

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA

*DIPARTIMENTO DI INGEGNERIA CIVILE, CHIMICA, AMBIENTALE E DEI
MATERIALI*

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA CIVILE

TESI DI LAUREA

in

LABORATORIO DI INFRASTRUTTURE VIARIE E TRASPORTI

**METODI DI MAP-MATCHING PER L'IDENTIFICAZIONE
DEGLI ITINERARI DEI CICLISTI**

CANDIDATO:

Giovanni Birgillito

RELATORE:

Chiar.mo Prof. Ing. Joerg Schweizer

Anno Accademico 2014 - 2015

Sessione III

INDICE

INTRODUZIONE	4
CAPITOLO I: IL SISTEMA DI POSIZIONAMENTO NAVSTAR GPS	7
1. I SISTEMI DI POSIZIONAMENTO GNSS	7
2. IL SISTEMA GPS	8
3. I SEGMENTI DEL SISTEMA GPS	9
3.1 IL SEGMENTO SPAZIALE	10
3.2 IL SEGMENTO DI CONTROLLO	12
3.3 IL SEGMENTO UTENTE	14
4. ERRORI NELLE MISURAZIONI	17
5. APPLICAZIONI E DISPOSITIVI MOBILI	19
6. SISTEMI DI POSIZIONAMENTO ALTERNATIVI	20
7. RIFERIMENTI CARTOGRAFICI	21
8. MAPPE DIGITALIZZATE E AMBIENTE GIS	24
CAPITOLO II: IL MAP-MATCHING	26
1. IL PROCEDIMENTO DI MAP-MATCHING	26
2. ORIGINI DEI DATI	29
3. SCOPI E APPLICAZIONI	31
4. PROBLEMATICHE PIÚ COMUNI	33
5. CLASSIFICAZIONE DEGLI ALGORITMI DI MAP-MATCHING	38
CAPITOLO III: STUDIO DEI METODI DI MAP-MATCHING	44
1. DEFINIZIONE DELLO STUDIO, PIATTAFORME UTILIZZATE E RACCOLTA DATI DI INPUT	44
2. PREPROCESSING E DATI DI OUTPUT	50
3. PRIMA METODOLOGIA: ALGORITMO ‘orig’	54

3.1	INDICI DI VALUTAZIONE	56
3.3	RISULTATI E CONSIDERAZIONI	60
4.	PRIMA METODOLOGIA: ALGORITMO PROPOSTO ‘new’	66
4.1	ANALISI DEI PARAMETRI	67
4.2	RISULTATI E CONSIDERAZIONI	76
5.	CONFRONTO TRA GLI ALGORITMI ‘orig’ E ‘new’	80
6.	SECONDA METODOLOGIA: ALGORITMO ‘Marchal’	86
6.1	ANALISI DEI PARAMETRI	90
6.2	RISULTATI E CONSIDERAZIONI	96
7.	SECONDA METODOLOGIA: ALGORITMO PROPOSTO ‘NewMarchal’ ..	101
7.1	ANALISI DEI PARAMETRI	112
7.2	RISULTATI E CONSIDERAZIONI	118
8.	CONFRONTO TRA GLI ALGORITMI ‘Marchal’ E ‘NewMarchal’	120
9.	CONFRONTO GENERALE TRA GLI ALGORITMI	127
10.	RISULTATI FINALI.....	138
CONCLUSIONI.....		142
BIBLIOGRAFIA.....		147
APPENDICE		150

INTRODUZIONE

Negli ultimi anni si è assistito al considerevole aumento della disponibilità di dati GPS e della loro qualità ed accuratezza, dovuto alla diffusione e all'evoluzione tecnologica di smartphone e di applicazioni di localizzazione e navigazione per dispositivi mobili.¹ L'enorme quantità di dati a disposizione degli analisti ha quindi condotto alla necessità di elaborare degli algoritmi di map-matching, con lo scopo di individuare i percorsi scelti dagli utenti sulla rete.

Il processo di map-matching consiste nell'integrare proprio tali dati, provenienti da un sistema di posizionamento, - solitamente una lista ordinata di punti, identificati tramite coordinate geografiche ricavate mediante un sistema di localizzazione GNSS (*Global Navigation Satellite System*), tipo GPS - con dati di mappe digitali, al fine di identificare l'arco su cui un veicolo sta viaggiando e determinarne l'esatta posizione sul medesimo.²

Obiettivo dell'ingegneria dei trasporti è identificare o ricostruire il percorso scelto dall'utente per lo spostamento; pertanto, nell'ambito d'interesse di tale disciplina, lo scopo del map-matching si esprime nella ricerca del percorso che rappresenti la migliore stima dell'itinerario realmente scelto dall'utente.³

Una serie di algoritmi, basati su metodi matematici, sono stati sviluppati in letteratura da studiosi e ricercatori al fine di eseguire tale operazione.

Il problema è in genere affrontato attraverso appositi software su piattaforma GIS (*Geographic Information System*), un sistema informativo computerizzato che permette l'acquisizione, l'elaborazione, l'analisi, la gestione e la rappresentazione grafica di dati di tipo georeferenziato, che sono dati (*attributi*) associati ad elementi geografici, mediante l'utilizzo di mappe digitali.

¹ Schweizer J., Rupi F., 2014.

² Quddus M., Washington S., 2015.

³ Marchal F. et al., 2004.

Il map-matching si colloca come utile strumento all'interno delle funzioni di monitoraggio e di gestione dei sistemi di trasporto: i dati ottenuti, infatti, possono essere usati per meglio comprendere il comportamento degli utenti sulla rete e le conseguenze di tale comportamento sia sul sistema di trasporto, sia sulle esternalità come l'inquinamento atmosferico.⁴ Inoltre, unitamente alle tecniche informatiche di modellazione e di simulazione del traffico ed alle reti di comunicazione, gli algoritmi di map-matching possono consentire un controllo più efficace e migliorare le prestazioni nell'ambito dei Sistemi di Trasporto Intelligenti (ITS, *Intelligent Transportation Systems*).⁵

Invero, una migliore comprensione della condotta dell'utente all'interno della rete di trasporto, in merito alle scelte modali o del percorso, è un passaggio essenziale che consente di migliorare la pianificazione dei sistemi di trasporto, ed in particolare di adottare provvedimenti che promuovano i modi di trasporto sostenibili, come la mobilità ciclabile.⁶ Il compito del gestore del sistema di trasporto, sia esso pubblico o privato, è infatti di fondamentale importanza, non soltanto al fine di risolvere i problemi strutturali dei trasporti, come la congestione, ma anche per migliorare il benessere collettivo, riducendo drasticamente le esternalità negative derivanti dal sistema di trasporto e favorendo lo sviluppo di una mobilità più sostenibile dal punto di vista ambientale, sociale ed economico.

Il presente lavoro si propone l'obiettivo di studiare alcune metodologie di map-matching per l'identificazione degli itinerari degli utenti, in particolare della mobilità ciclabile. Il lavoro si sviluppa in tre sezioni, che analizzano rispettivamente il sistema GPS, il map-matching e lo studio di alcune metodologie di map-matching, confrontandone i risultati e proponendo alcune soluzioni.

Nel primo capitolo della tesi è esposto il funzionamento dei sistemi di posizionamento ed in particolare del sistema GPS: ne sono discusse le caratteristiche, la suddivisione nei vari segmenti e la cartografia di riferimento, soffermandosi sulla versione

⁴ Quddus M. et al., 2003.

⁵ Quddus M. et al., 2007.

⁶ Menghini G. et al., 2009.

digitalizzata di uso comune in ambiente GIS, e sono analizzati gli errori di misurazione.

Nel secondo capitolo sono presentati i vari aspetti del procedimento di map-matching, dalla raccolta dei dati alle problematiche più comuni da affrontare, e le sue principali applicazioni; sono infine mostrate alcune possibili classificazioni degli algoritmi di map-matching sviluppati in letteratura.

Nel terzo capitolo è esposto lo studio eseguito su diversi algoritmi di map-matching, che sono stati testati su un database di spostamenti di ciclisti nell'area urbana di Bologna, registrati attraverso i loro smartphone sotto forma di punti GPS, e sulla relativa rete. Si analizzano altresì i risultati ottenuti in un secondo ambiente di testing, che è stato predisposto nell'area urbana di Catania, dove sono state registrate in modo analogo alcune tracce di prova, utilizzando la corrispondente rete cittadina. La comparazione degli algoritmi è eseguita graficamente e attraverso degli indicatori. Vengono inoltre proposti e valutati due algoritmi che forniscono un aggiornamento di quelli analizzati, al fine di migliorarne le prestazioni in termini di accuratezza dei risultati e di costo computazionale.

CAPITOLO I: IL SISTEMA DI POSIZIONAMENTO NAVSTAR GPS

1. I SISTEMI DI POSIZIONAMENTO GNSS

Con la definizione di Sistema Satellitare Globale di Navigazione (GNSS: *Global Navigation Satellite System*) si indica un sistema di geo-radiolocalizzazione e navigazione terrestre, marittima o aerea, che utilizza una rete di satelliti artificiali in orbita.

I sistemi sono dedicati alla fornitura di un servizio di posizionamento geo-spaziale a copertura globale che permette a piccoli appositi ricevitori elettronici di determinare le loro coordinate geografiche (longitudine, latitudine ed altitudine) su un qualunque punto della superficie terrestre, elaborando i segnali radio trasmessi dai satelliti della costellazione.

Tra i sistemi GNSS si includono il sistema statunitense *NAVigation Satellite Timing and Ranging Global Positioning System* (NAVSTAR GPS, noto come GPS, il sistema GNSS più conosciuto), il sistema russo *GLONASS (GLobal Orbiting NAVigation Satellite System)*, il sistema europeo Galileo, in fase di completamento, il sistema cinese COMPASS, evoluzione del sistema regionale Beidou, ed il sistema regionale indiano *IRNSS (Indian Regional Navigational Satellite System)*, in fase di sviluppo.⁷

⁷ Rao G.S., 2010.

2. IL SISTEMA GPS

Il GPS (*Global Positioning System*) è un sistema di posizionamento globale e di navigazione satellitare, basato sulla ricezione di segnali radio, emessi da una costellazione di satelliti artificiali in orbita attorno alla Terra, da parte di appositi ricevitori. Il sistema è progettato in modo da permettere il posizionamento tridimensionale di oggetti, anche in movimento, in ogni istante ed in ogni luogo del pianeta.⁸ Attraverso la rete di satelliti artificiali, esso fornisce ad un ricevitore GPS sulla Terra informazioni sulle sue coordinate geografiche, nel caso lo strumento riesca ad agganciare il segnale di almeno quattro satelliti del sistema. La localizzazione avviene per mezzo della trasmissione di un segnale radio, inviato da ciascun satellite, contenente informazioni sul timing e sulla posizione, e mediante l'elaborazione dei segnali ricevuti, effettuata dal ricevitore.

Il sistema GPS è gestito dal governo degli Stati Uniti d'America ed è attualmente liberamente accessibile da chiunque sia dotato di un ricevitore GPS sulla Terra. Il suo grado di accuratezza della posizione planimetrica è dell'ordine della decina di metri (possibile errore di 9 metri in un intervallo di confidenza del 95%, in condizioni standard), mentre gli errori sono leggermente maggiori sull'altitudine (15 metri di errore al 95° percentile, in condizioni standard).⁹ Le misure sono affette da errori dipendenti dalle condizioni meteorologiche, dalla disponibilità e dalla posizione dei satelliti rispetto al ricevitore, dalla qualità e dal tipo di ricevitore, dagli effetti di riflessione e di rifrazione del segnale radio negli strati dell'atmosfera terrestre e dagli effetti della relatività della variabile tempo.

Il sistema GPS fu progettato tra gli anni sessanta e settanta in piena Guerra Fredda dal Dipartimento della Difesa Statunitense (USDoD: *United States Department of Defense*) per scopi militari. Esso venne reso fruibile ai civili nel 1991, tramite il servizio SPS (*Standard Positioning System*), seppur con forti limitazioni e minore accuratezza e riservatezza rispetto agli standard militari (servizio PPS, *Precision Positioning System*, il cui segnale è tuttora criptato): fino al 2000 la precisione del GPS

⁸ Cina A., 2004.

⁹ USA Government – Department of Defense, 2008.

per usi civili era perfino intenzionalmente ridotta per decisione del governo statunitense (*Selective Availability*), mediante la manipolazione degli orologi e delle effemeridi.¹⁰ Il gap tra i due sistemi è stato nel tempo diminuito e la precisione del sistema disponibile per uso civile notevolmente migliorata, nonostante permangano delle limitazioni, dovute al fatto che esso resta subordinato al controllo del Governo Statunitense.

3. I SEGMENTI DEL SISTEMA GPS

Il *Global Positioning System* è composto da tre sezioni (Fig. 1.1): il segmento spaziale (*space segment*), il segmento di controllo (*control segment*) ed il segmento utente (*user segment*). L'Aeronautica Militare degli Stati Uniti (USAF, *United States Air Force*) sviluppa, gestisce e monitora il segmento spaziale ed il segmento di controllo.

