	<i>False</i>
✗	✓	✓	<i>Missed</i>

In pratica si considerano *false* le minuzie estratte da BioLab, ma non da BstFVCEXtr e VeriFinger 10.0: specularmente, lo stesso criterio è stato applicato alle *missed*, per cui BioLab avrà una minuzia mancante nel momento in cui essa sarà rilevata da entrambi gli altri due estrattori. Questa scelta nasce dall’esigenza di rendere ancora più affidabili le valutazioni effettuate dagli estrattori di GroundTruth, con l’intento di limitare il più possibile anche eventuali errori da parte di questi ultimi. Le seguenti tabelle [Figura 3.10] [Figura 3.11] riportano dunque le percentuali, in termini di *false* e *missed*, introdotte da BioLab rispetto a BstFVCEXtr e VeriFinger 10.0.

NFIQ	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
NFIQ3	0.00 %	19.15 %	39.57 %	18.93 %	40.58 %
NFIQ4	0.00 %	27.32 %	102.22 %	27.12 %	109.58 %
NFIQ5	0.00 %	28.54 %	157.53 %	28.83 %	177.51 %

Figura 3.10: *False* e *missed* generate da BioLab in rapporto a BstFVCEXtr

NFIQ	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
NFIQ3	0.00 %	24.03 %	29.55 %	23.91 %	28.34 %
NFIQ4	0.00 %	33.59 %	57.32 %	33.09 %	53.48 %
NFIQ5	0.00 %	40.90 %	82.06 %	39.67 %	75.98 %

Figura 3.11: *False* e *missed* generate da BioLab in rapporto a VeriFinger 10.0

3.2.3 Analisi dei problemi individuati

Anche nel caso di FVC2000 Db3_a e Db3_b, FVC2002 Db1_a, Db1_b, Db2_a, Db2_b ed FVC2006 DB2_A e DB2_B, le problematiche rinvenute sono simili a quelle riscontrate sui *database* precedenti. Forniamo qui qualche esempio introduttivo, in quanto i singoli casi verranno descritti e discussi in dettaglio nella Sezione 6.1. Le seguenti immagini, più che offrire un quadro generale sugli errori, hanno l'intento di mettere in luce situazioni in cui gli altri estrattori (prendiamo sempre ad esempio VeriFinger 10.0) si comportano meglio rispetto a BioLab.



(a) BioLab



(a) VeriFinger 10.0

Figura 3.12: Minuzie estratte in un'immagine di NFIQ 3

Nell'esempio mostrato [Figura 3.12] si presti attenzione al taglio presente in basso a destra: come si può notare, quest'ultimo provoca l'estrazione di svariate terminazioni da parte di BioLab, mentre le minuzie qui rinvenute da VeriFinger 10.0 sono in minor numero. Il problema in questo caso è legato al fatto che il primo estrattore non riesce in fase di *enhancing* a ricostruire le *ridges* interrotte.

Analizziamo ora l'esempio illustrato in Figura 3.13.

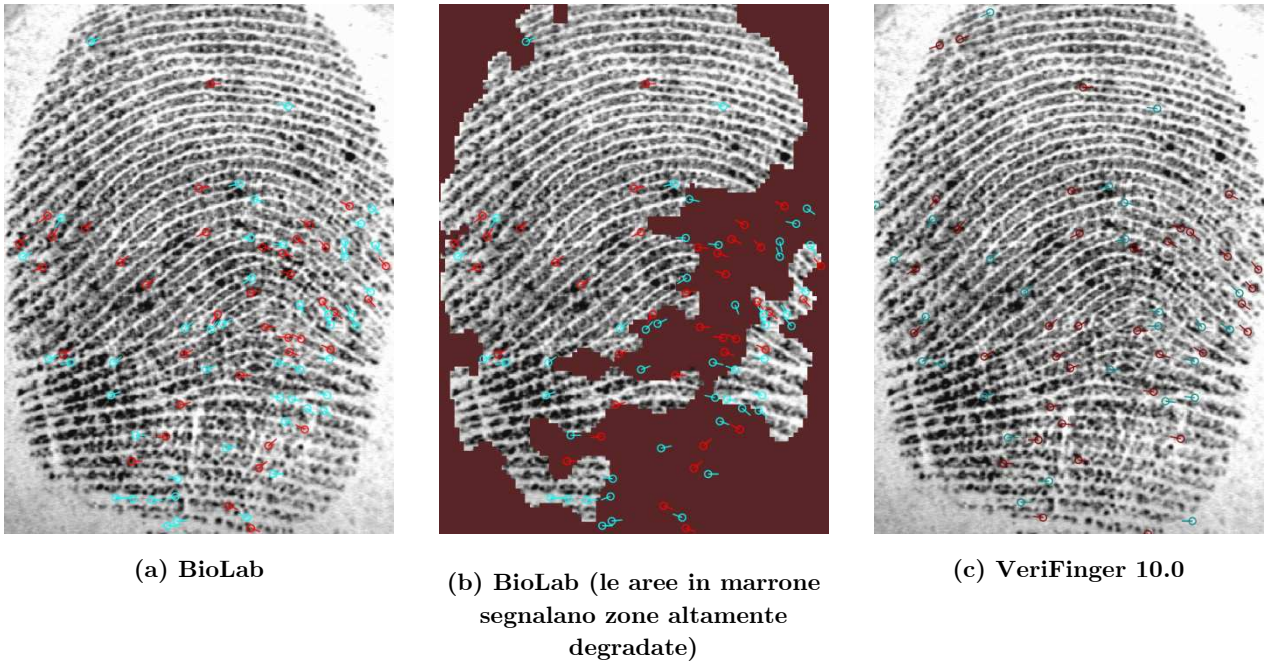


Figura 3.13: Minuzie estratte in un'immagine di NFIQ 4

In questo caso, analogo a quello illustrato in Figura 3.7, l'elevato numero di minuzie si concentra in una regione soggetta a disturbi presenti nell'immagine. Ciò è ancora più evidente se si considera l'area tinta di marrone nell'immagine al centro: essa infatti indica, data una soglia limite, il numero di zone interessate dal fenomeno di *bad quality*. Per quanto riguarda VeriFinger 10.0, anche per quest'ultimo la maggior parte delle minuzie viene rintracciata in quest'area, ma sicuramente la portata del problema ha un'entità diversa sulla base di ciò che è possibile osservare.

Partendo dalle problematiche qui intraviste si è proceduto all'interno del Capitolo 6 ad un'analisi più capillare e all'individuazione di parametri che potessero arginare il numero di *false* e di *missed* originate da BioLab.

4 MQF

4.1 Le prove effettuate

Le prestazioni di un sistema di *matching* sono strettamente legate alla qualità delle impronte utilizzate in fase di confronto, come già accennato nel Capitolo 2. Poter classificare le immagini in base alla qualità e costruire una gerarchia di importanza consente dunque ad un sistema biometrico di servirsi delle acquisizioni migliori in fase di comparazione. Anche se ad oggi NFIQ, con le sue evoluzioni (NFIQ 2.0), viene considerato lo *standard de facto* per la valutazione della qualità delle impronte digitali, la nostra attenzione si è focalizzata sull'algoritmo proposto da Yao et al. [1]. La scelta è stata indotta dall'osservazione degli esiti positivi riportati al termine della pubblicazione; gli autori mostrano infatti come la procedura proposta apporti un miglioramento anche nei confronti dello stesso NFIQ oltre al fatto che, a differenza di quest'ultimo, l'algoritmo introdotto nell'articolo si basa sulla distribuzione delle minuzie estratte e non sull'immagine dell'impronta.

MQF - appellativo con cui si fa riferimento all'algoritmo nell'articolo - si fonda sulla constatazione che minuzie poste a distanza molto ravvicinata nello spazio siano indice di aree di cattiva qualità [Figura 3.6]. L'approccio suggerito si basa a livello macroscopico su due operazioni geometriche: calcolo del poligono convesso (Convex Hull) contenente tutte le minuzie e triangolazione di Delaunay. La procedura può essere riassunta ad alto livello nei seguenti passi:

- calcolo del Convex Hull dato il vettore delle minuzie estratte: con questa operazione si ricava il poligono convesso di superficie minima contenente tutte le minuzie;
- applicazione della triangolazione di Delaunay sull'insieme delle minuzie estratte: in geometria questa operazione applicata ad un insieme di punti dà luogo ad una collezione di triangoli in cui il cerchio circoscritto ad ognuno di essi è vuoto, cioè nessun punto vi si trova all'interno;
- valutazione di condizioni basate su tre parametri cardine: area, perimetro e rapporto perimetro/area dei triangoli ottenuti. Vengono qui introdotte delle soglie indicative che determinano il grado di influenza di ogni triangolo, e di conseguenza di ogni

minuzia che ne fa parte, sulla qualità globale dell'immagine. Il valore risultante è unico e assegna al *set* di minuzie in esame (e quindi all'impronta corrispondente) un punteggio complessivo nell'intervallo $[0;100]$ in cui 0 indica il valore minimo di qualità mentre 100 quello massimo.

L'algoritmo è stato interamente implementato in C# in ambiente di sviluppo Visual Studio 2015, servendosi delle funzionalità messe a disposizione dalle librerie BioLab.dll e BioLab.Biometrics.dll sviluppate all'interno del laboratorio. Ultimata l'implementazione si è dato avvio ad una fase di *testing* per verificarne efficienza e prestazioni.

Gli esperimenti sono stati condotti sul *dataset* FVC2004 DB1_A utilizzando sei diversi estrattori (si veda la Sezione 3.1.2) oltre al Ground Truth corrispondente (GT_FM3). Inizialmente si è proceduto al calcolo dell'indice di qualità MQF per ogni *template* di ciascuno degli estrattori di minuzie. Una volta ottenuti questi risultati, seguendo i *test* menzionati nell'articolo [1], è stato calcolato l'indice di correlazione fra i valori di MQF e in successione quelli di NFIQ ed NFIQ 2.0. Come parametri di riferimento per area, perimetro e rapporto perimetro/area sono stati utilizzati gli stessi valori indicati nella pubblicazione (rispettivamente 75, 70 e 0.8). Trattandosi però di numeri indicativi, si è scelto di selezionare un intervallo di valori per ognuno e di testare tutte le possibili combinazioni. In questo modo è stato possibile osservare l'andamento della correlazione al variare dei tre parametri. Per area e perimetro sono stati testati valori compresi fra 50 e 100, mentre per il loro rapporto quelli fra 0.5 e 1.5. Non esponiamo qui tutti i grafici in quanto si avrebbe una corposa lista di valori che potrebbe occupare svariate pagine. Forniamo però una valutazione complessiva sull'analisi degli esiti, valutazione che si può intendere come di passaggio per successivi esperimenti. Al termine della computazione, i valori di MQF si sono rilevati piuttosto "scorrelati" rispetto a quelli di NFIQ, non riuscendo a superare la soglia dello 0.23; diversi sono stati invece gli esiti in raffronto ad NFIQ 2.0, dove gli indici di correlazione ottenuti oscillano fra un minimo di 0.45 ed un massimo di 0.60.

Avendo intravisto risultati potenzialmente significativi, si è deciso di verificare le prestazioni degli estrattori combinati con la metrica MQF. Per questo tipo di valutazione

ci si è serviti dell'algorithmo Minutia Cylinder-Code (MCC) [18] per la rappresentazione e il confronto di impronte digitali sviluppato e messo a disposizione dal laboratorio.

I *test* condotti hanno avuto un duplice scopo:

- da un lato si è valutato il comportamento di MQF in relazione ad ogni estrattore, con l'obiettivo di arrivare ad avere un quadro generale sui principali beneficiari di questa tecnica;
- parallelamente l'attenzione si è focalizzata sui singoli. Si sono poste a confronto le prestazioni raggiunte a seguito dell'applicazione di NFIQ, NFIQ 2.0 e MQF su ogni estrattore e questo ha permesso di verificare quale fosse per ognuno di essi la metrica in grado di fornire risultati migliori.

I primi *test* sono stati effettuati nel modo seguente:

1. è stato calcolato l'indice MQF sui *templates* di ogni estrattore;
2. per ogni impronta si è proceduto ad ordinare le relative immagini sulla base degli *scores* assegnati al punto precedente;
3. si sono valutate le prestazioni del sistema calcolando EER ed FMR per ogni estrattore.

Spendiamo qui qualche parola sui due indici menzionati. FMR (False Match Rate) indica la frequenza con cui il sistema dà accesso ad *impostors*, ossia ad utenti non abilitati, mentre FNMR (False Non Match Rate) segnala la frequenza con cui viene negato un accesso a soggetti *genuines* (gli aventi diritto). EER indica la percentuale di errore che si ha quando FMR è uguale a FNMR. Dunque un sistema è tanto più performante quanto più bassi sono i valori di EER conseguiti.

Vengono mostrati in Figura 4.1 i risultati ottenuti.