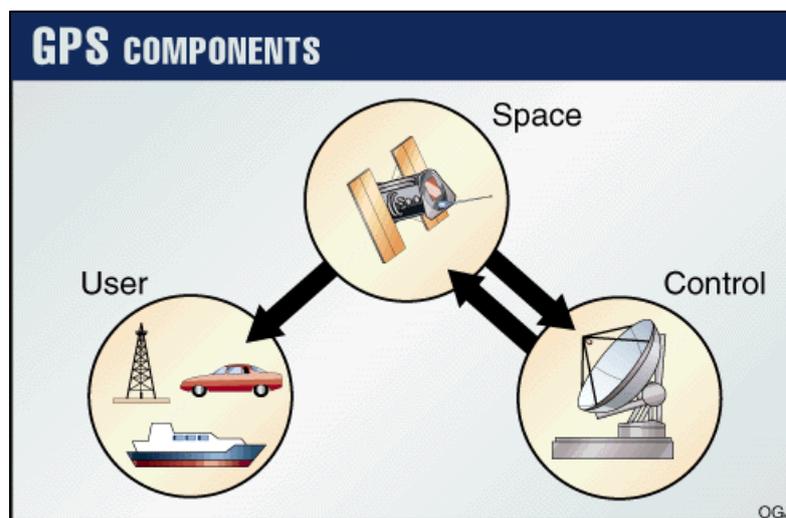


Figura 1.1: i segmenti del sistema GPS (<http://www.ogj.com>)¹¹

¹⁰ Cina A., 2004.

¹¹ Oil and Gas Journal, <http://www.ogj.com>

3.1 IL SEGMENTO SPAZIALE

Il segmento spaziale dal 2010 è costituito da una costellazione di 31 satelliti disposti su sei diversi piani orbitali con inclinazione di 55° sul piano equatoriale, ad una altitudine di circa 20.000 km (Fig. 1.3); in ogni caso gli USA si sono impegnati a mantenere la disponibilità di almeno 24 satelliti.¹² Essi descrivono un'orbita pressoché circolare, con eccentricità massima tollerata di 0,03, raggio di circa 26.560 km e periodo pari a circa 12 h, circa metà giorno siderale: non sono infatti geostazionari.¹³ Ciascun piano orbitale ha almeno quattro satelliti, ed i piani sono disposti in modo tale che ogni utilizzatore su un qualsiasi punto della Terra, in un qualunque momento, possa ricevere il segnale di almeno cinque satelliti. Ogni satellite possiede un certo numero di orologi atomici (al cesio o al rubidio), al fine di ridurre gli errori nel timing, ed è provvisto di pannelli solari fotovoltaici per la produzione di energia elettrica, abbinati ad una batteria, per garantire l'energia necessaria per le operazioni anche quando il sole è eclissato da altri corpi celesti. Le funzioni principali dei satelliti sono quelle di trasmettere informazioni agli utilizzatori tramite un segnale radio, mantenere un riferimento di tempo accurato grazie agli orologi di bordo, ricevere e memorizzare informazioni dal segmento di controllo ed eseguire manovre e correzioni di orbita.¹⁴

Ciascun satellite emette informazioni su due canali: L1, l'unico disponibile al servizio SPS (per uso civile), e L2, criptato, per l'uso esclusivo per il servizio PPS (ad uso militare), a frequenze diverse: rispettivamente, 1575,42 MHz (lunghezza d'onda λ corrispondente pari a circa 19 cm) e di 1227,6 MHz (lunghezza d'onda λ corrispondente pari a circa 24 cm). Il principale scopo della doppia frequenza è quello di eliminare diverse cause d'errore, ad esempio l'effetto del ritardo del segnale dovuto alla rifrazione ionosferica.¹⁵

Vi sono dei codici che modulano le frequenze di segnale, ovvero le criptano (Fig. 1.2): essi sono costituiti da onde quadre formate da transizioni di valori $+1/-1$, generate da

¹² USA Government, <http://www.gps.gov/systems/gps/space>

¹³ Cina A., 2004.

¹⁴ Cina A., 2004.

¹⁵ Cina A., 2004.

un algoritmo che si ripete periodicamente nel tempo.¹⁶ Il codice C/A (*Coarse Acquisition*), di frequenza pari a 1,023 MHz e lunghezza d'onda λ pari a circa 300 metri, è usato per modulare la portante L1, mentre il codice P (*Precision*), di frequenza pari a 10,23 MHz e lunghezza d'onda λ pari a circa 30 metri, riservato ad uso militare, modula entrambe le portanti L1 e L2.¹⁷ Le onde così sfasate devono essere demodulate, al fine di depurarle dai codici e poterle leggere; tale operazione è possibile per utenti civili solo per la portante L1.

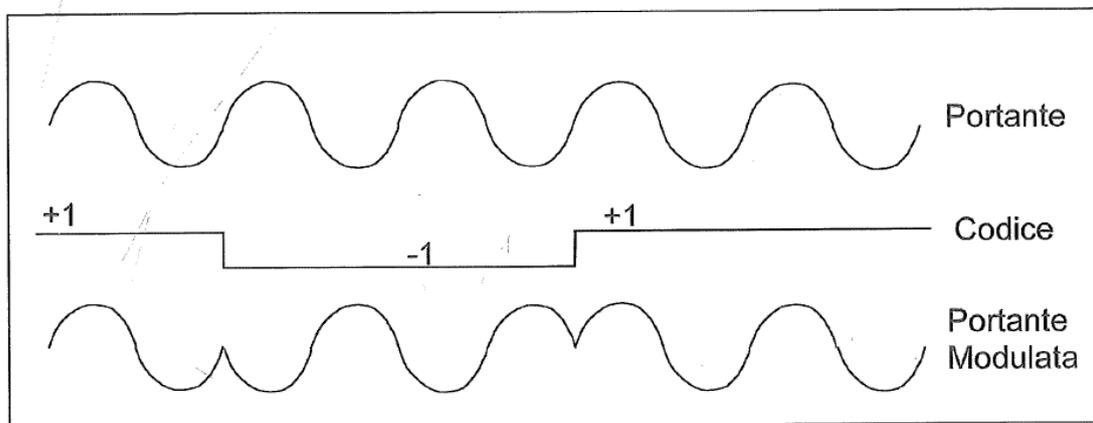


Figura 1.2: modulazione della portante (Cina A., 2004)¹⁸

Il segnale emesso contiene come informazioni l'istante di tempo della trasmissione del satellite (*satellite time-of-transmission*), le effemeridi del satellite (*satellite ephemeris*) - ovvero alcune informazioni di tipo astronomico come posizionamento o parametri orbitali -, il grado di funzionalità del satellite (*satellite health*), una correzione relativistica dell'orologio satellitare (*satellite clock correction*), alcune informazioni sul ritardo del segnale dovuti alla ionosfera (*ionospheric delay effects*), una

¹⁶ Cina A., 2004.

¹⁷ Cina A., 2004.

¹⁸ Cina A., 2004.

correlazione con il tempo coordinato universale (UTC), e lo stato della costellazione (*constellation status*), ovvero dell'insieme dei satelliti.¹⁹

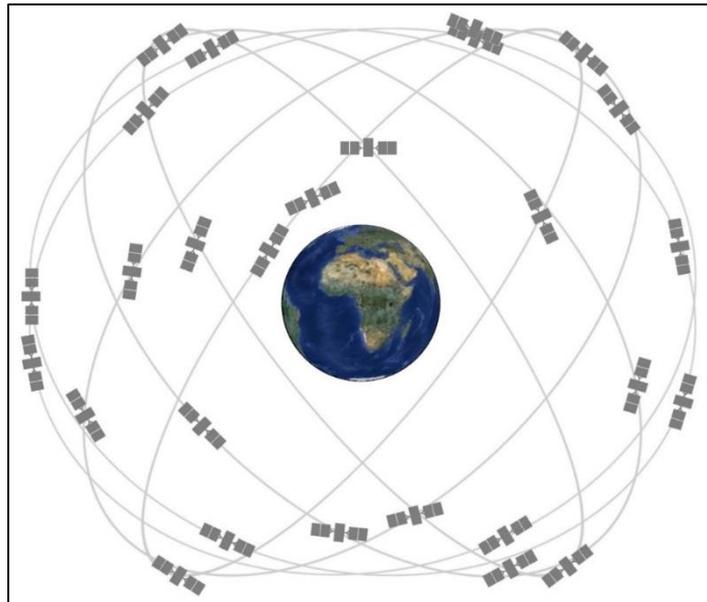


Figura 1.3: la costellazione dei satelliti del sistema GPS (<http://www.gps.gov>)²⁰

3.2 IL SEGMENTO DI CONTROLLO

Il segmento di controllo è composto da una stazione di controllo principale (MCS: *Master Control Station*), una stazione di controllo principale alternativa (*Alternative Master Control Station*), antenne terrestri (*Ground Antennas*) e stazioni di controllo dedicate (Fig. 1.4). Le sue funzioni sono di tracciare il movimento dei satelliti, monitorare ed analizzare le loro trasmissioni ed inviare loro dati e comandi.

Le traiettorie dei satelliti vengono rilevate da apposite stazioni dell'Aeronautica Militare Statunitense dislocate in diverse zone del pianeta; i dati vengono inviati alla

¹⁹ Rao G.S., 2010.

²⁰ USA Government, <http://www.gps.gov>

stazione di controllo principale, nei pressi di Colorado Springs (CO, USA), ed in base ad essi il Comando comunica ad intervalli regolari ad ogni satellite le necessarie correzioni sul tempo e sull'orbita tramite le antenne. Lo scopo di tali aggiornamenti è di sincronizzare gli orologi atomici a bordo dei satelliti, in modo da avere una differenza di pochi nanosecondi tra di essi, e di aggiornare le effemeridi del modello orbitale interno. Durante tali manovre, che comprendono il cambiamento dell'orbita di un satellite, questo viene messo fuori servizio (*unhealthy*), diventando in tal modo inaccessibile per un ricevitore; una volta terminata l'operazione, l'orbita viene quindi controllata ed acquisita da Terra ed il satellite rimesso in servizio con le nuove effemeridi.

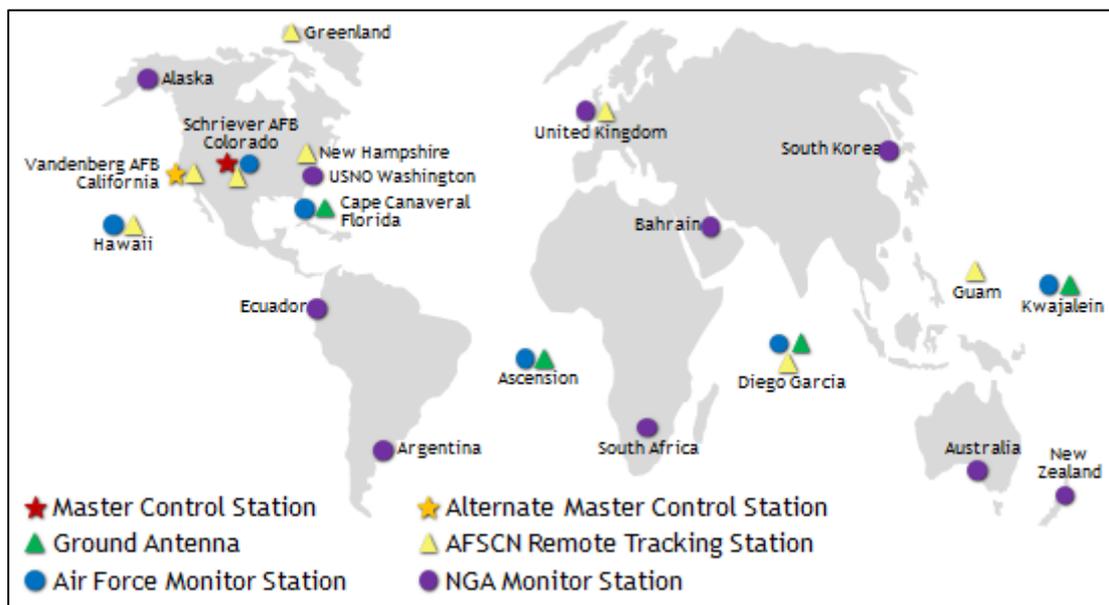


Figura 1.4: il segmento di controllo (<http://www.gps.gov>)²¹

²¹ USA Government, <http://www.gps.gov>

3.3 IL SEGMENTO UTENTE

Il segmento utente è costituito dai ricevitori GPS civili sulla Terra che si servono del sistema SPS e dai ricevitori GPS militari che invece sfruttano il PPS. Generalmente i ricevitori sono costituiti da un'antenna, un microprocessore che demodula il segnale ed un orologio.

Esistono in commercio ricevitori che consentono di captare ed elaborare il segnale GPS su dispositivi appositi o che costituiscono una funzione accessoria di un altro apparecchio. Per la navigazione esistono software appositi, anche open source, che utilizzano una cartografia anch'essa disponibile liberamente. Sono in continuo aumento i ricevitori GPS integrati all'interno di telefoni cellulari di nuova generazione (*smartphone*), tablet, PC o orologi, strumenti portatili ormai di uso comune anche per la navigazione satellitare.

I ricevitori hanno in genere necessità di agganciare il segnale di almeno tre satelliti per effettuare una prima misura di posizionamento. I satelliti supplementari migliorano la precisione del sistema permettendo misure sovrabbondanti e la correzione di eventuali errori, aumentando la precisione.

Il ricevitore GPS è innanzitutto capace di riconoscere il satellite attraverso il codice identificativo, ricercandolo nella banca dati di codici in possesso dello strumento. La principale funzione del ricevitore è quella di calcolare la distanza dal satellite tramite la misura del tempo impiegato a percorrere tale distanza dal segnale radio emesso dai satelliti stessi: il suddetto Δt è pari al rapporto tra la distanza tra i due elementi, supposti puntuali, e la velocità di propagazione delle onde elettromagnetiche nello spazio (Equazione 1.1). Ogni satellite infatti trasmette anche le effemeridi relative a se stesso, ed in tal modo il ricevitore GPS viene a conoscenza anche dei parametri dell'orbita da cui deriva la posizione del satellite, disponendo in tal modo di tutti gli elementi necessari a definire nello spazio la superficie di posizione, costituita da una sfera, luogo dei punti equidistanti dalla sorgente: il ricevitore effettua quindi una misura di *pseudorange*.

Il segnale inviato dal satellite contiene informazioni sull'istante di tempo di emissione del segnale stesso, grazie all'orologio atomico presente in ogni satellite: il ricevitore calcola la distanza di propagazione dal satellite a partire dalla differenza, in genere dell'ordine dei microsecondi, tra l'orario comunicato e quello del proprio orologio, sincronizzato con quello a bordo del satellite, tenendo conto della velocità di propagazione nello spazio del segnale. La misura avviene correlando i due segnali (quello ricevuto e quello proprio): infatti il ricevitore è in grado di replicare localmente all'istante di arrivo il segnale ricevuto in modo assolutamente identico.²² La misura dello sfasamento (*offset*) tra i due segnali, effettuata ricercandone il massimo di correlazione,²³ permette di valutare la distanza di pseudorange.

$$\rho_{psr,ij}(t) = c \cdot \Delta t \quad (1.1)$$

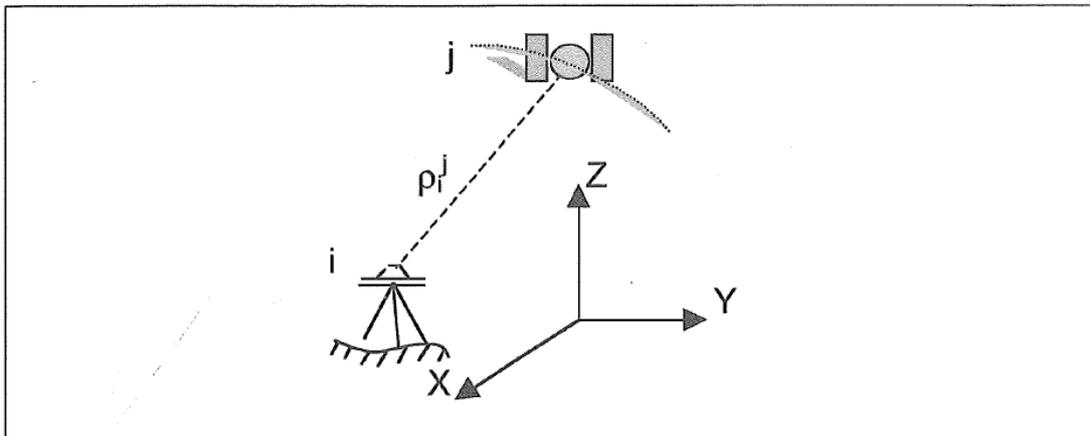


Figura 1.5: distanza geometrica (Cina A., 2004)²⁴

$$\rho_{ij}(t) = \sqrt{(x_j(t) - x_i)^2 + (y_j(t) - y_i)^2 + (z_j(t) - z_i)^2} \quad (1.2)$$

²² Cina A., 2004.

²³ Cina A., 2004.

²⁴ Cina A., 2004.

Il principio di funzionamento del meccanismo di localizzazione si basa sulla trilaterazione, una tecnica che valuta la posizione mediante la valutazione della distanza tra il ricevitore ed un certo numero di satelliti. Il sistema di riferimento è l'ellissoide geocentrico WGS84 (*World Geodetic System 1984*).

L'orologio in dotazione ai ricevitori GPS è molto meno preciso e sofisticato di quello a bordo dei satelliti, e per tale motivo è sottoposto frequentemente a correzioni: la sincronizzazione si avvia in genere all'accensione del dispositivo ricevente utilizzando le informazioni in arrivo dai satelliti. Se il ricevitore avesse anch'esso un orologio atomico perfettamente sincronizzato con quello dei satelliti, in assenza di altri errori, sarebbero sufficienti le informazioni fornite da tre satelliti per stabilirne l'esatta posizione, risolvendo un sistema non lineare polinomiale di tre equazioni in tre incognite (posizione spaziale alla generica epoca τ , generalmente latitudine, longitudine e altitudine), con tre dati a disposizione (distanza dai tre satelliti). Difatti, intersecando le tre sfere di raggio pari alle rispettive distanze misurate e di centro coincidente con le rispettive posizioni dei satelliti, è possibile individuare un unico punto significativo sulla superficie terrestre: l'intersezione di due sfere identifica una circonferenza, la quale a sua volta intersecata con la terza sfera determina nel caso generale due punti, di cui uno è però facilmente scartato in ragione della sua notevole distanza dalla superficie terrestre (Fig. 1.6).

Al fine di ovviare alla non elevata precisione delle informazioni temporali in suo possesso, il ricevitore impiega il segnale emesso da un quarto satellite, con funzione di correzione della variabile tempo: difatti, l'errore dell'orologio del ricevitore, variabile ad ogni epoca di misura, si considera come un'incognita.²⁵

²⁵ Cina A., 2004.

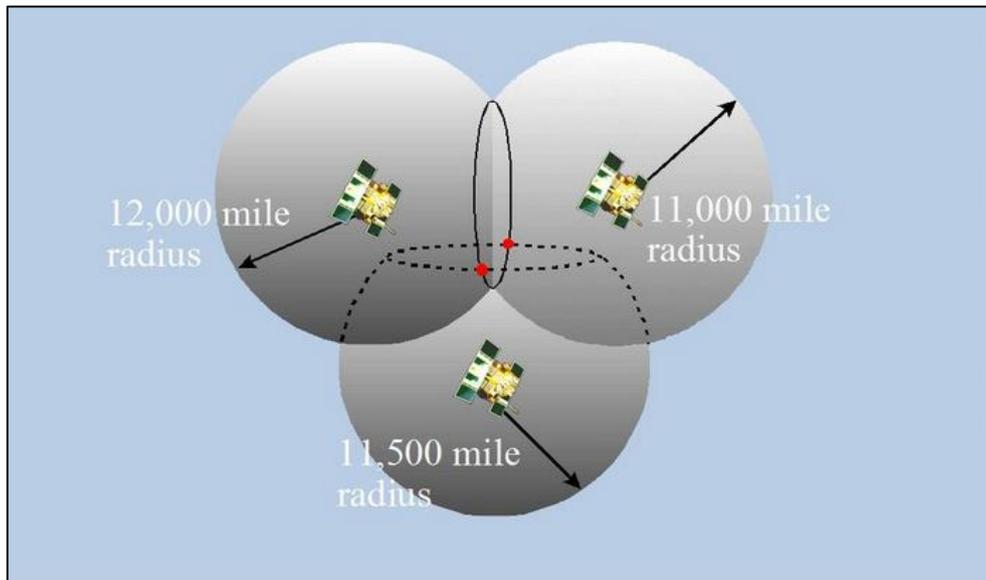


Figura 1.6: trilaterazione

4. ERRORI NELLE MISURAZIONI

Il sistema GPS è soggetto ad errori di vario tipo, sia nel segmento di terra che in quello spaziale. Essi possono essere fondamentalmente di tre tipologie: errori accidentali di misura, errori sistematici o di modello (*bias*) - ovvero errori di orologio, d'orbita o di rifrazione in atmosfera - ed errori di osservazione, come i *multipath errors*, interferenze elettromagnetiche o imprecisioni dovute alla parte elettronica del ricevitore.²⁶

Gli errori di tipo accidentale possono essere stimati empiricamente come circa l'1% della lunghezza d'onda: ciò porta a precisioni diverse a seconda del tipo di codice o di portante.²⁷

²⁶ Cina A., 2004.

²⁷ Cina A., 2004.

Gli errori correlati alla propagazione del segnale sono in prevalenza legati al ritardo di propagazione, che si genera negli strati atmosferici della troposfera (parte di spazio più interna dell'atmosfera, compresa tra la superficie della terra ed una altezza variabile da 8 a circa 20 km) e della ionosfera (parte di spazio più esterna dell'atmosfera, compresa tra 60 e 450 km di altitudine). Tali errori non sono eliminabili e possono influenzare notevolmente le prestazioni del sistema: sono indipendenti dalle frequenze e quindi di entità uguale per le portanti L1 e L2 nella troposfera, mentre dipendono dalle frequenze nella ionosfera.²⁸ Essi sono funzioni dei parametri atmosferici, quali temperatura e pressione, e della posizione del satellite, in particolare dell'angolo zenitale; essendo legati al tasso di elettroni liberi contenuti in atmosfera, hanno una relazione anche con l'attività solare.²⁹

Gli errori di timing sono in gran parte imputabili agli orologi in dotazione a bordo; nel corso degli anni si è assistito ad un notevole miglioramento tecnologico che ne ha diminuito l'entità. Tuttavia, generalmente, sono presenti più orologi, al fine di avere un timing più accurato e di limitare i danni in caso di malfunzionamento.

Gli errori sui parametri orbitali, come quelli delle effemeridi dei satelliti, sono costantemente corretti dal segmento di terra, monitorando tutta la costellazione satellitare.

Altri errori possono essere dovuti al posizionamento non ottimale dei satelliti o alla loro scarsa visibilità, ad esempio nel caso in cui il ricevitore agganci il segnale di un numero di satelliti inferiore a quattro. Si possono altresì verificare i *warm start* o *cold start* quando il ricevitore non rileva i punti GPS iniziali di un percorso a causa del tempo necessario ad agganciare il segnale dei satelliti dalla sua accensione.³⁰

Infine, le caratteristiche del territorio circostante il ricevitore possono influenzare la qualità della ricezione del segnale, fino a determinarne la perdita: è il caso delle gallerie, di intersezioni a livelli sfalsati o dell'effetto canyon, una zona urbana

²⁸ Cina A., 2004.

²⁹ Rao G.S., 2010.

³⁰ Schüssler N., Axhausen K., 2009.

circondata da edifici di notevoli altezza, che causa il *multipath error*, dovuto alla riflessione del segnale da parte di edifici, muri o ampie superfici.³¹

5. APPLICAZIONI E DISPOSITIVI MOBILI

Le applicazioni più comuni del sistema GPS sono la localizzazione del posizionamento e la navigazione assistita, che richiedono la disponibilità di una cartografia di base, tramite opportuni dispositivi (navigatori satellitari) o apposite applicazioni per tablet o smartphone. Da tali dispositivi elettronici, sempre più diffusi al giorno d'oggi, può estrarsi un'enorme mole di dati che possono essere oggetto di analisi.

La funzione di geolocalizzazione e di navigazione viene svolta dai navigatori satellitari e dai dispositivi mobili (telefoni di nuova generazione e tablet, tra gli altri).

In particolare, il navigatore satellitare è ormai diventato un oggetto di uso comune, in conseguenza del netto calo del costo dei ricevitori GPS e del loro progresso tecnologico. Nato con l'esclusiva funzione di navigazione sulla rete stradale, esso incorpora un ricevitore GPS, uno schermo (in genere LCD), un altoparlante, un processore che esegue le istruzioni, date solitamente da un sistema operativo di sviluppo aziendale, ed uno slot per schede di memoria per la memorizzazione della cartografia.

Negli ultimi anni si sono diffusi, in misura sempre maggiore, anche i ricevitori "ibridi", che svolgono la medesima funzione. Si tratta di dispositivi portatili, come personal computer, palmari, tablet o smartphone, che vengono resi idonei alla navigazione satellitare attraverso un ricevitore GPS integrato, o con il collegamento di un ricevitore GPS esterno e l'adozione di un software dedicato in grado di gestire la cartografia, nonostante sia un'altra la loro funzione principale. Nei dispositivi mobili più recenti è stato introdotto il sistema A-GPS (*Assisted GPS*), tramite cui far pervenire al terminale GPS, attraverso la rete di telefonia mobile o una rete wireless, le informazioni sui

³¹ Schüssler N., Axhausen K., 2009.

satelliti visibili dalla cella radio a cui l'utente è agganciato o dalla posizione approssimata comunicata dalla rete wireless: in tal modo, un telefono A-GPS è in grado di ricavare la propria posizione iniziale riducendo notevolmente i tempi di aggancio dei satelliti, in quanto si assume che i satelliti in vista dalla cella o dalla posizione della rete siano gli stessi visibili dal cellulare.³²

Il sistema GPS può trovare applicazione anche in altri campi ingegneristici: a titolo di esempio, nel tracciamento in tempo reale di veicoli terrestri, navali o aerei, e nel monitoraggio dei versanti o delle correnti marine.

Esso viene altresì utilizzato frequentemente per scopi fotogrammetrici, topografici e cartografici, permettendo di velocizzare notevolmente l'esecuzione delle misure ed aumentarne la loro precisione. In Italia esiste una rete di punti determinati dall'IGM (*Istituto Geografico Militare*), chiamata IGM95, i quali definiscono il datum WGS84: tali punti fissi sono stati definiti con precisione planimetrica di 2,5 cm e altimetrica di 4 cm (95% di probabilità)³³ e fanno parte della più vasta rete globale. A causa dei continui fenomeni dinamici che interessano la crosta terrestre - sia in direzione orizzontale che in quella verticale - le coordinate di questi vertici vengono spesso ricalcolate ed aggiornate.³⁴

Infine, il sistema GPS può trovare applicazione nella sincronizzazione di precisione degli orologi terrestri associati ad un ricevitore GPS, grazie alla presenza degli orologi atomici sui satelliti e al segnale orario trasmesso da questi verso la Terra.

6. SISTEMI DI POSIZIONAMENTO ALTERNATIVI

Oltre al GPS, attualmente sono in uso o in fase di sviluppo altri sistemi di posizionamento. Il sistema russo GLONASS (*GLobal Orbiting NAVigation Satellite*

³² Rao G.S., 2010.