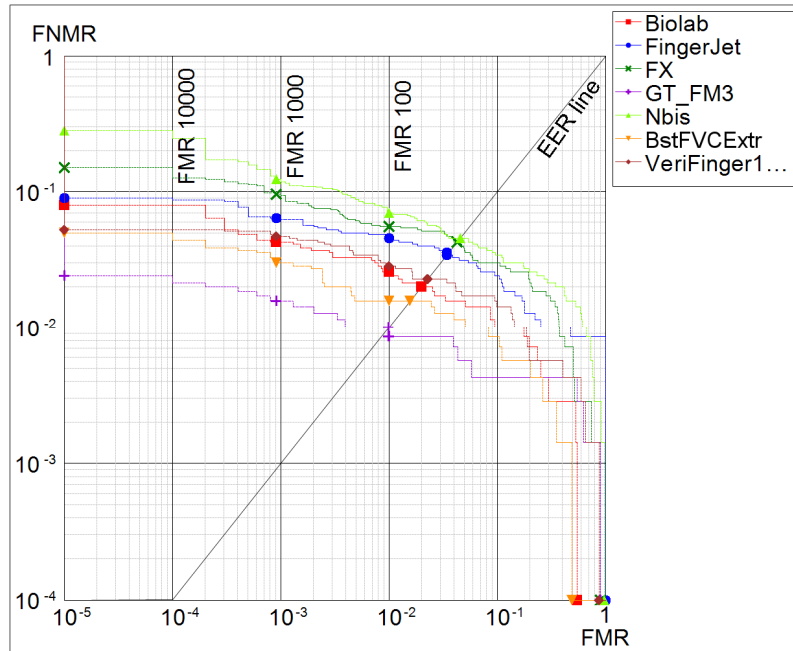
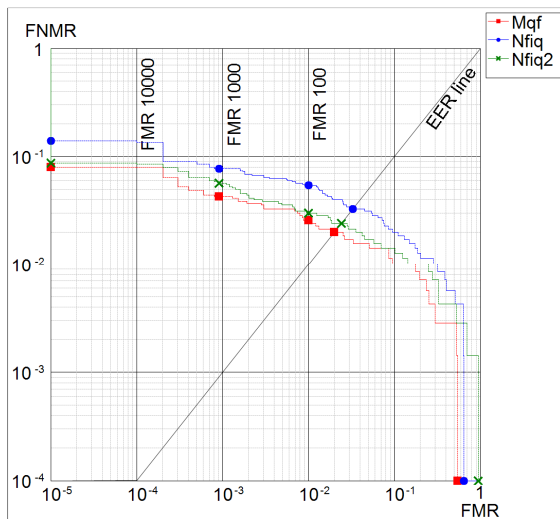
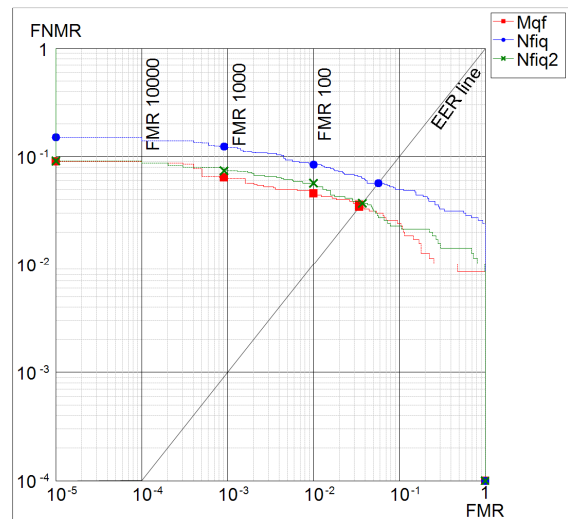


Figura 4.1: Curve DET in seguito all'applicazione di MQF sui *templates* di ogni estrattore

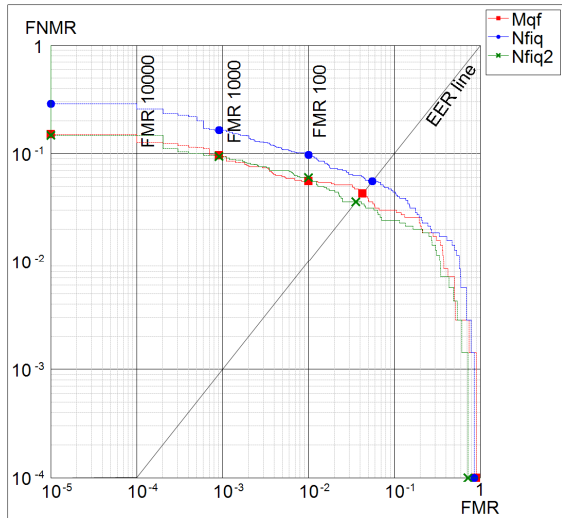
Il procedimento adottato per valutare le prestazioni sui singoli è del tutto analogo a quello precedente: il *focus* è stato però posto sull'efficacia delle tre metriche MQF, NFIQ ed NFIQ 2.0 su ogni estrattore. In pratica si sono valutate le prestazioni del sistema in seguito all'ordinamento qualitativo delle immagini realizzato sulla base degli *scores* assegnati dai tre algoritmi. Le immagini seguenti [Figura 4.2] mostrano gli esiti raggiunti.



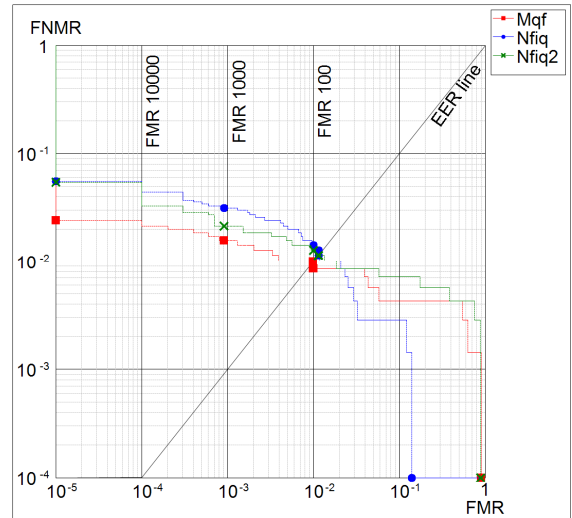
(a) BioLab



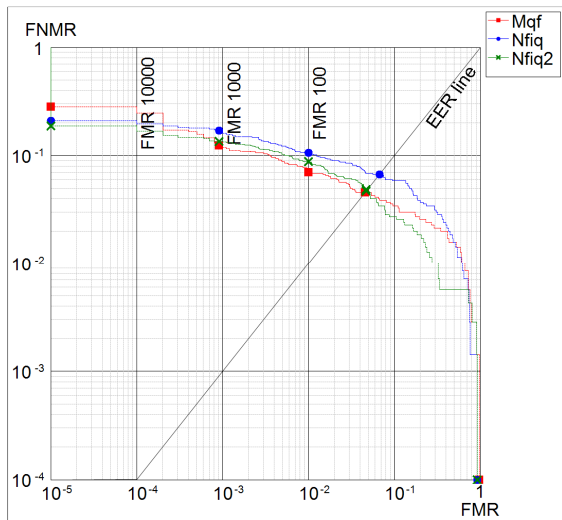
(b) FingerJet



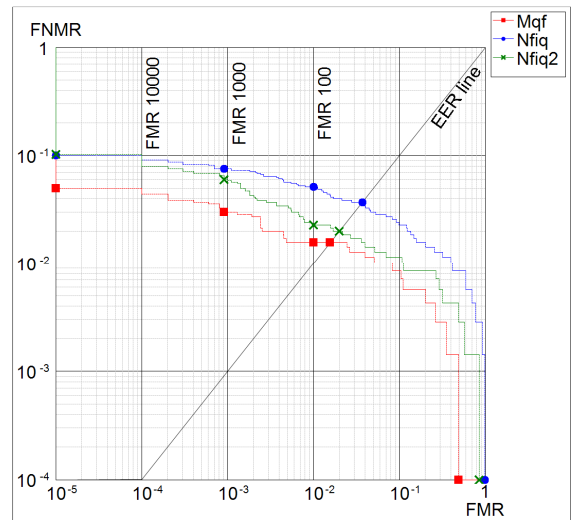
(c) FX



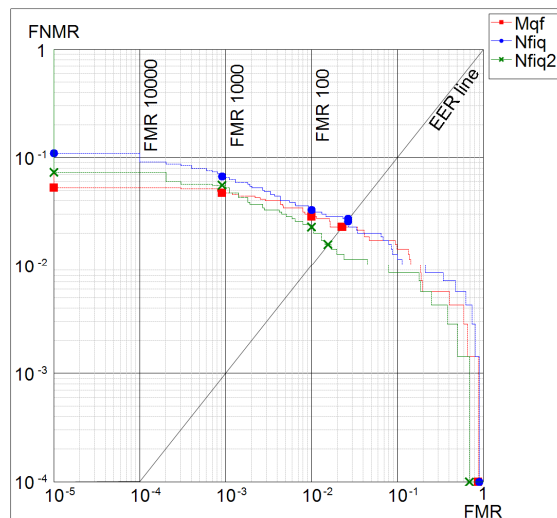
(d) GT_FM3



(e) Nbis



(f) BstFVCEXtr



(g) VeriFinger 10.0

Figura 4.2: Curve DET in seguito all'applicazione di MQF sui *templates* dei singoli estrattori

Sull'onda dei risultati generalmente promettenti ottenuti da questa seconda fase di *test*, ci si è così chiesti se questa metrica di qualità, in grado di fornire una valutazione di tipo globale sull'impronta, non potesse essere applicata per offrire un giudizio locale su ogni singola minuzia. Si sono così implementate una serie di applicazioni con l'obiettivo di verificare se questa intuizione fosse fondata, oltre che per comprendere quali fossero i casi più emblematici da prendere in considerazione.

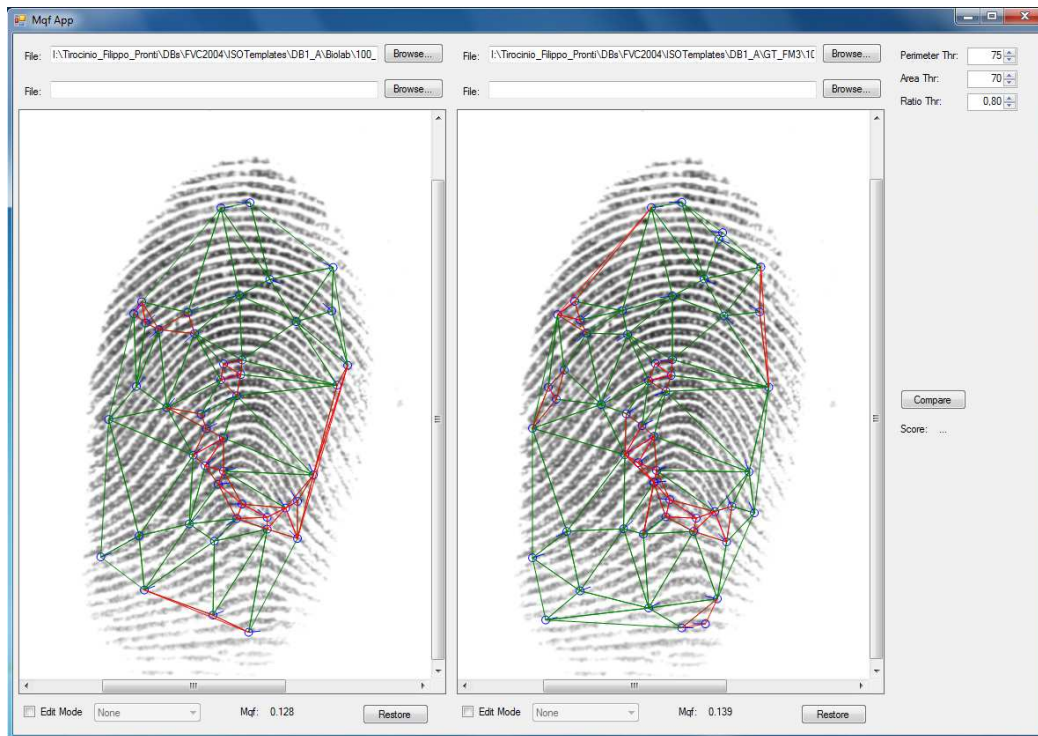


Figura 4.3: Esempio di applicazione per lo studio del comportamento di MQF

La Figura 4.3 riporta un esempio di applicazione che è stata realizzata per un approccio visivo all'analisi: l'immagine mostra la triangolazione di Delaunay applicata al *set* di minuzie di due diversi *templates* ricavati dalla stessa impronta. I triangoli segnalati in verde sono quelli non considerati dall'algoritmo in quanto aventi area e perimetro al di sopra di una determinata soglia; quelli rossi, al contrario, sono quelli che hanno una ricaduta sul calcolo della metrica MQF. Sperimentando varie combinazioni a livello grafico e cercando di porsi in condizioni limite per valutare il comportamento dell'algoritmo ove ad esempio la qualità delle immagini era peggiore, si è intravisto un possibile percorso di sperimentazione e si è deciso di muoversi in questa direzione.