³³ Cina A., 2004.

³⁴ Cina A., 2004.

System), anch'esso nato per scopi militari, è stato reso integralmente disponibile ai civili nel 2007.³⁵ Tuttavia, non tutti gli smartphone sono in grado di ricevere sia i segnali GPS che i segnali GLONASS. La Cina ha realizzato il sistema di posizionamento regionale Beidou per uso civile, esteso a tutta l'Asia, ed il sistema di navigazione globale COMPASS, che è in fase di completamento.³⁶

L'Unione europea (UE) e l'Agenzia Spaziale Europea (ESA) hanno in progetto il completamento di una propria rete di satelliti indipendenti già in parte attivi: il sistema di posizionamento Galileo, per scopi civili e militari, liberamente fruibile dagli utenti,³⁷ i cui dati permettono di migliorare il rilevamento della posizione.

7. RIFERIMENTI CARTOGRAFICI

Le informazioni sulla localizzazione acquisite mediante GPS o altri sistemi di posizionamento devono essere confrontate con le cartografie a disposizione o le mappe digitalizzate, in particolare nel caso del map-matching con le reti stradali. Ciò avviene mediante la tecnica della proiezione cartografica: il punto individuato sulla superficie terrestre, curva, viene proiettato su un sistema di coordinate piano, generando inevitabilmente delle distorsioni di distanze, superfici e angoli. Al fine di minimizzare le distorsioni ed i conseguenti errori possono essere utilizzate diverse tecniche di proiezione: dirette - come polari, equatoriali o oblique - o indirette, come coniche o cilindriche.

Il sistema di rappresentazione cartografica UTM (*Universal Transverse Mercator*) è quello utilizzato in genere nella cartografia mondiale, e quello a cui il sistema GPS fa riferimento. Esso utilizza una proiezione cilindrica inversa, derivata da quella di Mercatore, della superficie terrestre su un piano, tramite un cilindro con asse perpendicolare all'asse terrestre, tangente al meridiano centrale di ogni fuso. Essa è

³⁵ Rao G.S., 2010.

³⁶ Rao G.S., 2010.

³⁷ Rao G.S., 2010.

una proiezione conforme, ovvero conserva gli angoli, simmetrica rispetto all'equatore, ed in cui il meridiano centrale ha deformazione costante. Il pianeta è diviso in 60 fusi, ognuno di ampiezza pari a 6° di longitudine, che viene identificato da un numero, a partire dall'antimeridiano di Greenwich in direzione est; inoltre, esso è diviso in 20 fasce, ognuna di ampiezza pari a 8° di latitudine, che viene individuata da una lettera progressiva, da sud verso nord (dalla lettera C fino alla X, escludendo I e O). Dall'intersezione tra i 60 fusi e le 20 fasce si generano le 1200 zone del sistema UTM, ognuna individuata in modo univoco da un numero e una lettera. I punti geografici estremi dell'Italia si trovano a $47^\circ 05' 31''$ latitudine Nord (settentrionale) e $35^\circ 29' 24''$ latitudine Nord (meridionale), e a $18^\circ 31' 18''$ longitudine Est (orientale) e $6^\circ 37' 32''$ longitudine Est (occidentale); la nazione è coperta quindi dai fusi 32, 33 e 34 (longitudine $6-24^\circ$ est) e dalle fasce S e T (latitudine $32-48^\circ$ nord) in questo sistema (Fig. 1.7). Al fine di lavorare con coordinate di segno sempre positivo, è aggiunta una falsa origine alla coordinata Est di 500.000 metri. La proiezione UTM si utilizza dal parallelo di 80° sud a quello di 80° nord, mentre per i poli viene utilizzata un'altra proiezione, in quanto gli errori sarebbero eccessivi: la proiezione di Mercatore infatti distorce in maniera maggiore la dimensione e le forme degli oggetti avvicinandosi ai poli. Il datum è il sistema geodetico di riferimento che consente di definire in termini matematici la posizione dei punti sulla superficie della Terra, ovvero di passare dal punto reale ad un punto su una superficie immaginaria rappresentativa: per i sistemi di posizionamento è il WGS84 (*World Geodetic System 1984*), di tipo globale, dovendo coprire tutta la superficie terrestre, la quale è approssimata ad un ellissoide geocentrico, ovvero la cui origine degli assi coincide con il baricentro terrestre. Tale sistema è stato sviluppato dal DMA (*Defence Mapping Agency*).³⁸ Per contenere l'inevitabile deformazione viene adottata per ogni fuso una correzione, detta fattore di scala, che in Italia è pari a 0,9996, ovvero si considera il meridiano centrale non nella sua vera lunghezza ma contratto di un fattore 0,9996.

³⁸ Cina A., 2004.

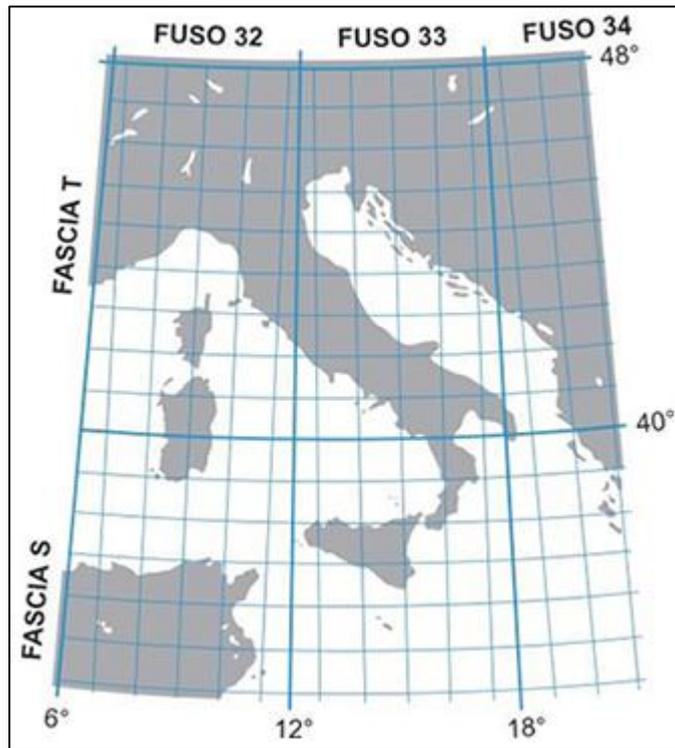


Figura 1.7: proiezione cartografica UTM dell'Italia

Il sistema UTM-WGS84 non coincide con quello nazionale da cui si origina la cartografia ufficiale italiana: quest'ultimo, infatti, è definito dall'ellissoide di Hayford (1924), orientato localmente sul vertice di Roma Monte Mario (datum Roma40, proiezione di Gauss-Boaga).³⁹ È comunque possibile passare da un sistema di riferimento ad un altro attraverso delle trasformazioni di datum, per sistemi con diversi ellissoidi di riferimento, o attraverso delle trasformazioni di coordinate, al fine di rendere omogenee le misure.

³⁹ Cina A., 2004.

8. MAPPE DIGITALIZZATE E AMBIENTE GIS

Le mappe digitalizzate sono di fondamentale importanza in quanto rappresentano la rete stradale esistente, con la quale vanno confrontati i dati di posizionamento acquisiti dai sistemi satellitari. Inoltre, esse possono essere caricate su piattaforma GIS (*Geographic Information System*), un sistema informativo computerizzato che permette l'acquisizione, l'elaborazione, l'analisi, la gestione e la rappresentazione grafica di dati di tipo georeferenziato, ovvero attributi riferiti alla posizione o agli elementi geografici presenti sulla mappa stessa. Il GIS è l'ambiente software in cui caricare, estrapolare e manipolare tali dati: in tal modo è possibile rappresentare in modo immediato ed efficace le differenti proprietà di una regione, evidenziandone le caratteristiche direttamente sulla mappa e facilitandone la lettura. I GIS fanno parte dei SIT (*Sistemi Informativi Territoriali*), ovvero il complesso di risorse, strumenti e procedure che permettono l'acquisizione e la distribuzione dei dati.

I dati sono organizzati in strutture di tipo database, relazionati agli elementi della mappa, e contengono in genere tre tipologie di informazioni: geometriche, topologiche e informative. Le informazioni geometriche sono relative alla rappresentazione cartografica degli oggetti rappresentati, quali la forma (punto, linea, poligono), la dimensione e la posizione geografica (coordinate); le informazioni topologiche sono riferite alle relazioni reciproche tra gli oggetti, come connessione (nel caso di reti), adiacenza o inclusione; infine i dati informativi riguardano gli attributi quantitativi o testuali, associati ad ogni elemento.

Le informazioni possono essere codificate in due modi diversi, attraverso dati di tipo vettoriale o raster. I dati vettoriali sono costituiti da elementi semplici, quali punti, linee e poligoni, codificati e memorizzati sulla base delle loro coordinate. A ciascun elemento è associato un record del database informativo che contiene tutti gli attributi dell'oggetto rappresentato.

I dati raster sono composti da una matrice di celle, generalmente di forma quadrata o rettangolare, a cui sono associate le informazioni. In genere i dati vettoriali sono adatti alla rappresentazione di dati che variano in modo discreto (ad esempio per indicare

servizi di tipo puntuale), mentre i dati raster sono più adatti alla rappresentazione di dati che variano in modo continuo (come i modelli digitali di elevazione).

Anche le mappe possono essere soggette ad errori, dovuti alla mancanza di informazioni (ad esempio l'assenza di alcuni archi della rete) o al processo di creazione o di digitalizzazione (ad esempio la larghezza di una strada o l'associazione degli attributi agli elementi della rete), che possono condurre ad un carente o errato processo di map-matching. Tali errori possono essere stimati usando sia la scala della mappa sia prove sul campo.⁴⁰

⁴⁰ Quddus M. et al., 2006.

CAPITOLO II: IL MAP-MATCHING

1. IL PROCEDIMENTO DI MAP-MATCHING

Negli ultimi anni si è assistito ad una notevole crescita della disponibilità di dati GPS e della loro qualità ed accuratezza, dovuta alla diffusione ed all'evoluzione tecnologica di smartphone e di applicazioni di localizzazione e navigazione per dispositivi mobili.⁴¹ L'enorme mole di dati a disposizione degli analisti ha quindi condotto alla necessità di elaborare e di migliorare continuamente gli algoritmi di map-matching, rispetto sia alla loro accuratezza, sia alla loro efficienza computazionale.

Il processo di map-matching consiste nell'integrare due tipologie di dati: quelli di localizzazione - solitamente una lista ordinata di coordinate geografiche -, ricavati tramite un sistema di localizzazione GNSS (*Global Navigation Satellite System*), tipo GPS, e quelli delle mappe digitali. I due elementi essenziali per la procedura sono quindi uno strumento che traccia la posizione del veicolo, ed una mappa digitale della rete, che fa da riferimento per la localizzazione.

Lo scopo del procedimento è identificare il corretto arco che un veicolo sta percorrendo - o ha percorso - e determinarne l'esatta posizione sullo stesso.⁴² In genere, nell'ambito d'interesse dell'ingegneria dei trasporti, l'obiettivo è identificare o ricostruire il percorso scelto dall'utente per lo spostamento, pertanto il problema del map-matching si traduce nel ricavare il percorso che rappresenti la migliore stima dell'itinerario realmente scelto dall'utente.⁴³

La questione è affrontata di solito tramite software dedicati su piattaforma GIS (*Geographic Information System*), un sistema informativo computerizzato che permette l'acquisizione, l'elaborazione, l'analisi, la gestione e la rappresentazione

⁴¹ Schweizer J., Rupi F., 2014.

⁴² Quddus M., Washington S., 2015.

⁴³ Marchal F. et al., 2004.

grafica mediante l'utilizzo di mappe digitali di dati di tipo georeferenziato, ovvero database di dati (*attributi*) associati ad elementi geografici.

Nelle figure seguenti sono rappresentati i passi del procedimento di map-matching: dapprima vengono localizzati i punti (Fig. 2.1), poi essi vengono confrontati con la mappa digitalizzata (Fig. 2.2) ed infine è ricostruito il percorso scelto (Fig. 2.3). Ogni punto costituisce una rilevazione satellitare della posizione ad un dato istante di tempo: all'aumentare dell'intervallo di rilevazione, il numero dei punti sarà minore, e di conseguenza l'insieme di dati sarà meno dettagliato.

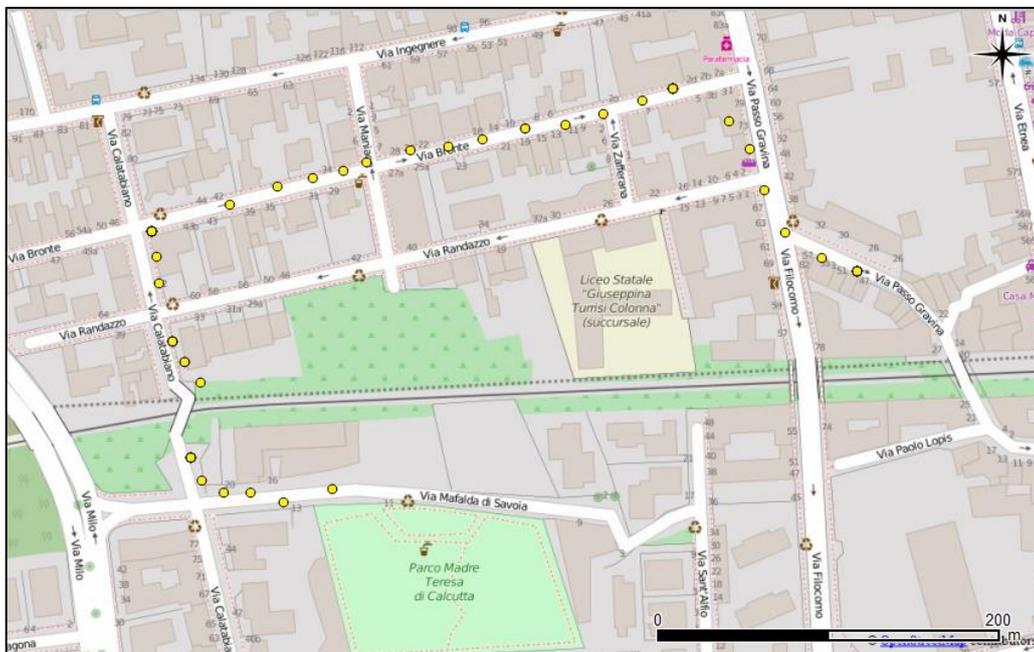


Figura 2.1: localizzazione dei punti GPS (in giallo)

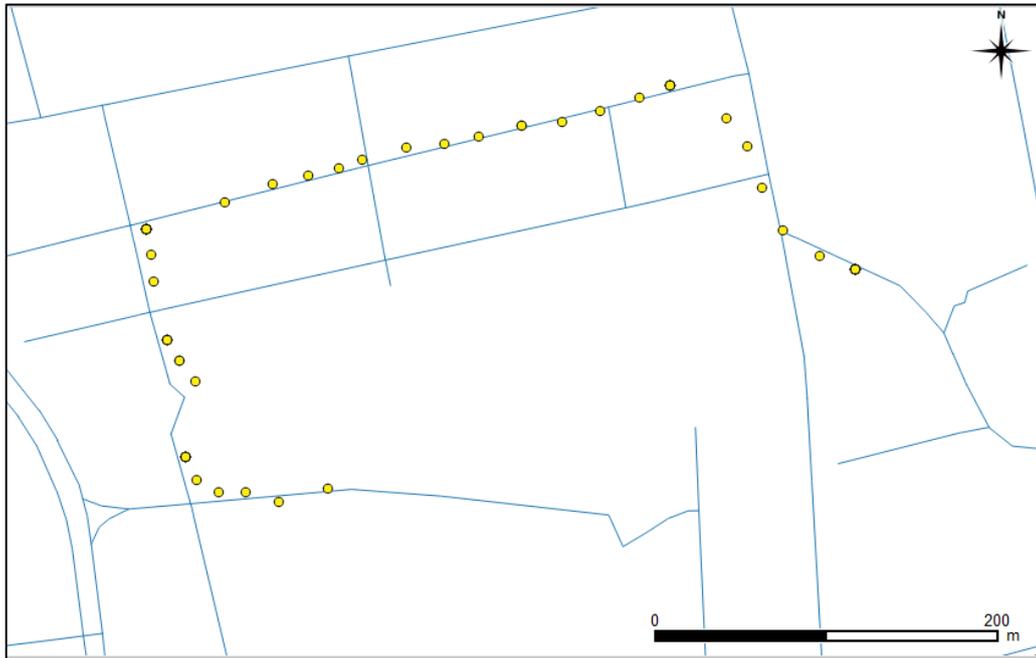


Figura 2.2: punti GPS (in giallo) confrontati con la mappa digitalizzata (in blu)

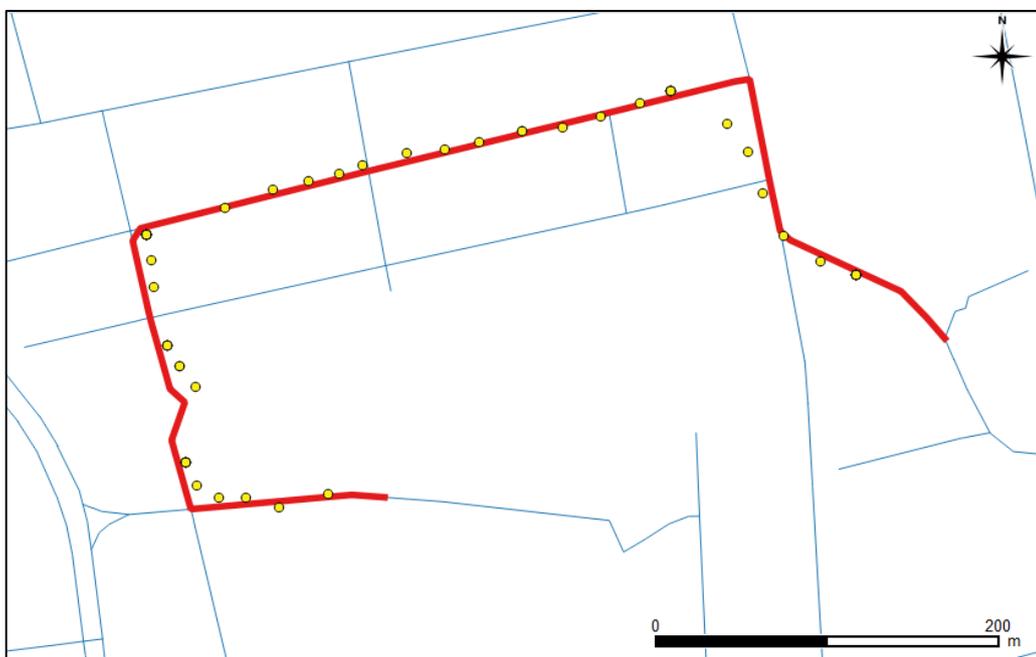


Figura 2.3: individuazione del percorso scelto (in rosso)

Nella figura successiva è invece riportato un esempio di map-matching esposto in letteratura: a partire dai punti tracciati è valutato il percorso, evidenziato sulla rete.

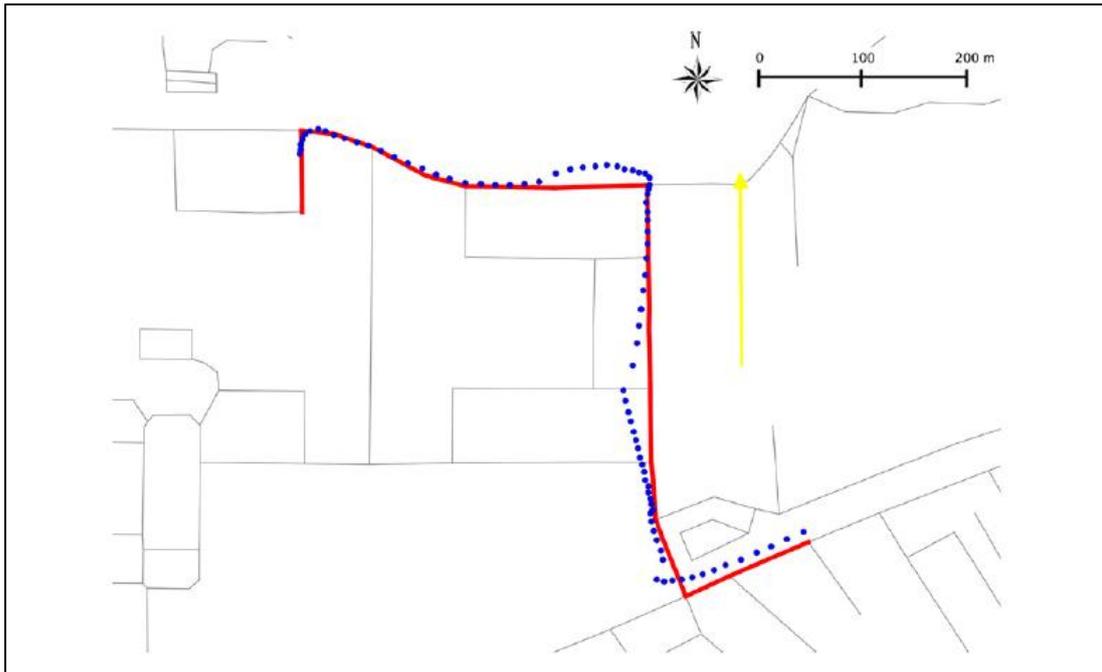


Figura 2.4: map-matching (Bierlaire et al., 2012)⁴⁴

2. ORIGINI DEI DATI

Le origini dei dati da processare sono i sistemi di posizionamento e localizzazione, tipo il GPS, e le reti disponibili, in genere scaricabili gratuitamente dal web, come la rete OSM (*OpenStreetMap*).⁴⁵

⁴⁴ Bierlaire M. et al., 2012.

⁴⁵ OpenStreetMap, <http://www.openstreetmap.org>

I dati ricavati dai sistemi di localizzazione tipo GPS costituiscono le tracce, ovvero l'insieme delle informazioni riguardanti uno spostamento. Esse sono disponibili in forma digitale, in genere come files in formato .gpx (*GPS eXchange Format*), .csv (*Comma Separated Values*), o altri formati meno diffusi come .tcx o .kml. Ogni traccia $T = \{ P_1, P_2, \dots, P_n \}$ è costituita da una successione ordinata di n punti P_i (*trackpoints*), identificati dalle coordinate geografiche longitudine e latitudine, e ad ogni coppia sono associati una stima dell'altitudine e l'istante di tempo in cui il rilevamento viene effettuato, detto *timestamp*. A questi dati fondamentali possono essere aggiunte altre informazioni, come un numero identificativo (ID) del punto tracciato, dello spostamento o dell'utente, una stima della velocità o della pendenza, o alcuni commenti sul percorso.

Le reti disponibili sono invece dei grafi a due dimensioni $G(V, E)$, costituiti da un insieme E di archi orientati (*links* o *edges*), rappresentanti dei segmenti di strada, formati da una o più corsie (*lanes*), e da un insieme V di nodi (*nodes* o *vertexes*), rappresentanti le intersezioni o i punti terminali degli archi, che contengono informazioni sulle possibili manovre di svolta o di attraversamento (*connections*); solitamente le informazioni geometriche di posizionamento sono riferite all'asse stradale. Le reti possono essere correlate a dei database che associano determinate caratteristiche agli elementi geometrici della rete stessa; tali attributi sono visualizzabili ed eventualmente modificabili su piattaforma GIS. I dati associati possono riguardare gli archi, ad esempio la denominazione (*name*), il grado di importanza (*highway, secondary, ecc.*), il suo numero di corsie e la larghezza (*lanes e width*), la tipologia di veicoli ammessi (*cycleway, railway, footway, busway, sidewalk, ecc.*), o restrizioni sull'accesso (*delivery, oneway, noexit, ecc.*), ma anche il territorio (*landuse*) o i nodi (*roundabout, traffic light signal, ecc.*). Nella figura seguente è riportato un esempio di rete.



Figura 2.5: rete di Bologna (centro), tratta da OpenStreetMap⁴⁶

3. SCOPI E APPLICAZIONI

Il procedimento di map-matching può rappresentare un utile strumento per il monitoraggio e la gestione dei sistemi di trasporto, in ordine sia all'acquisizione dei dati che all'analisi dei risultati.

Per quanto concerne il primo punto, i dati sono raccolti in modo sistematico e non intrusivo;⁴⁷ il tracciamento dei punti GPS ha dimostrato di possedere un ottimo potenziale anche come alternativa o integrazione alle classiche indagini a campione

⁴⁶ OpenStreetMap, <http://www.openstreetmap.org>

⁴⁷ Bierlaire M. et al., 2012.

sugli spostamenti degli utenti, per gli aspetti riguardanti i tempi e le distanze di viaggio, i modi di trasporto ed i motivi dello spostamento.⁴⁸

In ordine al secondo aspetto, i risultati del map-matching possono essere usati per meglio comprendere il comportamento degli utenti sulla rete e le conseguenze di tale comportamento sia sul sistema di trasporto, sia sulle esternalità, come l'inquinamento atmosferico.⁴⁹ Una migliore comprensione della condotta dell'utente all'interno della rete di trasporto, in merito alle scelte modali o del percorso, è infatti di fondamentale importanza nell'ambito della pianificazione dei trasporti, al fine di adottare provvedimenti che promuovano i modi di trasporto sostenibili, come la mobilità ciclabile,⁵⁰ riducendo la dipendenza dai veicoli privati motorizzati ed incoraggiando l'impiego di modi di trasporti alternativi a basso impatto.⁵¹ Inoltre, viene resa possibile una calibrazione più efficace dei modelli di scelta⁵² ed una migliore valutazione delle preferenze degli utenti, sulla base della scelta del percorso e dell'utilità associata ai potenziali itinerari alternativi.⁵³ tali informazioni possono essere utili, ad esempio, nella valutazione dell'effettivo uso della rete ciclabile e delle zone che richiedono dei miglioramenti.⁵⁴ Difatti, è ormai assolutamente necessario perseguire politiche di mobilità sostenibile, specialmente in ambito urbano, al fine di migliorare le condizioni ambientali e la salute dei cittadini; in particolare, per una buona ed efficace progettazione delle reti ciclabili è indispensabile conoscere le preferenze degli utenti e fornire loro sicurezza e comfort.⁵⁵

Attraverso i dati provenienti da dispositivi mobili (*FCD: floating car data*), se processati in tempo reale, è possibile altresì redigere una valutazione immediata delle prestazioni dei singoli archi e dell'intera rete, ad esempio in termini di saturazione e di tempi di viaggio, essendo i FCD geolocalizzati e contenendo informazioni temporali. Insieme alle tecniche informatiche di modellazione e di simulazione del

⁴⁸ Bohte W., Maat K., 2009.

⁴⁹ Quddus M. et al., 2003.

⁵⁰ Menghini G. et al., 2009.