L'idea principale è stata quella di provare a considerare, per ogni minuzia, i triangoli in cui essa era coinvolta, andando ad effettuare su di essi una serie di manipolazioni che

restituissero un indice di qualità. Sono stati calcolati diversi indicatori locali partendo dalle seguenti informazioni: media delle aree, media dei perimetri, aree massime e minime, perimetri massimi e minimi. Con i valori ottenuti da queste combinazioni di parametri, ci si è poi serviti dell’algoritmo *Minutiae Cylinder-Code* (MCC) [18] per valutare le prestazioni del sistema. Queste ultime sono state misurate fissando una soglia locale basata sulle caratteristiche geometriche appena descritte e scartando le minuzie al di sotto di questa *threshold*. Si sono poi confrontati gli esiti conseguiti attraverso l’impiego dei diversi approcci. Purtroppo, però, i risultati non sono stati quelli sperati. Sulla base di questo fallimento si è tentato infine un ultimo esperimento. Il tentativo effettuato è stato quello di combinare le informazioni locali legate ad area massima e perimetro massimo con l’indice di coerenza delle orientazioni locali. Per cercare di identificare il grado di influenza di ogni singolo parametro su questa stima locale, si è ricorsi all’introduzione di pesi: per avere un valore normalizzato tra 0 e 1, si è posta arbitrariamente la somma dei tre uguale ad 1. Il punteggio è stato calcolato con la seguente formula:

$$score_m = r_m * w1 + max\ area_m * w2 + max\ perimeter_m * w3$$

Anche in questo caso si è deciso di effettuare gli esperimenti in un intervallo di valori per vedere quali fossero le combinazioni migliori. Per ogni peso w_n sono stati testati tutti i valori tra 0 e 1, ma anche in questo caso non si è giunti ad alcun risultato soddisfacente. Il fallimento anche di quest’ultimo tentativo ha di fatto dimostrato l’inapplicabilità di MQF come metrica di qualità locale, motivo per cui ci si è successivamente mossi in direzione di approcci diversi.

4.2 Risultati ottenuti

Forniamo qui una panoramica riassuntiva sulle prove effettuate fino ad ora ed elaboriamo qualche considerazione. MQF, sulla base dei risultati di una prima fase di *test*, ha dimostrato di essere una buona metrica per valutare un *template* nella sua interezza. In questo senso ne è una prova tangibile il fatto che presenti nel complesso indici di correlazione piuttosto elevati rispetto a NFIQ 2.0. Inoltre, come si è apprezzato dai grafici [Figura 4.2], le prestazioni del sistema tendono in più di una situazione a miglioramenti se comparate con quelle relative all’utilizzo di NFIQ ed NFIQ 2.0. Un caso emblematico è ad

esempio quello mostrato in Figura 4.2 (f), dove sull'estrattore BstFVCEXtr si riscontrano risultati migliorativi per tutte le classi di NFIQ. In generale, si registrano prestazioni più positive in corrispondenza di immagini di alta qualità, in particolare per le classi di NFIQ 1 e 2. Se quindi in una prima fase di sperimentazioni si è di fatto dimostrata l'efficacia di questa tecnica, applicata così come descritta nell'articolo [1], non si può dire la stessa cosa per le prove successive. L'idea di servirsi della triangolazione di Delaunay per formulare valutazioni a livello di singola minuzia non ha apportato benefici in termini di prestazioni. Posto un ipotetico valore "q" relativo alla qualità della minuzia, questo fallimento si è materializzato a fronte dei tentativi di calcolarlo basandosi sulle seguenti informazioni:

- perimetro massimo;
- perimetro minimo;
- area massima;
- area minima;
- media della somma dei perimetri;
- media della somma delle aree.

Ulteriore prova di questo fallimento si è avuta combinando gli indici di qualità calcolati come descritto ai punti precedenti con il già efficace e sicuramente molto attendibile indice di coerenza. Come già anticipato nella Sezione 4.2, questo ultimo esperimento è stato decisivo nel determinare in modo ancora più definitivo l'impossibilità di avvalersi della metrica per formulare considerazioni a livello locale. In conclusione questa tecnica, basata sulla disposizione spaziale delle minuzie all'interno del *template*, si è rivelata nel complesso molto valida e ha dimostrato di poter competere con algoritmi già molto affidabili basati su approcci diametralmente opposti (NFIQ ed NFIQ 2.0 considerano solamente la qualità dell'immagine). Non si può però ovviamente tralasciare il fatto che le prove effettuate siano state condotte su FVC2004 DB1_A e che per determinare una sua validità a livello universale occorrerebbero *test* nell'ordine di migliaia di immagini (si veda la Sezione 3.2.1). Possiamo considerare il lavoro effettuato come punto di partenza per successive attività di ricerca per quanto riguarda l'applicazione di MQF così come concepita dai suoi autori, mentre constatiamo una sua inapplicabilità nel tentativo di servirsene per trarne considerazioni a livello locale.

5 Algoritmo ibrido

5.1 Le prove effettuate

Nel corso di questa fase della sperimentazione si ha un decisivo cambio di direzione rispetto all'approccio adottato in precedenza. Se con MQF lo scopo perseguito era quello di andare ad identificare minuzie scadenti basandosi su un valore di stima locale, qui lo scenario cambia. L'attenzione viene posta principalmente sull'analisi di scheletro ed *enhanced image* [Sezione 2.3] utilizzati per l'estrazione delle minuzie. Le operazioni e gli algoritmi che si vanno ad illustrare e ad implementare rientrano nella categoria delle tecniche di *minutiae filtering*. Come nel caso di MQF si tratta di operazioni di *post processing* e potremmo definirle quindi come misure correttive applicate al termine degli altri processi per ridurre il numero di artefatti formati in corso di acquisizione e digitalizzazione. L'algoritmo che si è scelto di implementare e da cui si sono estrapolate le procedure testate è quello proposto da Jabeen S. & Khan S. [2].

L'algoritmo descritto si basa sull'attuazione di quattro macro procedure, applicabili in cascata senza un ordine prestabilito, ed ha come obiettivo quello di eliminare il maggior numero di false minuzie cercando di arginare d'altro canto la rimozione di quelle corrette. Nel testo viene riportata una classificazione delle principali tipologie di minuzie considerate spurie, di cui si fornisce una catalogazione in Figura 5.1. Fra queste si possono distinguere le seguenti strutture:

- *bridge*: si ha quando intercorre un ponte fra due *ridges* parallele poste a distanza molto ravvicinata. Questa struttura dà origine a due false biforcazioni;
- *ridge break*: si viene a creare una fessura all'interno di una stessa *ridge* (solitamente dovuta a presenza di rumore in fase di acquisizione) che dà origine a due false terminazioni;
- *spike*: trattasi di una piccola insenatura che sorge su una *ridge* con conseguente formazione di una falsa biforcazione ed una falsa terminazione.

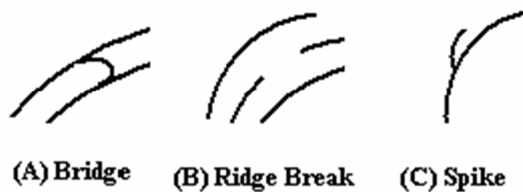


Figura 5.1: A partire da sinistra esempi di Bridge, Ridge Break e Spike

Il primo passo nell'implementazione di questo algoritmo è stato la realizzazione del cosiddetto *Boundary Elimination* basato su *Plus Rule*: questa procedura consiste nella rimozione di tutte le minuzie collocate ai bordi dell'immagine in quanto solitamente etichettate come false terminazioni.

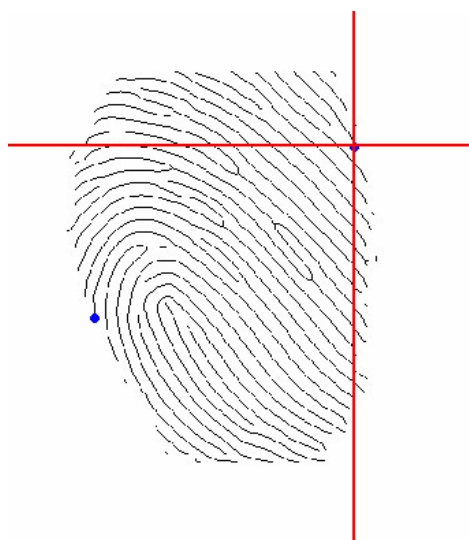


Figura 5.2: Il tipico *pattern* “+” utilizzato dalla Plus Rule

I *pixels* nell'intorno di ogni minuzia vengono scansionati seguendo il *pattern* disegnato da un segno “+” [Figura 5.2] apposto sulla minuzia (da qui il nome dell'algoritmo). Il procedimento si muove alla ricerca di almeno un *pixel* nero (valore 0) nelle quattro direzioni; se anche solo in una di esse questo non viene individuato, la minuzia è considerata *boundary* e viene rimossa.

Poiché una funzionalità analoga era già presente nell'algoritmo di estrazione delle minuzie oggetto di questo lavoro di tesi, i *test* iniziali si sono concentrati sul confronto dei risultati ottenuti comparando le due diverse procedure. Per effettuare un raffronto diretto si sono valutate le prestazioni dell'estrattore in seguito all'utilizzo delle due diverse implementazioni. L'estrattore è fortemente modularizzato e permette

l'abilitazione/disabilitazione di ogni singola procedura in fase di estrazione. Dunque i passi percorsi sono stati in sequenza:

1. estrazione dei *templates* con BioLab;
2. estrazione dei *templates* con BioLab con funzione di eliminazione delle minuzie ai bordi disabilitata;
3. applicazione del Boundary Elimination sui *templates* estratti al punto precedente;
4. confronto fra i *templates* ottenuti ai punti 1 e 3 attraverso il già utilizzato Minutiae Cylinder-Code (MCC) [18].

L'esito di questa prima fase di *test* ha posto in luce una generale equivalenza fra le due metodologie, con risultati tendenzialmente più performanti a seguito dell'utilizzo dell'implementazione di BioLab. Portati a termine questi primi tentativi, sono state implementate anche le ulteriori tre procedure dell'algoritmo. In fase di realizzazione ci si è però resi conto di alcune incongruenze rintracciate all'interno dello pseudocodice proposto dagli autori dell'articolo e si è proceduto ad alcune piccole correzioni rispetto alla formulazione originale. Le procedure sono di seguito illustrate (α , β , γ , λ sono parametri numerici introdotti nell'articolo [2]).

Bridge Elimination

Considerata una biforcazione “i” del *template*, l'algoritmo itera su tutte le altre biforcazioni “j” e per ognuna:

1. calcola distanza euclidea e differenza dell'angolo di orientazione [Sezione 2.3].
Ricava infine un valore “d” ottenuto come il prodotto delle precedenti misure;
2. nel caso in cui “d” sia inferiore ad una soglia definita come $\beta * \lambda$, la minuzia “j” viene aggiunta alla lista delle candidate.

Al termine della computazione, se la lista delle candidate presenta almeno un elemento, vengono rimosse la minuzia “i” e la minuzia appartenente alla lista delle candidate posta a distanza euclidea minore. Questo procedimento viene ovviamente iterato per tutte le biforcazioni presenti nel *template*.

Ridge Break Elimination

Analogamente alla procedura precedente, considerata una terminazione “i” del *template*, l’algoritmo itera su tutte le altre terminazioni “j” e per ognuna:

1. calcola distanza euclidea e differenza dell’angolo di orientazione [Sezione 2.3].
Ricava infine un valore “d” ottenuto come il prodotto delle precedenti misure;
2. nel caso in cui “d” sia inferiore ad una soglia definita come $\alpha * \lambda$, la minuzia “j” viene aggiunta alla lista delle candidate.

Al termine della computazione, se la lista delle candidate presenta almeno un elemento, vengono rimosse la minuzia “i” e la minuzia appartenente alla lista delle candidate posta a distanza euclidea minore. Questo procedimento viene ovviamente iterato per tutte le terminazioni presenti nel *template*.

Spikes Elimination

Considerata una terminazione “i” del *template*, l’algoritmo itera su tutte le altre biforcazioni “j” e per ognuna:

1. calcola distanza euclidea e differenza dell’angolo di orientazione [Sezione 2.3].
Ricava infine un valore “d” ottenuto come il prodotto delle precedenti misure;
2. nel caso in cui “d” sia inferiore ad una soglia definita come $\gamma * \lambda$, la minuzia “j” viene aggiunta alla lista delle candidate.