⁵¹ Shannon T. et al., 2006.

⁵² Menghini G. et al., 2009.

⁵³ Schweizer J., Rupi F., 2014.

⁵⁴ Schweizer J., Rupi F., 2014.

⁵⁵ Joo S. et al., 2015.

traffico ed alle reti di comunicazione, infine, gli algoritmi di map-matching possono consentire un monitoraggio più efficace e migliorare le performance nell'ambito dei Sistemi di Trasporto Intelligenti (ITS), le cui applicazioni spaziano dall'assistenza all'utente nella scelta del percorso, alla gestione di situazioni di emergenza ed incidenti, alla gestione del servizio di trasporto pubblico e di altri servizi LBS (*Location Based Services*), che richiedono informazioni sul posizionamento.⁵⁶ Ad esempio, dei bus equipaggiati con un sistema di navigazione possono localizzare la propria posizione in tempo reale ed inviare i loro dati ad un centro di controllo; questo li elabora ed è così in grado di prevedere l'arrivo del medesimo bus ed il tempo di attesa alle fermate, o di programmare il ciclo semaforico all'incrocio successivo, in modo da diminuirne l'attesa, incrementando in tal modo il livello di servizio del sistema di trasporto pubblico.⁵⁷

4. PROBLEMATICHE PIÙ COMUNI

Se i dati GPS non fossero affetti da errori, e le mappe digitali fossero ad alta risoluzione ed esenti da imprecisioni, non ci sarebbe bisogno di applicare gli algoritmi di map-matching, dato che i punti tracciati sarebbero automaticamente correlati agli archi e ai nodi corrispondenti sulla rete. Tuttavia, la presenza di tali errori rende necessario lo sviluppo di algoritmi per il map-matching sempre più sofisticati, che filtrano i dati e tengono conto delle situazioni anomale che possono verificarsi sulla rete.

Come già visto nel quarto paragrafo del primo capitolo, gli errori sul posizionamento GPS possono classificarsi in errori accidentali di misura, errori sistematici o di modello (*bias*), ed errori di osservazione; complessivamente la precisione è dell'ordine della decina di metri - 9 metri in un intervallo di confidenza al 95%, secondo i dati ufficiali del Dipartimento della Difesa del Governo Statunitense.⁵⁸ Gli errori dipendono

⁵⁶ Quddus M. et al., 2007.

⁵⁷ Quddus M. et al., 2007.

⁵⁸ USA Government – Department of Defense, 2008.

principalmente dalla portante (L1 o L2), dal ritardo di propagazione del segnale che si genera in atmosfera, in particolare nella troposfera e nella ionosfera, dalla sincronizzazione degli orologi, e dalle caratteristiche fisiche dell'ambiente circostante.

Inoltre, per il map-matching è di fondamentale importanza anche l'intervallo di ricezione del segnale: il rilevamento del posizionamento tramite GPS, infatti, non è continuo, ed un intervallo di notevoli dimensioni tra due misurazioni successive, quindi un rilevamento a bassa frequenza, può rischiare di causare una rilevante perdita di potenziali informazioni, rendendo molto arduo il tentativo di matching della traccia. Difatti, la precisione nella ricostruzione dei percorsi diminuisce con dei dati GPS molto sparsi: una quantità maggiore di dati fornisce maggiori informazioni e rende minore l'incertezza.⁵⁹ In generale la frequenza di osservazione richiesta dipende dallo scopo del map-matching stesso: quest'aspetto è misurato in parte dal parametro di performance di navigazione richiesta "continuità" (*continuity*).⁶⁰

Ulteriori circostanze penalizzanti possono verificarsi nel caso i dati siano raccolti tramite smartphone: la maggior parte delle persone, infatti, tiene il proprio cellulare in tasca, giacca, o in borsa, indebolendo ancora di più la ricezione del segnale,⁶¹ che può essere ridotto anche da cover o scocche di rivestimento particolari schermanti l'antenna ricevente.

La correttezza del map-matching dipende, in larga misura, dalla qualità della mappa usata: una mappa di bassa qualità condurrà ad errori notevoli.⁶²

Purtroppo anche le reti e le mappe digitali sono affette da imprecisioni, originate principalmente dal processo di creazione e di digitalizzazione della mappa,⁶³ anche se sempre minori con l'avanzamento tecnologico, e che in genere possono essere stimate utilizzando la scala della mappa o misurazioni sul posto.⁶⁴ La mancanza di un arco della rete, ad esempio, o l'errata attribuzione di caratteristiche sulle restrizioni all'accesso ad esso associate, come nel caso di una pista ciclabile o di una strada

⁵⁹ Quddus M. et al., 2007.

⁶⁰ Quddus M. et al., 2007.

⁶¹ Bierlaire M. et al., 2012.

⁶² Quddus M. et al., 2007.

⁶³ Quddus M. et al., 2006.

⁶⁴ Quddus M. et al., 2006.

facente parte di una zona ad accesso limitato, conducono inevitabilmente ad un matching errato, in misura più o meno grande; lo stesso effetto può derivare anche da connessioni errate o mancanti ai nodi. Altre imprecisioni possono presentarsi nella rappresentazione degli archi curvilinei o di intersezioni particolarmente complesse, o di ampie rotatorie, spesso raffigurate in maniera molto semplificata come semplici nodi.⁶⁵

Oltre ai suddetti errori, definiti topologici, ovvero originati dalle omissioni e dalle semplificazioni dell'ambiente reale, possono verificarsi anche errori di tipo geometrico, ovvero la non esatta corrispondenza tra la posizione reale e quella indicata sulla mappa di un dato elemento come un arco o un nodo,⁶⁶ gli archi, in particolare, sono rappresentati tramite linee posizionate generalmente in corrispondenza dell'asse stradale, non divulgando quindi informazioni sulla larghezza delle corsie.

Il primo problema affrontato da un algoritmo di map-matching consiste nell'individuazione del punto o dell'arco di origine dello spostamento: ciò può essere particolarmente difficile nel caso di *warm start*, in cui i primi punti dello spostamento vengono persi o tracciati molto approssimativamente, o nelle reti urbane ad alta densità: l'arco più vicino al punto tracciato può non rappresentare sempre la scelta più corretta.⁶⁷

Nel caso in cui l'utente sia fermo in un determinato luogo (Fig. 2.6), può verificarsi un tracciamento di posizioni consecutive diverse, dovuto agli errori di localizzazione, capace di indurre un algoritmo di map-matching in errore, specialmente se questo utilizza tra le misurazioni la quantità di punti associati ad un determinato arco.

⁶⁵ Quddus M. et al., 2007.

⁶⁶ Quddus M. et al., 2006.

⁶⁷ Quddus M. et al., 2006.

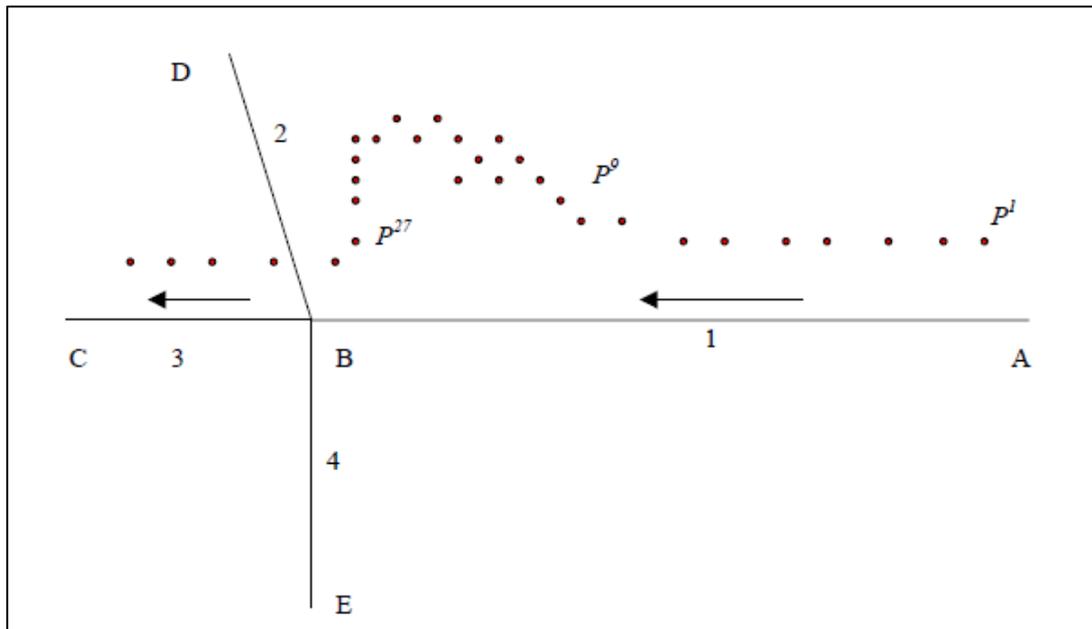


Figura 2.6: tracciamento dei punti GPS a veicolo fermo (punti P9 - P27) (Quddus et al., 2003)⁶⁸

Una situazione particolarmente impegnativa, generalmente in ambito urbano, è quella delle intersezioni a Y (*Y-junctions*), rappresentata nella figura seguente 2.7: individuare l'arco corretto tra BC e BD, avendo tracciato il solo punto P3, può rivelarsi molto complicato, specialmente per metodologie di map-matching basate sul concetto di distanza.

⁶⁸ Quddus M. et al., 2003.

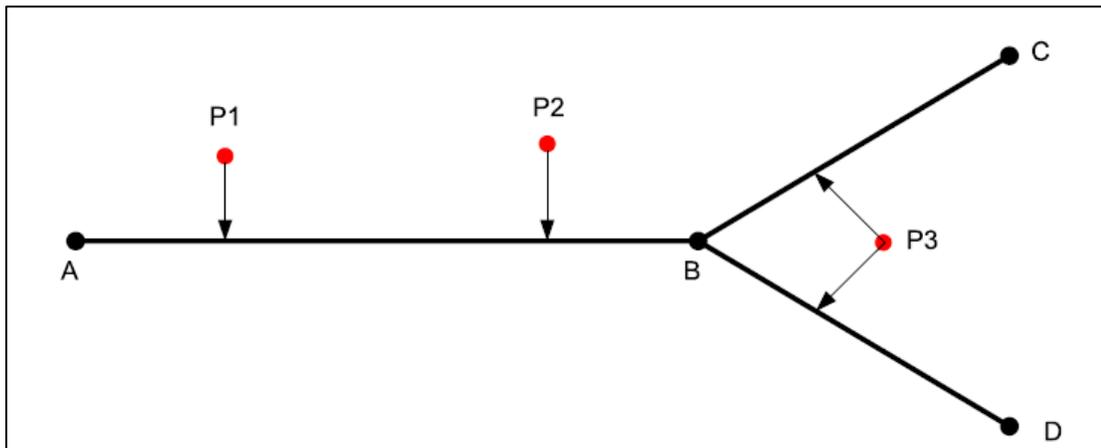


Figura 2.7: map-matching in intersezione a Y (Quddus et al., 2007)⁶⁹

Oltre a ciò, una delle assunzioni comunemente riconosciute in letteratura è che il veicolo sia vincolato a muoversi esclusivamente sulla rete: benché ciò sia valido per la maggior parte dei veicoli e nella maggior parte dei casi, potrebbero sorgere dei problemi per luoghi esterni alla rete, come parcheggi o terreni privati.⁷⁰ Il suddetto problema è di specifica influenza per gli spostamenti in bicicletta: difatti, gli utenti della mobilità ciclabile sono inclini a seguire percorsi particolari, ad esempio all'interno di parchi, i cui sentieri sono raramente riportati nelle reti, o ad attraversare zone pedonali, difficilmente ricostruibili, oppure a percorrere strade poco trafficate o corsie preferenziali in direzione opposta al senso di marcia consentito; ciò rende particolarmente difficile rintracciare i reali percorsi dei ciclisti. Il vincolo è invece fortissimo per i veicoli del trasporto pubblico, anche su sede non vincolata, i quali seguono un percorso prestabilito, rendendo molto più semplice l'individuazione in tempo reale dell'arco in cui si trovano.

Infine, un ultimo problema da affrontare, ma non meno importante, consiste nell'entità del costo computazionale richiesto per l'esecuzione dell'algoritmo: sebbene i miglioramenti nelle tecnologie hardware di processori e di memorie siano continui,

⁶⁹ Quddus M. et al., 2007.

⁷⁰ Quddus M. et al., 2007.

ridurre al minimo il numero di operazioni da eseguire e i tempi di esecuzione senza eccessive perdite di precisione rimane una delle priorità.

Data la notevole quantità di errori che possono essere prodotti, una validazione dell'algoritmo di map-matching è essenziale al fine di produrre statistiche che valutino la sua efficacia in termini di corretta identificazione degli archi e di determinazione della posizione del veicolo,⁷¹ e di scovare le eventuali anomalie nel matching. In letteratura gli autori fanno spesso riferimento alla “accuratezza” (*accuracy*), ovvero alla percentuale di segmenti di percorso correttamente individuati,⁷² ed alla “integrità” (*integrity*), ovvero il livello di fiducia riponibile nelle informazioni fornite dal sistema, comprendente quindi la capacità di segnalare eventuali malfunzionamenti.⁷³

5. CLASSIFICAZIONE DEGLI ALGORITMI DI MAP-MATCHING

Gli algoritmi sviluppati da studiosi e ricercatori, al fine di eseguire l'operazione di map-matching, e disponibili in letteratura sono numerosi e di svariato tipo. Tali algoritmi utilizzano differenti tecniche di analisi matematica, come l'analisi geometrica, l'analisi topologica spaziale della rete, o altre tecniche più complesse, come le teorie probabilistiche, il filtro di Kalman o la logica fuzzy.