Al termine della computazione, se la lista delle candidate presenta almeno un elemento, vengono rimosse la minuzia “i” e la minuzia appartenente alla lista delle candidate posta a distanza euclidea minore. Questo procedimento viene ovviamente iterato per tutte le terminazioni presenti nel *template*.

Ancora una volta, per poter meglio osservare gli effetti sortiti dall’attuazione delle diverse componenti, si è ritenuto opportuno lo sviluppo di un’applicazione grafica [Figura 5.3] che mostrasse il comportamento dell’algoritmo in ogni suo modulo.



Figura 5.3: A sinistra, le minuzie del *template* di partenza posizionate sullo scheletro delle *ridges*; al centro e a destra i risultati ottenuti dall'applicazione dell'algorithm sulle minuzie iniziali.

I parametri configurabili che figurano nella schermata acquisita sono quelli proposti all'interno della pubblicazione ed indicati come valori *standard*: in particolare con il valore λ (lambda nell'immagine) viene indicata la distanza media fra *ridges* parallele in un'immagine acquisita a 500 DPI. Per il calcolo di questo parametro sono state proposte diverse procedure nel corso degli anni, ma in linea di massima questo valore può essere considerato attendibile e piuttosto ponderato. L'ausilio di uno strumento grafico si è rilevato in questa occasione più che mai utile in quanto si sono potute osservare le singole minuzie via via rimosse al variare dei quattro parametri α , β , γ , λ . Dopo essersi accertati del corretto funzionamento dell'algorithm, ha avuto inizio un'ultima fase di *test* nella quale sono state verificate prestazioni ed efficienza. Per poter valutare il processo nella sua interezza ed avere un *feedback* anche sulle singole procedure, i *test* sono stati organizzati in modo da prendere in considerazione le seguenti cinque combinazioni: Boundary Elimination, Bridge Elimination, Break Elimination, Spikes Elimination, Complete Algorithm. Le prove condotte si sono mosse fondamentalmente in due direzioni: una volta a verificare il legame dei risultati ottenuti con i *templates* di Ground Truth, l'altra incentrata sulla valutazione delle prestazioni del sistema a fronte dell'applicazione dell'algorithm (ci si è serviti ancora una volta di Minutiae Cylinder-Code (MCC) [18]). I

templates di partenza sui quali è stato applicato il False Minutiae Removal Algorithm sono stati quelli ricavati dagli estrattori BioLab, FingerJet, Nbis e VeriFinger 10.0. Purtroppo, in modo del tutto inaspettato, dai risultati ottenuti si è registrato un generale peggioramento delle prestazioni dei sistemi; dato ancora più negativo è da riscontrare nel fatto che questo fenomeno abbia investito in modo trasversale tutti gli estrattori, chiudendo dunque le porte a successive sperimentazioni.

5.2 Risultati ottenuti

Riportiamo qui i risultati ottenuti a fronte delle prove appena descritte. Per valutare le prestazioni su ogni estrattore si sono utilizzati dapprima gli indici EER, FMR100, FMR1000 e FMR10000 e successivamente il Minutiae Cylinder-Code (MCC). Tutti gli schemi [Figura 5.4] sono così organizzati: la prima riga riporta gli esiti ottenuti con l'utilizzo dell'estrattore originario, mentre le seguenti considerano tutte le procedure prese singolarmente ed in seguito applicate simultaneamente. Gli estrattori su cui sono state effettuate le prove sono BioLab, FingerJet, Nbis e VeriFinger10.0. La prima divisione di *test* è stata condotta su FVC2004 DB1_A, mentre quelli successivi su DL02.

AlgName	GenuineCount	ImpostorCount	EER	FMR100	FMR1000	FMR10000
Biolab	9240	9730	50.00 %	71.35 %	72.58 %	74.33 %
Biolab_BoundaryElimination	9240	9730	50.00 %	71.29 %	72.44 %	73.83 %
Biolab_BridgeElimination	9240	9730	50.00 %	71.31 %	72.75 %	74.50 %
Biolab_RidgeBreakElimination	9240	9730	50.00 %	71.83 %	72.94 %	74.49 %
Biolab_SpikesElimination	9240	9730	50.00 %	71.46 %	72.62 %	73.21 %
Biolab_Complete	9240	9730	50.00 %	71.87 %	73.08 %	75.58 %

Extractor	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
Biolab	0.00 %	18.21 %	24.57 %	18.44 %	28.43 %
Biolab_BoundaryElimination	0.00 %	20.62 %	23.53 %	20.84 %	27.31 %
Biolab_BridgeElimination	0.00 %	19.03 %	22.83 %	19.20 %	26.43 %
Biolab_RidgeBreakElimination	0.00 %	25.05 %	19.99 %	25.15 %	23.05 %
Biolab_SpikesElimination	0.00 %	20.22 %	21.42 %	20.37 %	24.86 %
Biolab_Complete	0.00 %	29.39 %	15.78 %	29.35 %	18.28 %

(a) BioLab (FVC2004 DB1_A)

AlgName	GenuineCount	ImpostorCount	EER	FMR100	FMR1000	FMR10000
FingerJet	9240	9730	50.00 %	71.90 %	73.53 %	75.84 %
FingerJet_BoundaryElimination	9240	9730	50.00 %	71.98 %	73.37 %	75.98 %
FingerJet_BridgeElimination	9240	9730	50.00 %	72.05 %	73.50 %	76.99 %
FingerJet_RidgeBreakElimination	9240	9730	50.00 %	72.61 %	74.42 %	76.20 %
FingerJet_SpikesElimination	9240	9730	50.00 %	72.09 %	74.17 %	75.84 %
FingerJet_Complete	9240	9730	50.00 %	73.70 %	75.52 %	77.36 %

Extractor	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
FingerJet	0.75 %	19.48 %	51.56 %	19.26 %	61.20 %
FingerJet_BoundaryElimination	0.75 %	22.62 %	46.95 %	22.40 %	56.02 %
FingerJet_BridgeElimination	0.75 %	20.81 %	48.23 %	20.59 %	57.30 %
FingerJet_RidgeBreakElimination	0.75 %	26.74 %	43.06 %	26.44 %	51.01 %
FingerJet_SpikesElimination	0.75 %	22.67 %	44.32 %	22.59 %	52.83 %
FingerJet_Complete	0.75 %	33.03 %	32.01 %	32.82 %	38.36 %

(b) FingerJet (FVC2004 DB1_A)

AlgName	GenuineCount	ImpostorCount	EER	FMR100	FMR1000	FMR10000
NBis	9240	9730	50.00 %	73.11 %	75.47 %	77.80 %
NBis_BoundaryElimination	9240	9730	50.00 %	72.91 %	74.79 %	77.16 %
NBis_BridgeElimination	9240	9730	50.00 %	73.11 %	75.47 %	77.80 %
NBis_RidgeBreakElimination	9240	9730	50.00 %	73.11 %	75.47 %	77.80 %
NBis_SpikesElimination	9240	9730	50.00 %	73.11 %	75.47 %	77.80 %
NBis_Complete	9240	9730	50.00 %	72.91 %	74.79 %	77.16 %

Extractor	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
NBis	0.00 %	93.41 %	144.21 %	93.32 %	154.40 %
NBis_BoundaryElimination	0.00 %	93.94 %	119.71 %	93.88 %	128.97 %
NBis_BridgeElimination	0.00 %	93.41 %	144.21 %	93.32 %	154.40 %
NBis_RidgeBreakElimination	0.00 %	93.41 %	144.21 %	93.32 %	154.40 %
NBis_SpikesElimination	0.00 %	93.41 %	144.21 %	93.32 %	154.40 %
NBis_Complete	0.00 %	93.94 %	119.71 %	93.88 %	128.97 %

(c) Nbis (FVC2004 DB1_A)

AlgName	GenuineCount	ImpostorCount	EER	FMR100	FMR1000	FMR10000
VeriFinger	9240	9730	50.00 %	70.82 %	71.93 %	73.80 %
VeriFinger_BoundaryElimination	9240	9730	50.00 %	71.01 %	72.16 %	74.45 %
VeriFinger_BridgeElimination	9240	9730	50.00 %	70.83 %	71.83 %	73.85 %
VeriFinger_RidgeBreakElimination	9240	9730	50.00 %	71.18 %	72.37 %	73.29 %
VeriFinger_SpikesElimination	9240	9730	50.00 %	70.96 %	72.16 %	73.90 %
VeriFinger_Complete	9240	9730	50.00 %	71.23 %	72.93 %	74.89 %

Extractor	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
VeriFinger	0.00 %	15.93 %	24.23 %	15.70 %	28.89 %
VeriFinger_BoundaryElimination	0.00 %	19.61 %	21.44 %	19.31 %	25.82 %
VeriFinger_BridgeElimination	0.00 %	16.40 %	23.29 %	16.21 %	27.59 %
VeriFinger_RidgeBreakElimination	0.00 %	21.76 %	22.03 %	21.59 %	26.14 %
VeriFinger_SpikesElimination	0.00 %	17.40 %	23.33 %	17.14 %	27.83 %
VeriFinger_Complete	0.00 %	26.70 %	17.99 %	26.47 %	21.44 %

(d) VeriFinger 10.0 (FVC2004 DB1_A)

AlgName	GenuineCount	ImpostorCount	EER	FMR100	FMR1000	FMR10000
Biolab	9240	9730	5.61 %	9.24 %	13.74 %	17.71 %
Biolab_BoundaryElimination	9240	9730	6.23 %	10.60 %	14.98 %	19.62 %
Biolab_BridgeElimination	9240	9730	5.96 %	9.57 %	13.44 %	18.00 %
Biolab_RidgeBreakElimination	9240	9730	8.30 %	13.17 %	16.34 %	19.47 %
Biolab_SpikesElimination	9240	9730	6.34 %	10.04 %	12.84 %	18.12 %
Biolab_Complete	9240	9730	9.86 %	14.65 %	17.13 %	21.20 %

Extractor	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
Biolab	0.00 %	19.64 %	34.72 %	19.52 %	35.54 %
Biolab_BoundaryElimination	0.00 %	22.79 %	33.05 %	22.72 %	33.81 %
Biolab_BridgeElimination	0.00 %	19.97 %	33.48 %	19.84 %	34.25 %
Biolab_RidgeBreakElimination	0.00 %	25.47 %	27.39 %	25.34 %	27.99 %
Biolab_SpikesElimination	0.00 %	21.36 %	31.05 %	21.18 %	31.74 %
Biolab_Complete	0.00 %	29.91 %	22.81 %	29.78 %	23.26 %

(e) BioLab (DL02)

AlgName	GenuineCount	ImpostorCount	EER	FMR100	FMR1000	FMR10000
FingerJet	9240	9730	14.73 %	22.23 %	26.85 %	30.85 %
FingerJet_BoundaryElimination	9240	9730	14.98 %	23.00 %	27.95 %	32.59 %
FingerJet_BridgeElimination	9240	9730	15.10 %	22.72 %	27.08 %	32.36 %
FingerJet_RidgeBreakElimination	9240	9730	16.71 %	25.47 %	30.35 %	34.29 %
FingerJet_SpikesElimination	9240	9730	16.51 %	25.39 %	30.29 %	32.95 %
FingerJet_Complete	9240	9730	19.11 %	29.73 %	35.44 %	39.77 %

Extractor	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
FingerJet	8.21 %	26.69 %	38.17 %	25.99 %	39.74 %
FingerJet_BoundaryElimination	8.21 %	29.46 %	36.23 %	28.84 %	37.72 %
FingerJet_BridgeElimination	8.21 %	27.53 %	36.27 %	26.84 %	37.81 %
FingerJet_RidgeBreakElimination	8.21 %	32.01 %	31.41 %	31.26 %	32.89 %
FingerJet_SpikesElimination	8.21 %	29.55 %	32.14 %	28.84 %	33.62 %
FingerJet_Complete	8.21 %	37.47 %	24.68 %	36.79 %	26.01 %

(f) FingerJet (DL02)

AlgName	GenuineCount	ImpostorCount	EER	FMR100	FMR1000	FMR10000
NBis	9240	9730	8.56 %	18.14 %	23.70 %	29.29 %
NBis_BoundaryElimination	9240	9730	9.67 %	20.64 %	26.21 %	31.16 %
NBis_BridgeElimination	9240	9730	8.56 %	18.14 %	23.70 %	29.29 %
NBis_RidgeBreakElimination	9240	9730	8.56 %	18.14 %	23.70 %	29.29 %
NBis_SpikesElimination	9240	9730	8.56 %	18.14 %	23.70 %	29.29 %
NBis_Complete	9240	9730	9.67 %	20.64 %	26.21 %	31.16 %

Extractor	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
NBis	0.00 %	93.75 %	121.23 %	93.88 %	122.28 %
NBis_BoundaryElimination	0.00 %	94.06 %	110.09 %	94.18 %	110.93 %
NBis_BridgeElimination	0.00 %	93.75 %	121.23 %	93.88 %	122.28 %
NBis_RidgeBreakElimination	0.00 %	93.75 %	121.23 %	93.88 %	122.28 %
NBis_SpikesElimination	0.00 %	93.75 %	121.23 %	93.88 %	122.28 %
NBis_Complete	0.00 %	94.06 %	110.09 %	94.18 %	110.93 %

(g) NBis (DL02)

AlgName	GenuineCount	ImpostorCount	EER	FMR100	FMR1000	FMR10000
VeriFinger	9240	9730	3.64 %	4.21 %	4.84 %	5.52 %
VeriFinger_BoundaryElimination	9240	9730	3.78 %	4.37 %	5.39 %	6.79 %
VeriFinger_BridgeElimination	9240	9730	3.64 %	4.23 %	4.85 %	5.88 %
VeriFinger_RidgeBreakElimination	9240	9730	3.87 %	4.69 %	5.56 %	6.27 %
VeriFinger_SpikesElimination	9240	9730	3.68 %	4.26 %	4.83 %	5.43 %
VeriFinger_Complete	9240	9730	4.16 %	5.08 %	6.16 %	8.05 %

Extractor	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
VeriFinger	0.00 %	16.89 %	20.79 %	16.68 %	21.45 %
VeriFinger_BoundaryElimination	0.00 %	20.92 %	15.86 %	20.76 %	16.39 %
VeriFinger_BridgeElimination	0.00 %	17.05 %	20.51 %	16.85 %	21.16 %
VeriFinger_RidgeBreakElimination	0.00 %	20.50 %	19.08 %	20.28 %	19.63 %
VeriFinger_SpikesElimination	0.00 %	18.04 %	20.04 %	17.81 %	20.65 %
VeriFinger_Complete	0.00 %	25.38 %	13.54 %	25.19 %	13.92 %

(h) VeriFinger 10.0 (DL02)

Figura 5.4: Prestazioni raggiunte sui due *database* DL02 ed FVC 2004 DB1_A

6 Ottimizzazione parametri dell'algoritmo di estrazione

6.1 I parametri considerati

L'attività d'indagine che è stata svolta inizialmente si è focalizzata sull'individuazione degli errori commessi da BioLab in fase di estrazione. La tipologia di approccio è molto simile a quella utilizzata nell'articolo [2]. Anche noi, con un procedimento di catalogazione degli errori più frequenti analogo a quello operato dagli ideatori dell'algoritmo ibrido, abbiamo proceduto all'individuazione di problematiche ricorrenti. La sottile differenza consiste nel fatto che qui l'obiettivo finale è non tanto l'individuazione dell'errore commesso, ma la ricerca dell'eziologia del problema. Ciò che qui interessa è infatti capire perché determinate situazioni tendano a ripetersi e per fare ciò è stato necessario categorizzare gli errori in classi, suddividendoli sulla base di *pattern* ricorrenti. Poiché però queste situazioni non erano note a priori, l'approccio adottato è stato di tipo euristico: ogni qual volta si è rintracciata una nuova problematica non ancora catalogata, questa è stata aggiunta all'elenco che è andato via via prendendo forma. Per avere un approccio visivo e per poter scandagliare i problemi legati ad ogni singola immagine, si è deciso di utilizzare come *tool* di supporto FLab [Figura 6.1], un applicativo grafico sviluppato all'interno del laboratorio e molto utile, soprattutto per la possibilità di modificare *real time* i parametri utilizzati dalle procedure di BioLab.

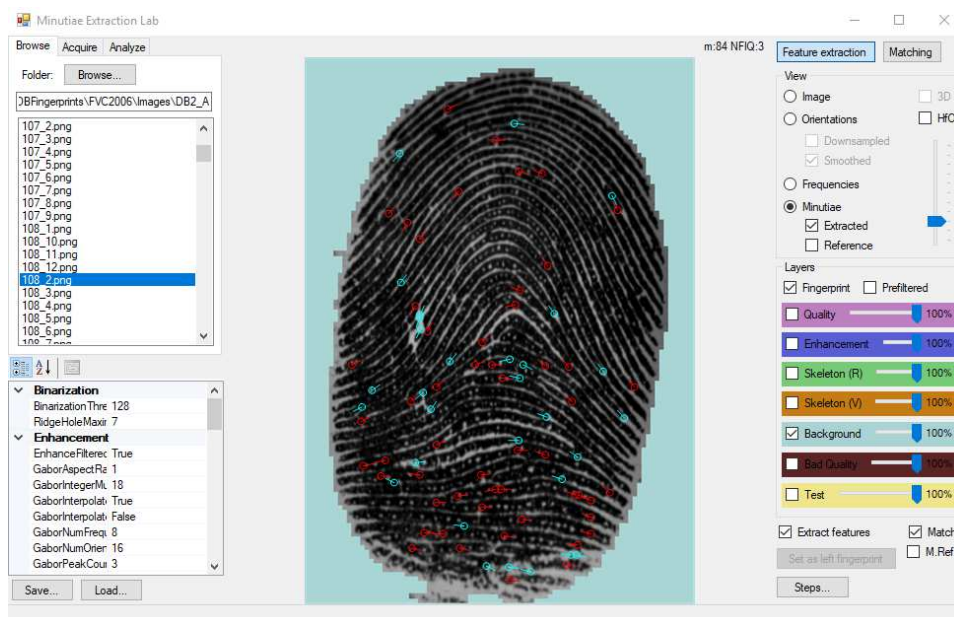


Figura 6.1: Impronta con NFIQ 3 e relative minuzie visualizzate all'interno di FLab

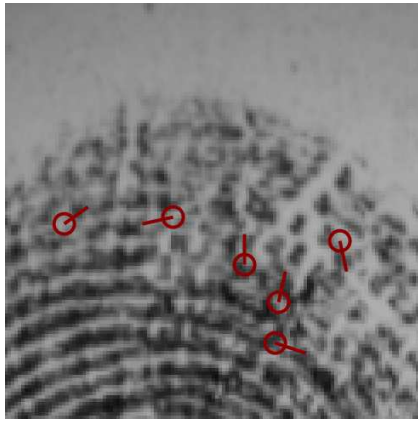
Un'importante funzione messa a disposizione dal programma è quella di poter sovrapporre al *template* di minuzie ottenuto sulla base del *set* di parametri specificato, un ulteriore *template* precedentemente estratto. Questo, nell'ottica di un'attività di indagine svolta principalmente ad occhio nudo, campione per campione, ha comportato una facilitazione del lavoro, soprattutto in fase di individuazione dei problemi relativi alle singole minuzie. Il lavoro di ricerca è stato condotto confrontando per ogni immagine *template* estratto da BioLab e *template* composto da *false* e *missed*. Si è organizzato il lavoro secondo lo schema mostrato in Figura 6.2.

	NFIQ	BAD QUALITY		LITTLE CUT		SKIMMING RULES		BIG CUT		WRONG BRIDGE	
		FALSE	MISSED	FALSE	MISSED	FALSE	MISSED	FALSE	MISSED	FALSE	MISSED
100_2	3		1				1				
100_3	3	1	1	1			1				
100_4	3	1	1				1			1	
11_8	3						1			1	
12_7	3	1								1	
12_8	3	1		1							
14_2	3						1				
14_7	3						1				
27_7	3						1				
27_8	3	1	1				1				
30_3	3	1								1	
30_4	3		1	1			1				
30_8	3										
33_8	3	1	1				1				

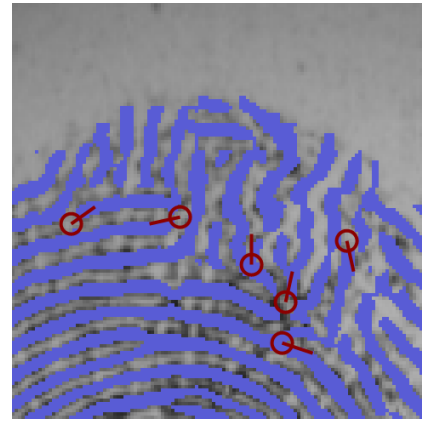
Figura 6.2: Approccio utilizzato in fase di raccolta dei dati

Partendo da sinistra, si hanno l'identificativo del campione esaminato e la categoria di NFIQ a cui questo afferisce (come già anticipato nella Sezione 3.2.1 ci focalizzeremo solamente su immagini di NFIQ 3, 4 e 5). Nelle colonne successive figurano le problematiche; con lo scopo di comprendere il grado di influenza di ogni singola tipologia di errore sull'occorrenza di *missed* o *false*, viene specificato inoltre se la problematica investe la prima o la seconda categoria. Ovviamente, ai fini di un'analisi finale da operare sul *database*, si è scelto di limitare il ventaglio di possibili casistiche ad un numero sufficientemente ragionevole. Si sono così identificate undici macro tipologie di problemi più comuni che andiamo a illustrare. Per convenzione, identifichiamo di seguito in rosso le minuzie *false* e in azzurro le *missed*.

Bad Quality

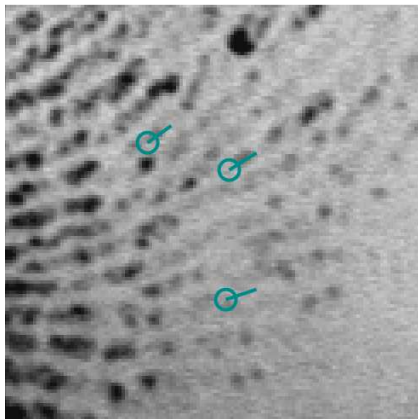


(a) Immagine originale

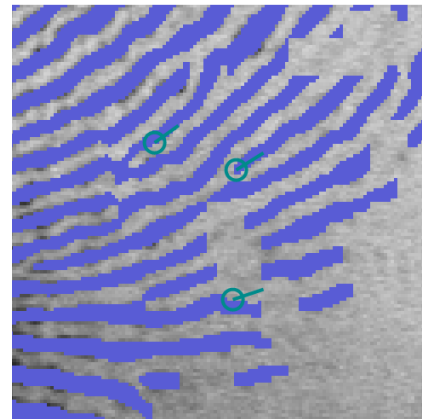


(b) Immagine *enhanced*

Figura 6.3: Bad Quality (esempio relativo a individuazione di *false*)



(a) Immagine originale



(b) Immagine *enhanced*

Figura 6.4: Bad Quality (esempio relativo a individuazione di *missed*)

In questa categoria figurano tutte le minuzie segnalate come non corrette (*false* e *missed*) rilevate in un'area di scarsa qualità. Come è possibile notare dagli esempi [Figura 6.3] [Figura 6.4], cercare di dedurre l'andamento delle creste è qui praticamente impossibile, anche attraverso un'attenta analisi visiva. Ne consegue la ricostruzione di una *enhanced image* del tutto irregolare. Nell'andare a identificare queste minuzie ci si è serviti anche dell'indicatore di aree di Bad Quality messo a disposizione da FLab.

Little Cut

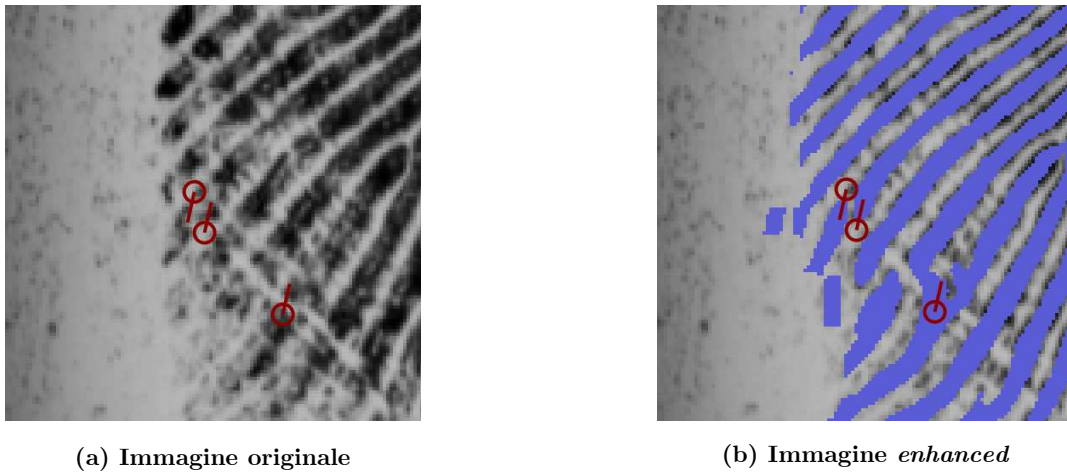


Figura 6.5: Little Cut

Ci si trova qui [Figura 6.5] in presenza di un taglio di piccola entità che solitamente crea una discontinuità sulla cresta, dando vita a due false terminazioni ravvicinate. È per natura un problema che origina solo *false*.