Innanzitutto occorre precisare la relatività delle caratteristiche degli algoritmi di map-matching, in base al loro scopo o alla loro area di applicazione: alcuni di essi, infatti, possono offrire delle ottime prestazioni in zone suburbane, ma non essere appropriati per zone urbane ad alta densità di rete, o viceversa;⁷⁴ inoltre, i valori dei parametri utilizzati possono variare notevolmente in base al differente ambiente di lavoro.⁷⁵

⁷¹ Quddus M. et al., 2007.

⁷² Schüssler N., Axhausen K., 2009.

⁷³ Ochieng W. et al., 2003.

⁷⁴ Quddus M. et al., 2007.

⁷⁵ Quddus M. et al., 2007.

Una prima classificazione può essere eseguita in base al periodo temporale durante il quale viene operato il processo di map-matching: gli algoritmi di tipo *on-line* lavorano in tempo reale sui dati, nel momento stesso dell'acquisizione - è il caso degli algoritmi implementati ad esempio nei navigatori satellitari. Tali applicazioni in *real time* richiedono una risposta quasi immediata e possono usufruire esclusivamente dei punti tracciati fino a quel momento: ciò impone una maggiore attenzione ai tempi di esecuzione piuttosto che all'accuratezza del risultato.⁷⁶ Gli algoritmi di tipo *off-line*, invece, processano i dati in un momento successivo alla loro totale acquisizione (*post-processing*) ed in genere forniscono risultati più accurati degli algoritmi *on-line*, non soltanto perché il costo computazionale riveste una minore importanza, ma anche perché in questi casi è possibile analizzare la completa successione dei punti, il che spesso aiuta l'algoritmo a selezionare il corretto percorso, specialmente alle intersezioni.⁷⁷

È possibile classificare gli algoritmi anche in base al loro modo di affrontare il problema in quattro gruppi: si distinguono approccio geometrico, topologico, probabilistico ed altre tecniche avanzate.

L'approccio di tipo geometrico si basa sulle informazioni geometriche della rete, considerando esclusivamente la geometria degli archi, senza prendere in considerazione il modo in cui gli archi sono connessi tra di loro; generalmente il concetto geometrico fondamentale tenuto in considerazione è la distanza tra il punto GPS tracciato e la rete. In particolare, se l'algoritmo associa ad ogni punto tracciato il nodo più vicino della rete, esegue un matching detto *point-to-point* (Bernstein et Kornhauser, 1998)⁷⁸; quest'approccio è molto veloce e semplice da implementare, ma sensibile al modo in cui la rete è stata creata.⁷⁹ Se invece l'algoritmo associa ogni punto tracciato alla curva - invero una successione di spezzate - più prossima ad esso, come in figura 2.8, esegue un matching di tipo *point-to-curve* (White et al., 2000)⁸⁰: quest'approccio è migliore del precedente, ma genera prestazioni insufficienti in reti

⁷⁶ Câmara Pereira F. et al., 2009.

⁷⁷ Câmara Pereira F. et al., 2009.

⁷⁸ Bernstein D., Kornhauser A., 1996.

⁷⁹ Quddus M. et al., 2007.

⁸⁰ White C. et al., 2000.

altamente dense.⁸¹ Infine, se l'algoritmo, dato un nodo iniziale, associa la curva della traiettoria (l'insieme di spezzate colleganti i punti tracciati) alla più vicina curva della rete (una successione di spezzate uscenti dal nodo iniziale) esegue un matching di tipo *curve-to-curve* (Phuyal, 2002):⁸² quest'approccio è sensibile ai punti estremi e dipende dalla scelta del nodo origine.⁸³

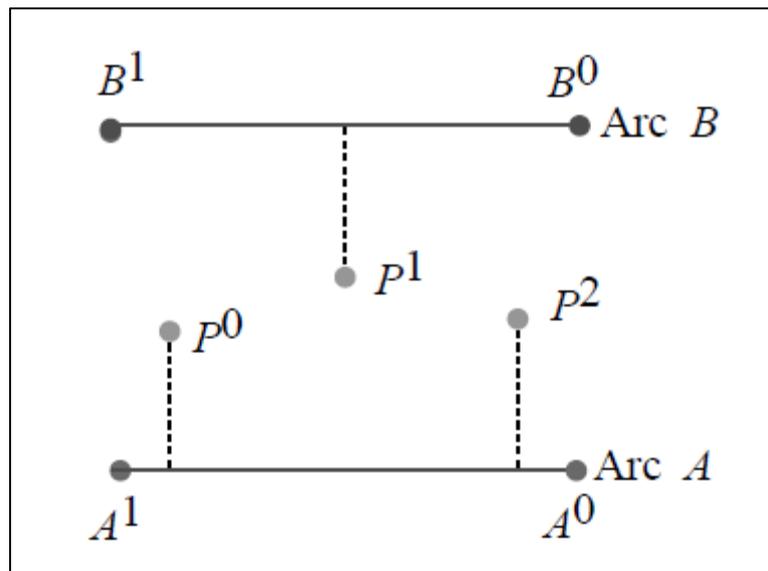


Figura 2.8: approccio geometrico *point-to-curve*, con matching errato del punto P1 (White et al., 2000)⁸⁴

L'approccio di tipo topologico prende in considerazione, invece, sia la geometria della rete, sia le sue caratteristiche topologiche, quali connettività e contiguità, ovvero le relazioni tra le entità geometriche costituenti la rete stessa (Fig. 2.9). Altre relazioni topologiche possono essere, oltre alla connettività degli archi, l'adiacenza nel caso dei poligoni e l'essere contenuto nel caso dei punti in un poligono.⁸⁵ Tali informazioni

⁸¹ Quddus M. et al., 2007.

⁸² Phuyal B., 2002.

⁸³ Quddus M. et al., 2007.

⁸⁴ White C., Bernstein D., Kornhauser A., 2000.

⁸⁵ Quddus M. et al., 2007.

possono essere pesate in maniera diversa nell'esecuzione di un algoritmo di map-matching di tipo topologico; in ogni caso, l'utilizzo di ulteriori informazioni oltre a quelle geometriche genera solitamente un miglioramento dell'efficacia del matching, sebbene quest'approccio sia sensibile ai punti che si collocano in posizioni estreme rispetto alle altre osservazioni (*outliers*). Un esempio di algoritmo topologico è stato fornito da Quddus et al.(2003).⁸⁶

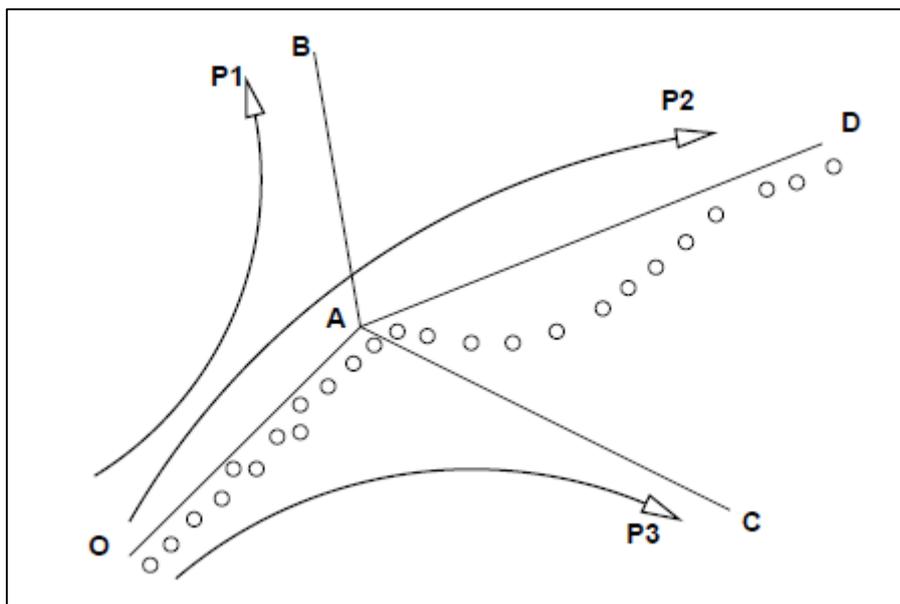


Figura 2.9: approccio topologico: tiene in considerazione i possibili archi uscenti dal nodo A, a cui corrispondono diversi percorsi (Marchal et al., 2004)⁸⁷

L'approccio di tipo probabilistico richiede la definizione di una regione di confidenza, in genere di forma ellittica - assumendo che l'errore abbia una distribuzione di tipo Gaussiano⁸⁸ - o rettangolare, centrata nel punto GPS tracciato, dipendente dalla precisione dello strumento di navigazione (Fig. 2.10). Tale regione è sovrapposta alla

⁸⁶ Quddus M. et al., 2003.

⁸⁷ Marchal F. et al., 2004.

⁸⁸ Quddus M. et al., 2007.

rete al fine di identificare l'arco su cui il veicolo sta viaggiando; ai possibili candidati ricadenti in tale regione vengono quindi applicati dei criteri di scelta (vicinanza, connettività, heading) con lo scopo di optare per quello con maggiore probabilità di costituire l'arco effettivamente percorso. Un esempio di algoritmo probabilistico è stato fornito da Ochieng et al.(2004).⁸⁹

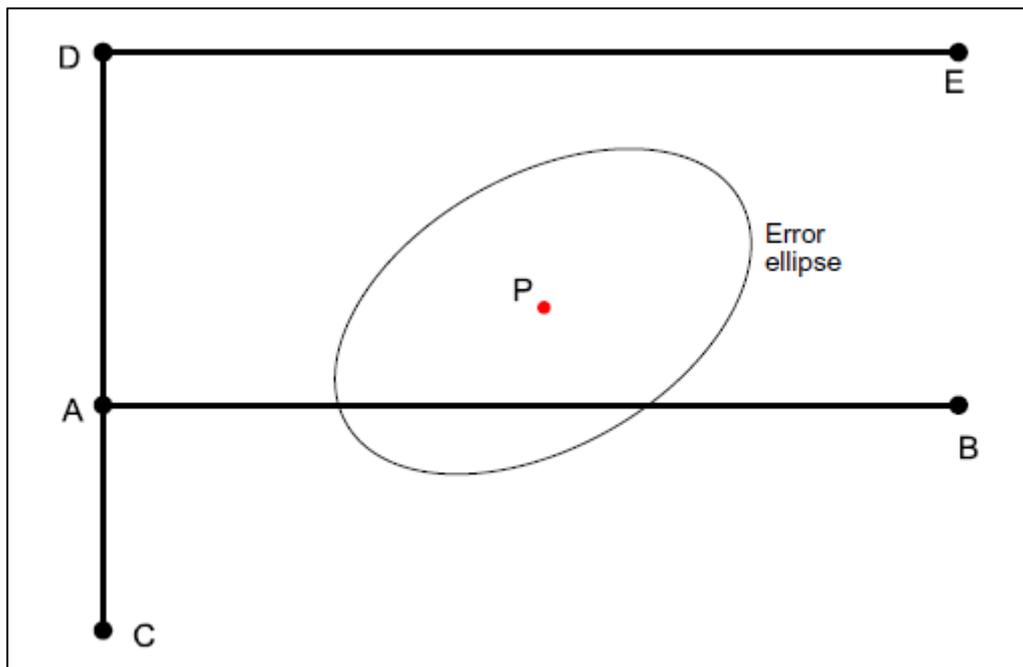


Figura 2.10: regione di errore ellittica del punto P (Quddus et al., 2007)⁹⁰

Infine, l'approccio di tipo avanzato si riferisce all'uso di strumenti matematici meno comuni e più complessi, come il filtro di Kalman (Kim et al., 2000),⁹¹ utilizzato per la valutazione di misure soggette ad errore, la teoria dell'evidenza (Yang et al., 2003),⁹² che valuta l'evoluzione delle probabilità nel tempo in base ai dati via via processati, o

⁸⁹ Ochieng W. et al., 2004.

⁹⁰ Quddus M. et al., 2007.

⁹¹ Kim W. et al., 2000.

⁹² Yang D. et al., 2003.

i modelli a logica fuzzy (Fu et al., 2004),⁹³ che attribuiscono ad ogni proposizione un grado di verità compreso tra 0 (certamente falso) e 1 (certamente vero).

Il principale svantaggio dei suddetti approcci è il fatto che ad ogni valutazione del punto GPS tracciato corrisponde l'assegnazione ad un arco o ad un percorso che rimangono in memoria: di conseguenza, l'analisi del successivo punto GPS si basa sul corretto matching del punto precedente, e ciò implica purtroppo che l'identificazione errata di un arco conduca all'individuazione di un percorso sbagliato.⁹⁴ Al fine di superare tale problema, è stata proposta in letteratura l'adozione della tecnica delle ipotesi multiple (*MHT: Multiple Hypothesis Technique*),⁹⁵ tramite la quale, prendendo in considerazione delle misure effettuate all'interno di una regione di errore definita a priori, si conserva in memoria un set di possibili alternative per il percorso effettivo, si assegna loro un punteggio (*score*) e si sceglie tra queste quella con il migliore risultato.⁹⁶

Alcune soluzioni proposte in letteratura (Marchal et al., 2004;⁹⁷ Schüssler et Axhausen, 2009⁹⁸) che adottano la tecnica delle ipotesi multiple effettuano una ricerca topologica degli archi uscenti da un nodo che si ritiene raggiunto: ad ogni passo sono formulate delle ipotesi sul percorso, che vengono tenute in memoria fino ad un certo numero limite (*threshold*), in base alle caratteristiche della rete, e calcolate le probabilità ad esse associate, espresse attraverso un punteggio, tramite cui si procede ad ogni passo scartando le ipotesi meno verosimili, fino all'identificazione del percorso più probabile.