Skimming Rules

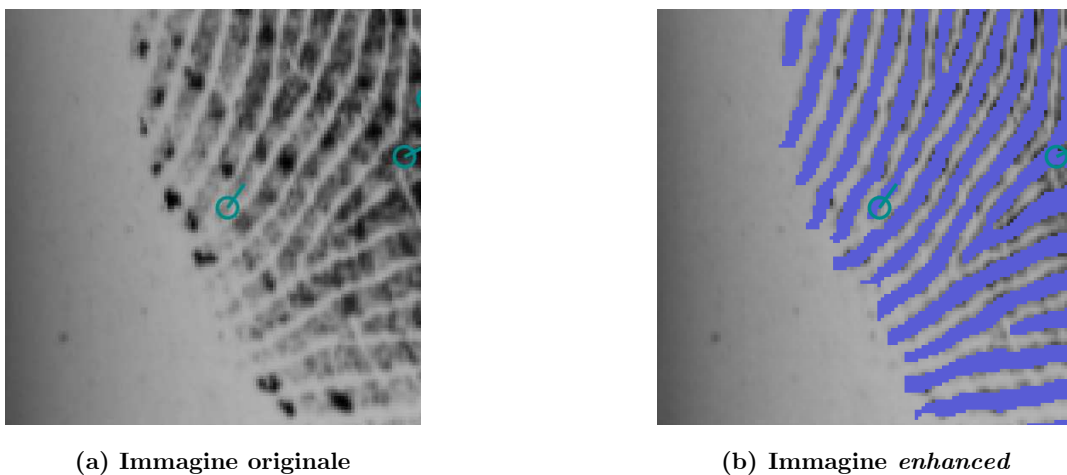


Figura 6.6: Skimming Rules

Come è possibile osservare, la minuzia sulla sinistra [Figura 6.6] dovrebbe essere estratta sulla base dell'*enhancing* effettuato; al contrario, a causa di regole di scrematura dell'algoritmo, questa è stata scartata. Spesso ciò che accade è che una minuzia possa essere rilevata troppo vicino ai bordi oppure molto vicina ad un'altra: sulla base di questi controlli, l'algoritmo decide di non considerarla. Per natura è un problema che riguarda solamente *missed*.

Big Cut

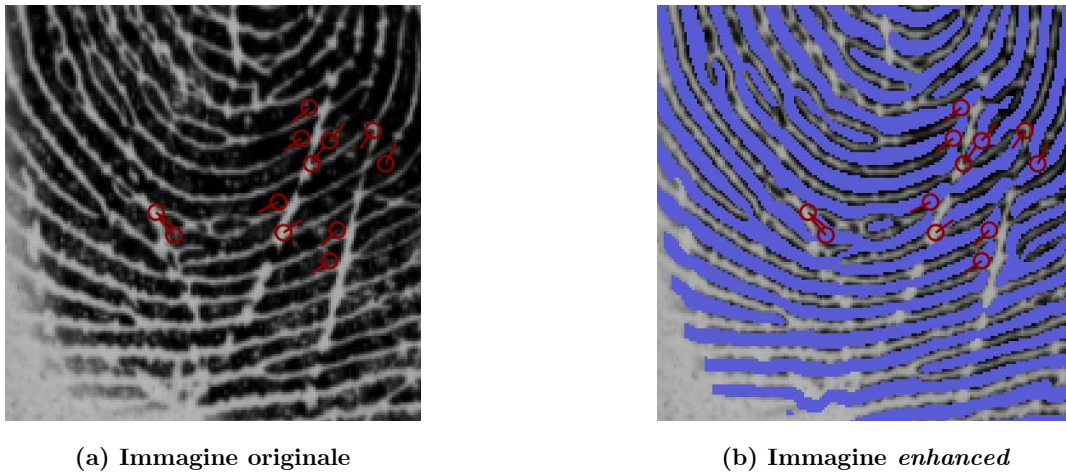


Figura 6.7: Big Cut

Situazione in cui compaiono tagli e lacerazioni della cute di dimensioni non trascurabili. Solitamente si creano solchi che tagliano trasversalmente le creste e che ne interrompono bruscamente l'andamento: in corrispondenza delle discontinuità si possono avere agglomerati di minuzie false dovute a ricongiungimenti errati [Figura 6.7].

Wrong Bridge

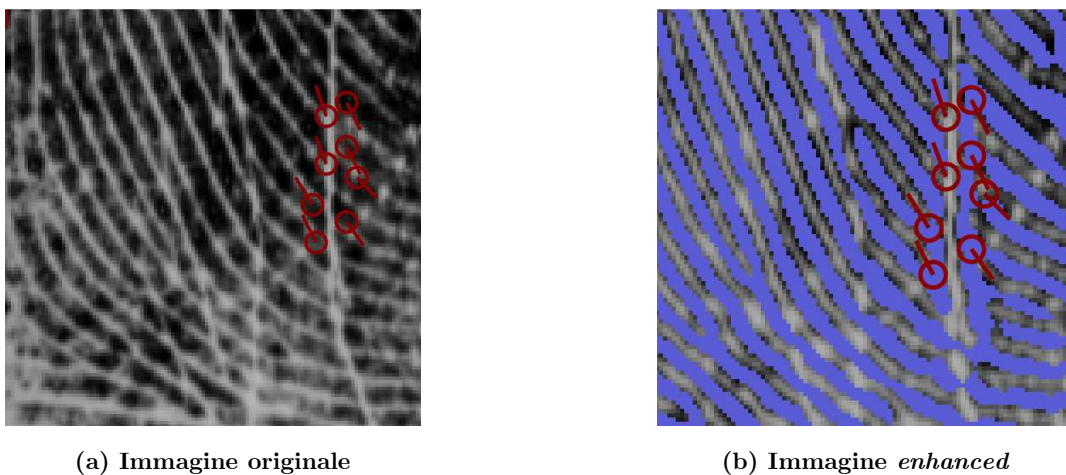


Figura 6.8: Wrong Bridge

Come suggerisce il nome assegnato a questa classe di errori, è una problematica legata a ponti che si creano erroneamente fra *ridges* parallele o comunque differenti. In questo caso specifico [Figura 6.8] il problema è causato da un taglio: l'algoritmo, non riuscendo a ricongiungere le due estremità, crea collegamenti fra creste parallele estraendo un conseguente numero di biforcazioni errate.

Birn From White

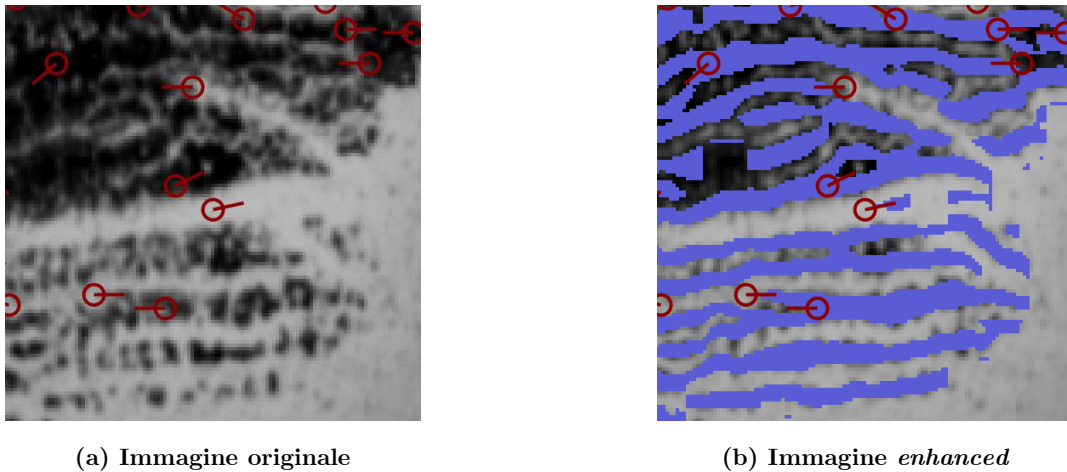


Figura 6.9: Birn From White

È questa una categoria che raccoglie più situazioni in cui si originano delle aree di *enhancing* molto piccole con conseguente estrazione di una o due false terminazioni. Nell'esempio riportato in Figura 6.9 la minuzia incriminata è quella centrale situata all'interno del grande solco bianco che taglia l'intera immagine. Come si nota in Figura 6.9 (a), in fase di *enhancing* si origina una piccola isola in cui viene segnalata una terminazione errata.

Spike

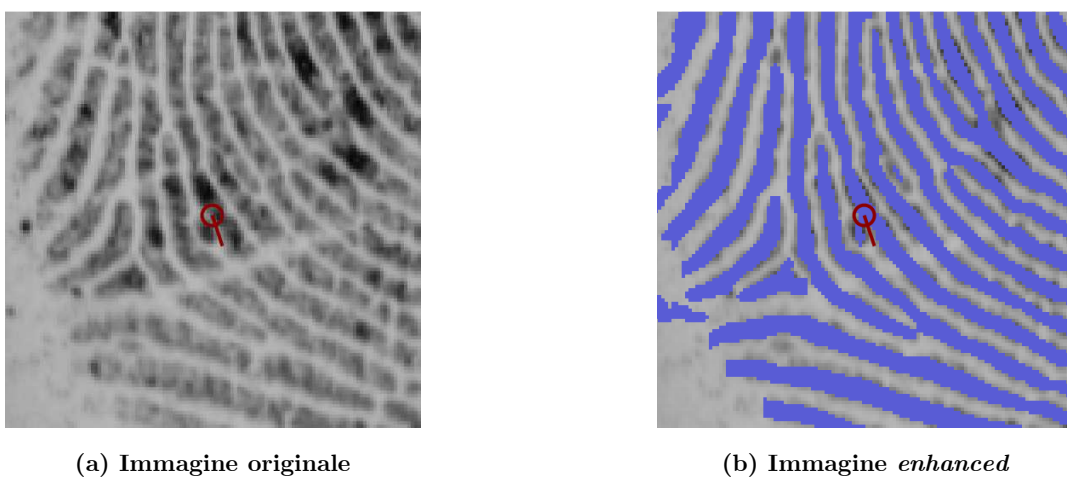


Figura 6.10: Spike

Situazione già nota in letteratura in cui una lieve impurità determina l'insorgere di una piccola insenatura [Figura 6.10]. Tipicamente comporta l'estrazione di due false minuzie, una biforcazione ed una terminazione.

Wrong Bifurcation

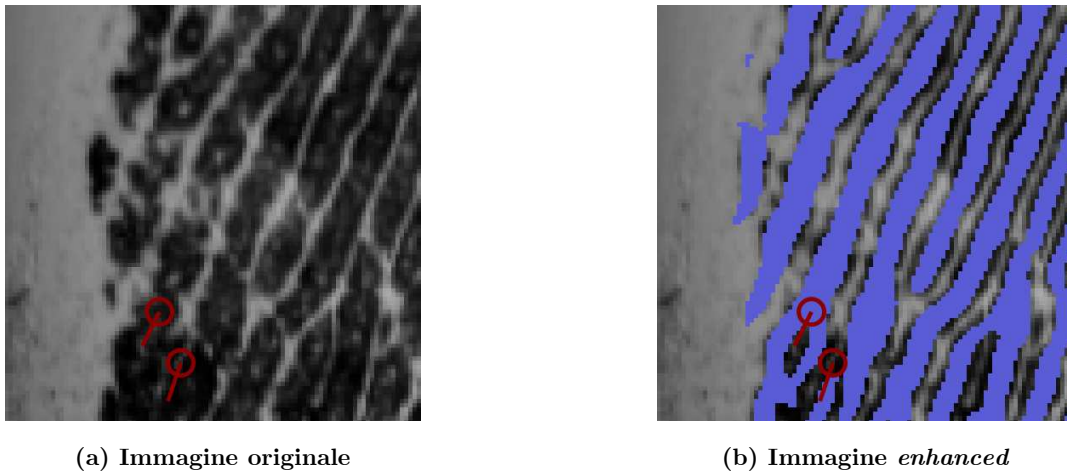


Figura 6.11: Wrong Bifurcation

Come apprezzabile dalla Figura 6.11, si crea una ramificazione errata che dà vita ad una falsa biforcazione sulla cresta considerata. Il *pattern* ricorrente è quello di un piccolo *lake* (così è stato ribattezzato in più di una pubblicazione) in cui figurano solitamente una o due *false* alle estremità.

Filled Island

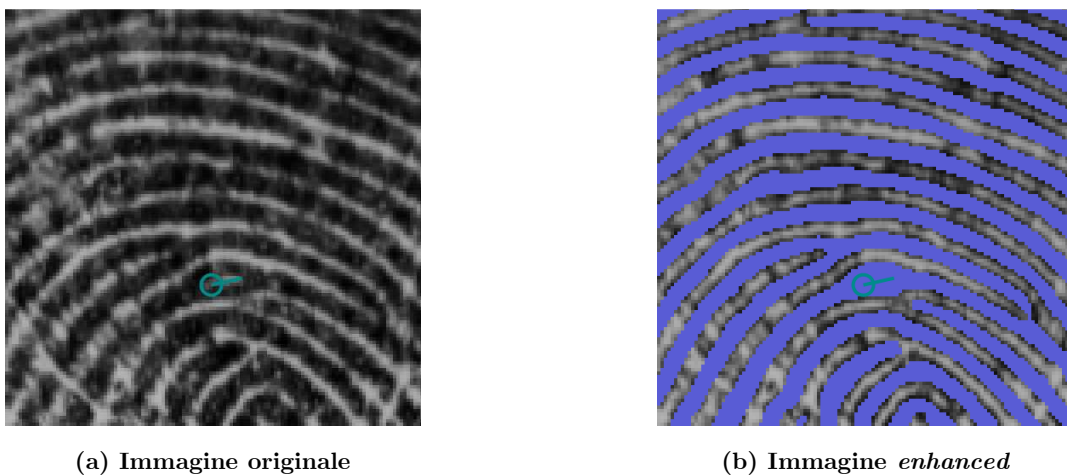
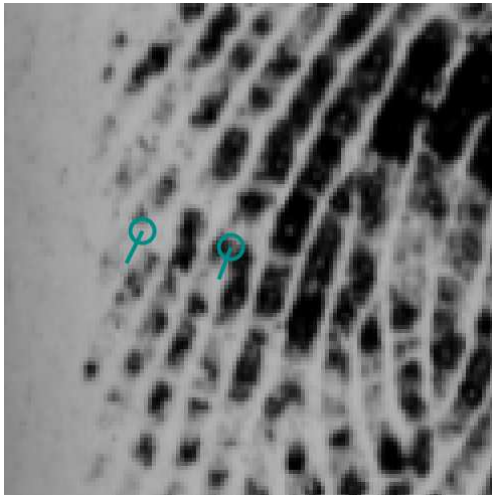


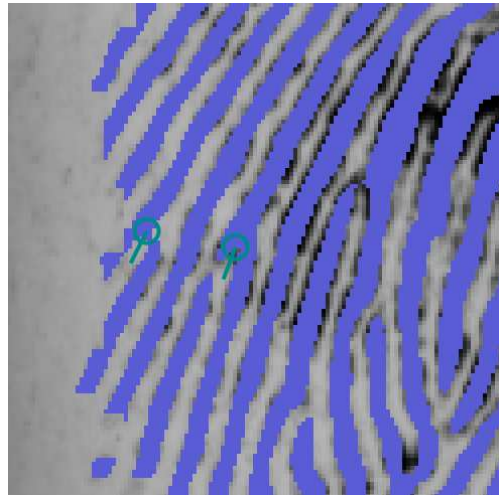
Figura 6.12: Filled Island

È un problema che comporta, per sua natura, l'insorgere di *missed*. Come si evince dalla Figura 6.12 (a), si ha un piccola area bianca incastonata all'interno di una *ridge*. Gli altri estrattori rilevano qui una biforcazione, mentre BioLab, da quanto si osserva nell'immagine a sinistra, colma questo piccolo *lake* in fase di *enhancement*. Dunque non viene rilevata in questo caso alcuna singolarità.

Missed Bridge



(a) Immagine originale

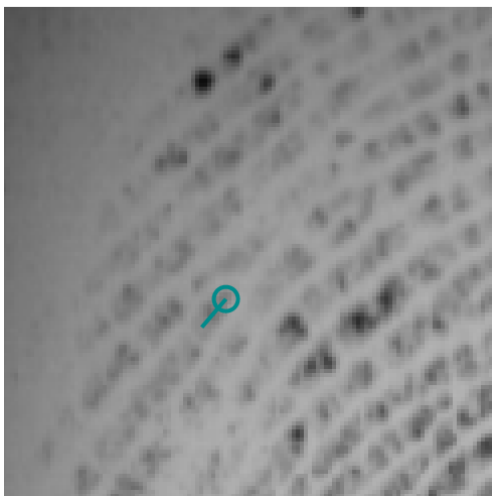


(b) Immagine *enhanced*

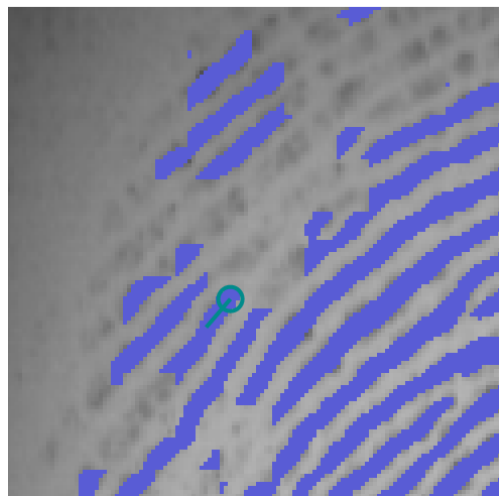
Figura 6.13: Missed Bridge

Anche in questo caso il problema è prettamente connesso alla segnalazione di *missed*. A causa di piccoli tagli, di zone aventi scarsa qualità o di errori verificatisi in fase di *enhancing*, hanno luogo mancati ricongiungimenti. Nel caso esibito [Figura 6.13] ci si concentra sulla minuzia centrale: siamo in presenza di quella che, ad una analisi a occhio nudo, è di fatto una biforcazione mancata: quest'ultima non viene rilevata in quanto l'algoritmo non riesce a colmare il piccolo taglio che intercorre.

Strange Break



(a) Immagine originale



(b) Immagine *enhanced*

Figura 6.14: Strange Break

L'appellativo deriva dal fatto che nei primi momenti in cui ci si è imbattuti in questa categoria di errori non fosse chiaro il perché di questo comportamento anomalo. Il problema deriva dal fatto che l'algoritmo incontra delle difficoltà in fase di estrazione, dovute principalmente alla presenza di aree di scarsissima qualità (si noti la sfocatura della Figura 6.14 (a)). Questo è un problema che per natura determina solitamente delle *missed* in quanto BioLab non rileva minuzie laddove gli altri estrattori riescono invece ad asserire l'andamento delle creste. Si è ritenuto però opportuno categorizzare qui anche minuzie *false* ogni qual volta queste si trovassero in prossimità di queste zone *not enhanced* e la loro formazione fosse conseguenza diretta di questo fenomeno.

Questo processo di classificazione ha portato alla compilazione di uno schedario in cui ad ognuno dei 1069 campioni sono state associate le relative problematiche. Le percentuali legate all'occorrenza di ogni situazione sono riportate di seguito ed hanno avuto un impatto importantissimo sulle successive direzioni prese in fase di studio dei parametri.

	BAD QUALITY		LITTLE CUT		SKIMMING RULES		BIG CUT		WRONG BRIDGE		BIRN FROM WHITE	
ALL	77,4%	49,5%	31,4%	0,8%	0,3%	61,7%	37,6%	1,1%	49,2%	6,8%	36,0%	0,0%
NFIQ3	72,8%	43,1%	28,1%	0,6%	0,4%	67,3%	37,1%	1,2%	51,6%	5,8%	33,6%	0,0%
NFIQ4	89,5%	60,7%	42,9%	1,8%	0,0%	53,9%	44,7%	0,9%	47,0%	11,4%	42,0%	0,0%
NFIQ5	82,9%	67,5%	30,9%	0,8%	0,0%	43,1%	27,6%	0,8%	39,0%	4,9%	39,8%	0,0%

	SPIKE		WRONG BIFURCAT.		ISLAND FILLED		MISSED BRIDGE		STRANGE BREAK	
ALL	61,1%	0,1%	22,8%	0,0%	0,0%	4,9%	13,2%	20,2%	3,2%	12,6%
NFIQ3	60,0%	0,1%	21,7%	0,0%	0,0%	5,1%	14,9%	20,5%	2,1%	12,9%
NFIQ4	68,9%	0,0%	24,2%	0,0%	0,0%	5,9%	10,5%	20,5%	4,1%	10,5%
NFIQ5	53,7%	0,0%	26,8%	0,0%	0,0%	1,6%	8,1%	17,9%	8,1%	14,6%

Figura 6.15: Statistiche relative all'occorrenza delle problematiche

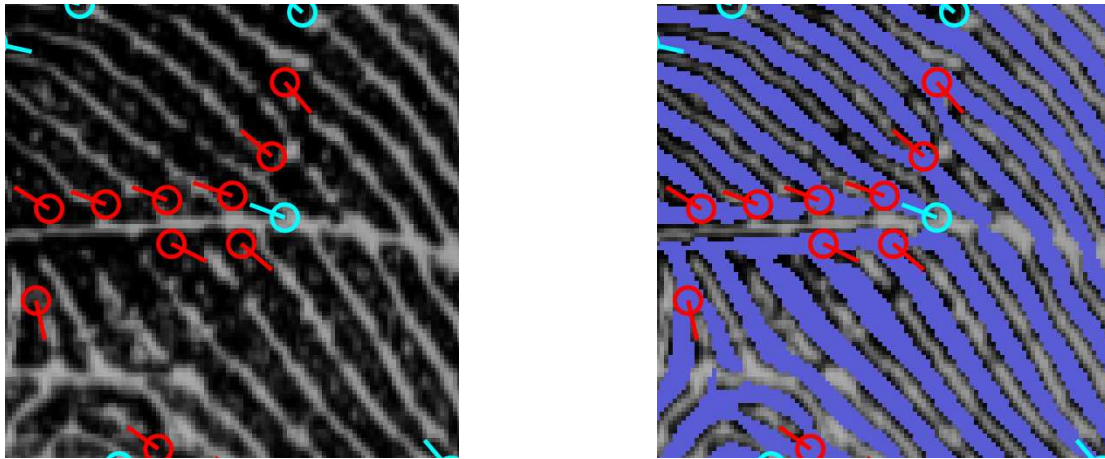
6.2 Le prove effettuate

Tutte le sperimentazioni effettuate in questa terza fase prendono il via dalle statistiche ricavate al punto precedente. Una maggiore priorità si è data all'analisi di situazioni influenzate dalla presenza di minuzie *false*: statisticamente, infatti, il numero di quest'ultime supera quello delle *missed*. Si è proceduto in questo modo: si sono calcolate le prestazioni di BioLab in situazione di utilizzo di parametri *standard* e, basandosi sull'analisi di qualche campione significativo, si è cercato di dedurre quali parametri potessero essere modificati per ottenere migliorie.

In un primo momento gli sforzi si sono concentrati soprattutto sulla problematica di Bad Quality. Come anticipato, FLab mette a disposizione strumenti in grado di analizzare le varie informazioni sul campione. Fra queste appare appunto il *tool* per la segnalazione di aree di scarsa qualità. Come si evince dalle percentuali ricavate nel paragrafo precedente, questa problematica affligge trasversalmente tutte le classi di NFIQ e, seppur in misura diversa, comporta la generazione di numerose *false*. I *test* effettuati si sono mossi in direzione di un abbassamento della soglia minima di qualità per la rimozione di una minuzia e sulla ricerca di combinazioni che potessero in qualche modo ridurre il numero di *false* identificate in queste zone. Purtroppo questi primi tentativi si sono rivelati vani e i comportamenti osservati sono stati di duplice natura:

- molto spesso, a fronte dell'individuazione di un ingente numero di *false* in aree contrassegnate come compromesse, vengono trovate altrettante minuzie corrette. Da qui un'inapplicabilità dell'indice di qualità sotto questo punto di vista;
- vi sono situazioni in cui la valutazione di qualità dell'algoritmo non corrisponde a quella riscontrabile ad occhio nudo. È questo il caso di immagini aventi aree molto compromesse che non vengono però considerate.

Avendo dunque constatato che gli spazi di miglioramento legati a questa problematica erano piuttosto ridotti e che nel complesso questo indice di qualità non fosse affidabile a livello locale per effettuare una scrematura deterministica delle minuzie, si è passati a prove successive. L'attenzione si è focalizzata in particolare su problematiche di Big Cut e Wrong Bridge: vengono qui citate insieme in quanto già in fase di analisi si era notato un forte legame tra le due. In particolare, a fronte di tagli e lacerazioni, non sempre l'algoritmo riesce a ricongiungere segmenti appartenenti alla stessa *ridge*. In realtà, però, l'alta percentuale di *false* non è in questo caso causata dal mancato riallacciamento. Viene mostrato di seguito un esempio emblematico [Figura 6.16] per meglio intendere ciò di cui si sta parlando.

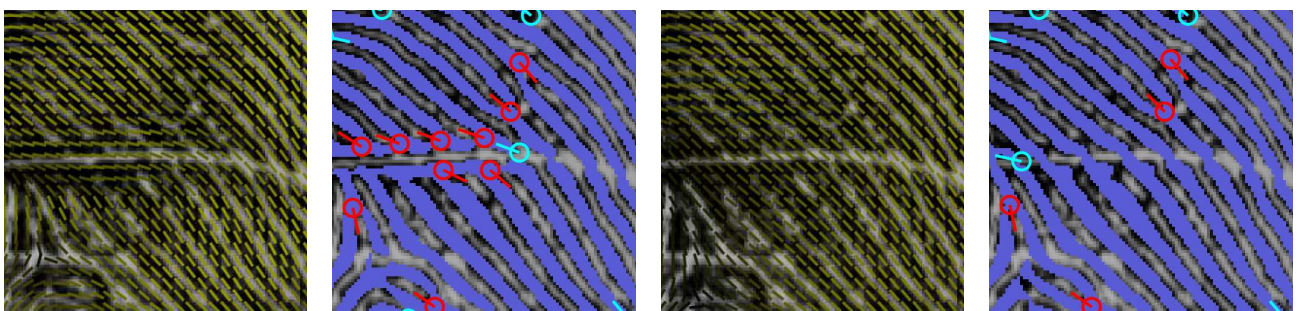


(a) Immagine originale

(b) Immagine *enhanced*

Figura 6.16: Esempio di taglio con conseguente formazione di false (segnalate in rosso)

Come si può notare, il problema è qui originato da tutte le false biforcazioni che si vengono a creare dal congiungimento errato di più *ridge* che si affacciano sullo stesso taglio: in questo senso si origina quella che potrebbe essere definita una cresta parallela all'andamento della lacerazione stessa. Di fronte al fenomeno, ricordando i cenni di teoria esposti al Capitolo 2, si è intuito che il problema potesse essere in questo caso collegato all'immagine delle orientazioni. La fase di *enhancement* viene infatti operata anche sulla base delle indicazioni fornite dall'immagine direzionale. Le immagini di cui sotto [Figura 6.17] riportano lo stesso caso analizzato in Figura 6.16 in cui si apprezzano i cambiamenti intercorsi a seguito di manipolazioni effettuate sui parametri relativi alle orientazioni.



(a) Immagine direzionale prima

(b) Immagine *enhanced* prima

(c) Immagine direzionale dopo

(d) Immagine *enhanced* dopo

Figura 6.17: Cambiamenti ravvisati a seguito delle modifiche apportate sull'immagine delle orientazioni

In zone compromesse in cui l’algoritmo non riesce ad intuire l’andamento delle orientazioni locali, le decisioni vengono prese basandosi sull’analisi dei valori adiacenti. Può però succedere, come in questo caso, che considerando aree di campionamento troppo piccole ne conseguano valutazioni errate. Abbiamo cercato quindi di trovare una soglia più alta per il parametro `OrientationWindowSize` per ottenere miglioramenti, come nel caso in Figura 6.17, stando attenti a non creare però artefatti in altre zone dell’immagine. I parametri considerati sono stati quelli compresi tra 30 e 60 ad intervalli di 5. I risultati mostrati nel paragrafo successivo hanno delineato miglioramenti consistenti in rapporto a `BstFVCEXtr`, mentre una situazione stagnante in relazione a `VeriFinger 10.0` (si confrontino con quelli esposti nella Sezione 3.2.2).

6.3 I risultati ottenuti

Vengono qui considerati come risultati ottimali di confronto quelli relativi alle prestazioni del sistema a seguito dell’utilizzo di parametri di *default*.

Ogni miglioramento successivo è valutato in termini di abbassamento delle percentuali delle *false*, tollerando in questo caso l’introduzione di qualche *missed* in più. Omettiamo i risultati conseguiti con parametri legati a `Bad Quality` in quanto sarebbero fondamentalmente irrilevanti ai fini di un’analisi conclusiva. Vengono di seguito riportati gli esiti ottenuti a seguito dei *test* sulle varie combinazioni di `OrientationWindowSize` (omettiamo di riportare quelli conseguiti in rapporto a `VeriFinger 10.0` in quanto non presentano miglioramenti rispetto a quelli mostrati in Sezione 3.2.2).

Risultati in relazione a `BstFVCEXtr`

NFIQ	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
NFIQ3	0.00 %	18.68 %	37.69 %	18.45 %	38.59 %
NFIQ4	0.00 %	26.49 %	100.54 %	26.40 %	107.88 %
NFIQ5	0.00 %	27.28 %	154.27 %	27.71 %	172.61 %

(a) `OrientationWindowSize = 30`

NFIQ	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
NFIQ3	0.00 %	21.10 %	31.96 %	20.77 %	32.18 %
NFIQ4	0.00 %	29.94 %	73.90 %	29.20 %	79.36 %
NFIQ5	0.00 %	34.76 %	107.76 %	33.06 %	123.40 %

(b) OrientationWindowSize = 35

NFIQ	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
NFIQ3	0.00 %	20.60 %	32.75 %	20.26 %	33.16 %
NFIQ4	0.00 %	29.38 %	78.17 %	28.66 %	84.59 %
NFIQ5	0.00 %	33.60 %	115.23 %	32.03 %	133.68 %

(c) OrientationWindowSize = 40

NFIQ	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
NFIQ3	0.00 %	20.72 %	32.24 %	20.38 %	32.65 %
NFIQ4	0.00 %	29.45 %	76.43 %	28.72 %	82.47 %
NFIQ5	0.00 %	33.88 %	112.90 %	32.25 %	130.63 %

(d) OrientationWindowSize = 45

NFIQ	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
NFIQ3	0.00 %	20.81 %	32.04 %	20.47 %	32.44 %
NFIQ4	0.00 %	29.48 %	75.33 %	28.76 %	81.16 %
NFIQ5	0.00 %	34.04 %	111.66 %	32.40 %	129.13 %

(e) OrientationWindowSize = 50

NFIQ	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
NFIQ3	0.00 %	20.89 %	31.96 %	20.55 %	32.35 %
NFIQ4	0.00 %	29.54 %	74.56 %	28.82 %	80.25 %
NFIQ5	0.00 %	34.15 %	111.07 %	32.49 %	128.43 %

(f) OrientationWindowSize = 55

NFIQ	NotEnrolled	Missed(All)	False(All)	Missed(Tmp)	False(Tmp)
NFIQ3	0.00 %	20.97 %	31.95 %	20.63 %	32.35 %
NFIQ4	0.00 %	29.57 %	74.01 %	28.87 %	79.58 %
NFIQ5	0.00 %	34.18 %	110.74 %	32.52 %	128.04 %

(g) OrientationWindowSize = 60

Figura 6.18: Prestazioni raggiunte con i vari valori di OrientationWindowSize

7 Conclusioni

Al termine di questo percorso traiamo qui le conclusioni fornendo qualche spunto per successive prossime ricerche. Potremmo operare una prima distinzione sulla base degli esiti conseguiti: se da una parte ci sono sicuramente i risultati concreti, strettamente legati agli algoritmi e misurabili in termini di prestazioni dei sistemi, dall'altra, a fronte anche di qualche fallimento, ci sono riflessioni valide anche sul piano conoscitivo. In generale, partendo dalle considerazioni su MQF, si può certamente affermare che l'applicazione di questa metrica fornisce un valido strumento di giudizio nella stima della qualità dell'immagine a livello globale. Dai *test* si evince addirittura che in corrispondenza di immagini di buona qualità e di particolari estrattori i risultati siano migliori di quelli conseguiti con NFIQ ed NFIQ 2.0. Al contrario, per quanto concerne invece l'idea di estrapolare le strategie adottate dall'algoritmo ed applicarle anche a livello locale, i risultati non hanno condotto agli esiti sperati. Triangolazione di Delaunay e analisi di perimetro ed area a livello di singola minuzia si sono dimostrati inefficaci per calcolare un possibile valore di *score*, anche qualora combinati con il già utilizzato indice di coerenza. Soffermandoci invece sull'algoritmo ibrido, risultati concreti sul piano meramente prestazionale non si sono ravvisati. Le procedure proposte sono sicuramente valide da un punto di vista teorico, ma nel complesso troppo deterministiche. Esse sono interamente orientate ad un miglioramento della qualità dei *templates* - nel caso specifico tutte le operazioni vengono effettuate sull'immagine *enhanced* - senza considerare il grado di influenza di tali situazioni d'errore sulle prestazioni del sistema. Per fare un po' di chiarezza, si è osservato che molti estrattori tendono a commettere errori ricorrenti, ma questi non incidono poi così drasticamente in termini di prestazioni negative. In pratica ciò che rende più robusto un estrattore rispetto ad un altro è il fatto che determinate minuzie vengano individuate sempre, a prescindere che poi queste siano false o vere a livello formale. Questa è stata forse una delle conclusioni più importanti in quanto ha permesso di conferire una direzione alle prove successive, aprendo dunque nuovi scenari di indagine. Inoltre questa osservazione ha fornito una giustificazione ed una spiegazione ad esiti che nel complesso non hanno soddisfatto le aspettative maturate.

In un'ultima fase, infine, il lavoro è stato più improntato ad un'analisi strutturale sulle caratteristiche dell'estrattore del BioLab. L'operazione di classificazione apre sicuramente le porte a possibili nuovi *test* che possano spingersi molto più a fondo di quanto non si sia fatto in questa trattazione. Da un punto di vista più pragmatico, gli esperimenti condotti sull'immagine delle orientazioni riportano miglioramenti in termini di prestazioni se paragonate a quelle iniziali e se preso come estrattore di riferimento BstFVCEXtr. L'intero processo di estrazione è fortemente influenzato dalla qualità dell'immagine direzionale e le migliorie apportate su quest'ultima si concretizzano dunque in una immagine *enhanced* di maggiore qualità e meno sensibile alle problematiche esaminate nei primi capitoli.

In ultima analisi, mentre le considerazioni effettuate fino ad ora entrano nel merito del lavoro svolto, non si può trascurare il contributo che un simile lavoro di ricerca ha avuto su un piano prettamente personale. Il percorso intrapreso ha infatti condotto, oltre che ad un indiscusso arricchimento sul fronte delle conoscenze, ad un'acquisizione di strumenti validi anche dal punto di vista metodologico: le strategie messe in atto e le scelte via via adottate sono state, anche se ad un livello tendenzialmente didattico, quelle impiegate in un qualsiasi percorso di ricerca. Questo lavoro di tesi vuole dunque rappresentare un trampolino di lancio per future sperimentazioni e in questo senso i procedimenti e gli esiti qui documentati possono indicare una direzione percorribile nell'ambito delle tecniche di estrazione delle minuzie.

Bibliografia

- [1] Yao Z. , Le Bars J. , Charrier C. & Rosenberger C. (2015). Quality Assessment of Fingerprints with Minutiae Delaunay Triangulation. ICISSP 2015 - 1st International Conference on Information Systems Security and Privacy, Proceedings.
- [2] Jabeen S. & Khan S. (2008). A hybrid false minutiae removal algorithm with boundary elimination. 10.1109/SYSOSE.2008.4724177.
- [3] Klatzow D. Justice Denied: The Role of Forensic Science in the Miscarriage of Justice. Penguin Random House South Africa; 2014.
- [4] Ashbaugh D.R. Quantitative-Qualitative Friction Ridge Analysis: An Introduction to Basic and Advanced Ridgeology. CRC Press; 1999.
- [5] Champod C., Lennard C. J., Margot P., Stoilovic M. Fingerprints and Other Ridge Skin Impressions. CRC Press; 2017.
- [6] https://www.fbi.gov/file-repository/about-us-cjis-fingerprints_biometrics-biometric-center-of-excellences-fingerprint-recognition.pdf/view
- [7] Birkhoff J.M. Nozioni di medicina legale. Uno strumento per le professioni medico-sanitarie e giuridiche. Franco Angeli Edizioni; 2011.
- [8] Maltoni D. , Jain A.K. , Maio D. & Prabhakar S. Handbook of Fingerprint Recognition. Springer; 2003.
- [9] Kass M. & Witkin A. Analyzing oriented patterns. Comput. Vision Graph. Image Process. , 37(3):362-385, 1987.
- [10] Jain A. K., Hong L. & Bolle R., "On-line fingerprint verification", IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(4), 1997, pp. 302–314.
- [11] <http://bias.csr.unibo.it/fvc2004/>
- [12] Cappelli R. (2004). SFinGe: an Approach to Synthetic Fingerprint Generation. International Workshop on Biometric Technologies.
- [13] <https://github.com/FingerJetFXOSE/FingerJetFXOSE>
- [14] <https://www.nist.gov/services-resources/software/nist-biometric-image-software-nbis>
- [15] <http://bias.csr.unibo.it/fvc2000/>
- [16] <http://bias.csr.unibo.it/fvc2002/>

- [17] <http://bias.csr.unibo.it/fvc2006/>
- [18] Cappelli R. , Ferrara M. & Maltoni D. , "Minutia Cylinder-Code: A New Representation and Matching Technique for Fingerprint Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 12, pp. 2128-2141, Dec. 2010.