⁹³ Fu M. et al., 2004.

⁹⁴ Schüssler N., Axhausen K., 2009.

⁹⁵ Pyo J. et al., 2001.

⁹⁶ Pyo J. et al., 2001.

⁹⁷ Marchal F. et al., 2004.

⁹⁸ Schüssler N., Axhausen K., 2009.

CAPITOLO III: STUDIO DEI METODI DI MAP-MATCHING

1. DEFINIZIONE DELLO STUDIO, PIATTAFORME UTILIZZATE E RACCOLTA DATI DI INPUT

Al fine di sviluppare una metodologia di map-matching efficace in termini di accuratezza dei risultati e di efficienza computazionale, sono stati studiati quattro differenti algoritmi, corrispondenti a due diverse metodologie di approccio al problema. Tali algoritmi sono:

- Algoritmo denominato ‘orig’, sviluppato da J.Schweizer e F.Rupi (2014)⁹⁹, di tipo probabilistico.
- Algoritmo denominato ‘new’, proposto come versione aggiornata del precedente.
- Algoritmo denominato ‘Marchal’, sviluppato da F.Marchal, J.Hackney e K.W.Axhausen (2004)¹⁰⁰, di tipo topologico-probabilistico, che impiega la tecnica delle ipotesi multiple.
- Algoritmo denominato ‘NewMarchal’, proposto nel presente lavoro come versione aggiornata del precedente.

Essi sono stati testati su un database di spostamenti di ciclisti nell’area urbana di Bologna, registrati attraverso i loro smartphone sotto forma di punti GPS, e sulla relativa rete.

I dati degli spostamenti di ogni ciclista sono costituiti da una serie di tracce, ognuna formata da una successione ordinata di punti individuati dal GPS dello smartphone, e registrate attraverso l’applicazione Endomondo¹⁰¹: i dati sono stati raccolti nell’ambito

⁹⁹ Schweizer J., Rupi F., 2014.

¹⁰⁰ Marchal F. et al., 2004.

¹⁰¹ Endomondo, <http://www.endomondo.com>

della “European Cycling Challenge”,¹⁰² a cui ha partecipato anche la città di Bologna tramite centinaia di ciclisti che hanno caricato in forma anonima e volontaria i loro spostamenti in bicicletta nel mese di Maggio del 2013.

Ogni traccia $T = \{ P_1, P_2, \dots, P_n \}$ è costituita da una successione ordinata di n punti P_i identificati tramite le coordinate geografiche longitudine e latitudine (*longitude_i*, *latitude_i*) e l’istante di tempo in cui il rilevamento viene effettuato (*timestamp_i*); ad ogni punto sono poi associate, ove disponibili, altre informazioni accessorie, quali il numero identificativo del punto (*id_i*), il numero identificativo della traccia (*pointPathId*), una stima dell’altitudine (*altitude_i*), della distanza percorsa (*distance_i*), del battito cardiaco (*heartRate_i*) e della velocità (*speed_i*). Il rilevamento della posizione è effettuato mediamente ogni 5 - 10 secondi e la sua precisione è stimabile in circa $\pm 20 m$, data la minore precisione delle antenne GPS degli smartphone e l’effetto di oscuramento degli edifici presenti nelle zone urbane. Nel database Endomondo il numero complessivo di punti GPS è di oltre 1.050.000, costituenti 9478 tracce, alcune delle quali registrate anche al di fuori del comune o della città metropolitana di Bologna. I dati sono strutturati in formato digitale .csv (*Comma separated values*), ovvero sono formati da tabelle in cui ogni riga è una linea di testo divisa in campi intervallati da un apposito carattere separatore - una virgola - ciascuno dei quali rappresenta un valore. Di seguito sono riportati una traccia estratta dal database Endomondo (Fig. 3.1) e l’insieme dei punti contenuti nel database (Fig. 3.2), ricadenti anche in zone esterne all’area di studio.

¹⁰² European Cycling Challenge, <http://www.europeancyclingchallenge.eu>

	A	B	C	D	E	F	G	H	I	J	K
1	#pointDBNode,pointPathId,id,timestamp,latitude,longitude,altitude,distance,heartRate,instruction,speed										
2	4,61565791,23648171762,2013-05-01 06:33:58,44.501085,11.372906,92.4862,0,163.572,0,2,0										
3	4,61565791,23648171763,2013-05-01 06:34:10,44.501085,11.372906,92.4862,0,163.572,0,2,0										
4	4,61565791,23648171764,2013-05-01 06:34:26,44.501342,11.372974,92.4862,0,163.572,0,2,0										
5	4,61565791,23648171765,2013-05-01 06:34:44,44.501116,11.372609,92.5245,0,163.572,0,2,0										
6	4,61565791,23648171766,2013-05-01 06:34:53,44.500938,11.372219,92.7759,0,163.572,0,2,0										
7	4,61565791,23648201371,2013-05-01 06:34:58,44.500844,11.372004,92.4862,0,163.572,0,2,0										
8	4,61565791,23648201372,2013-05-01 06:35:05,44.500707,11.371673,92.0066,0,163.572,0,2,0										
9	4,61565791,23648201373,2013-05-01 06:35:11,44.500678,11.371373,91.6051,0,163.572,0,2,0										
10	4,61565791,23648201374,2013-05-01 06:35:17,44.500715,11.371048,91.5325,0,163.572,0,2,0										
11	4,61565791,23648201375,2013-05-01 06:35:22,44.500793,11.370784,91.3983,0,163.572,0,2,0										
12	4,61565791,23648201376,2013-05-01 06:35:25,44.500843,11.370633,91.2977,0,163.572,0,2,0										
13	4,61565791,23648201377,2013-05-01 06:35:31,44.500961,11.370335,91.4709,0,163.572,0,2,0										
14	4,61565791,23648201378,2013-05-01 06:35:36,44.50104,11.370078,92.5016,0,163.572,0,2,0										
15	4,61565791,23648201379,2013-05-01 06:35:41,44.501136,11.369815,95.0831,0,163.572,0,2,0										
16	4,61565791,23648201380,2013-05-01 06:35:45,44.501217,11.369624,95.7515,0,163.572,0,2,0										
17	4,61565791,23648201381,2013-05-01 06:35:50,44.501111,11.369529,95.4842,0,163.572,0,2,0										
18	4,61565791,23648201382,2013-05-01 06:35:57,44.500925,11.369429,95.5101,0,163.572,0,2,0										
19	4,61565791,23648232030,2013-05-01 06:36:04,44.500678,11.369254,95.7432,0,163.572,0,2,0										
20	4,61565791,23648232031,2013-05-01 06:36:09,44.500472,11.369135,96.1125,0,163.572,0,2,0										
21	4,61565791,23648232032,2013-05-01 06:36:16,44.500496,11.368775,96.0374,0,163.572,0,2,0										
22	4,61565791,23648232033,2013-05-01 06:36:22,44.50061,11.368368,96.5584,0,163.572,0,2,0										
23	4,61565791,23648232034,2013-05-01 06:36:27,44.5007,11.368074,101.051,0,163.572,0,2,0										

Figura 3.1: esempio di dati GPS contenuti in una traccia del database Endomondo, in formato .csv

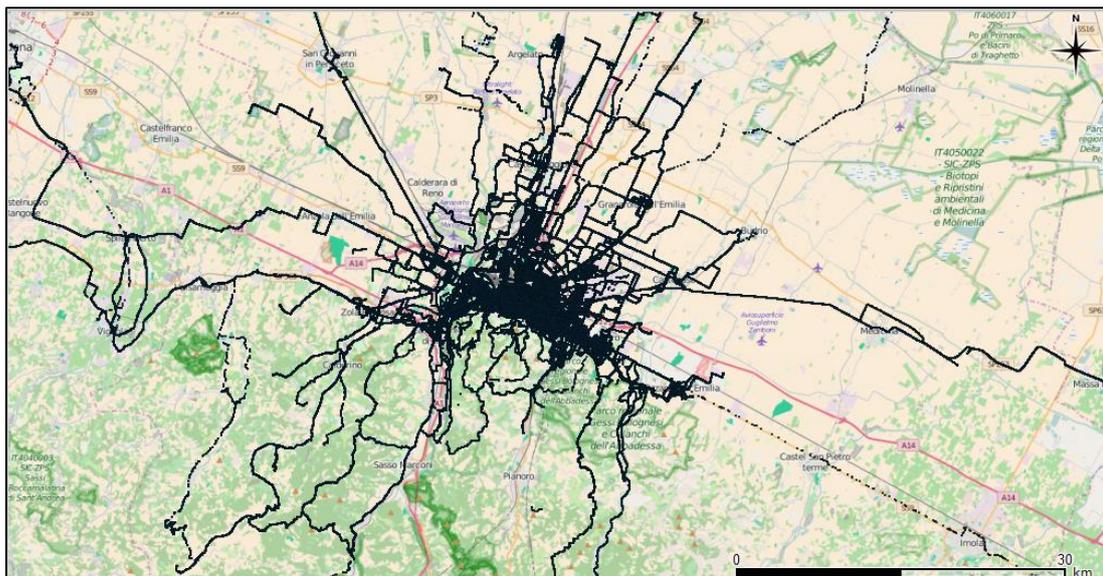


Figura 3.2: punti GPS del database Endomondo

La rete di trasporto della città di Bologna (Fig. 3.3) è stata ricavata attraverso il download dal database aperto di OpenStreetMap (OSM),¹⁰³ liberamente fruibile da tutti gli utenti; i dati sono disponibili come *shapefiles*, utilizzabili anche su piattaforma GIS (*Geographic Information System*), ovvero dati georeferenziati, comprendenti informazioni sulla geometria degli elementi e sugli attributi ad essi collegati.

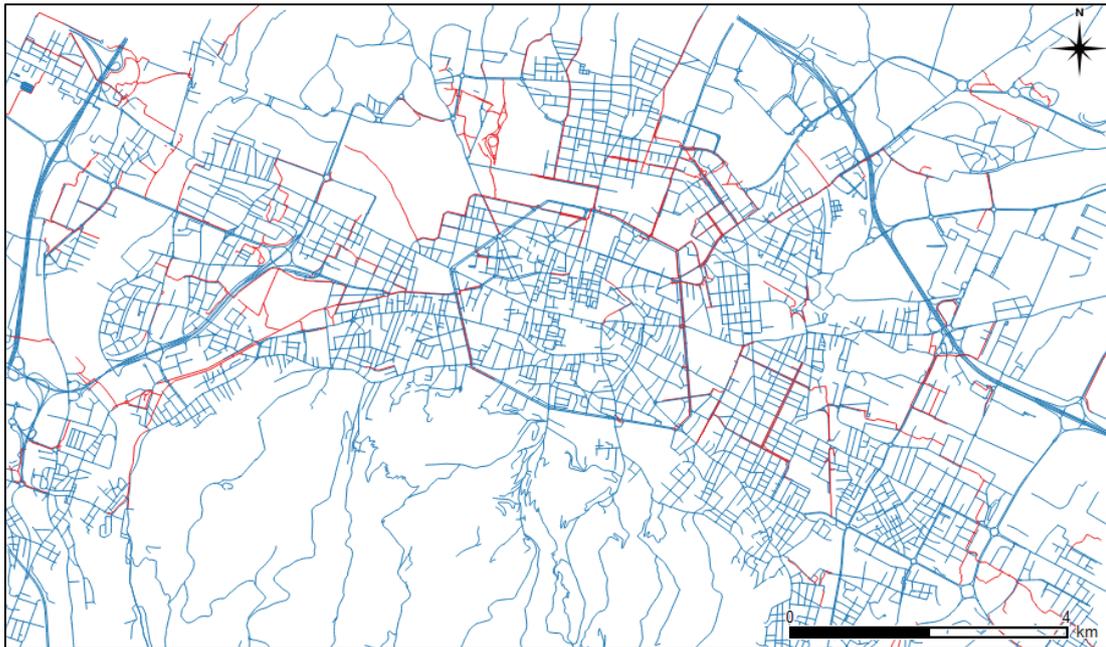


Figura 3.3: rete di Bologna (rete ciclabile in rosso)

Un secondo ambiente di testing per gli algoritmi è stato allestito nell'area urbana di Catania, dove sono state registrate alcune tracce di prova di spostamenti in bicicletta nel mese di Novembre 2015, ed utilizzata la relativa rete urbana. I dati sono stati raccolti tramite smartphone, mediante l'applicazione Runtastic Road Bike.¹⁰⁴ Tale campione di tracce è stato prodotto al fine di valutare l'accuratezza della rete e del segnale GPS sugli smartphone, e di sperimentare la risposta dei vari algoritmi anche

¹⁰³ OpenStreetMap, <http://www.openstreetmap.org>

¹⁰⁴ Runtastic Road Bike, <http://www.runtastic.com>

in situazioni complesse, come nel caso di itinerari particolarmente tortuosi e di gran lunga differenti rispetto ai corrispondenti *shortest paths*. I risultati così ottenuti possono essere confrontati con l'effettivo percorso scelto dall'utente, analisi che non può essere realizzata utilizzando il database Endomondo, nel quale il percorso effettivo è ignoto.

Anche in questo caso ogni traccia (*trkseg*) $T = \{ P_1, P_2, \dots, P_n \}$ è costituita da una successione ordinata di n punti P_i (*trackpoints*, *trkpt*) identificati tramite le coordinate geografiche longitudine e latitudine (lon_i, lat_i) e l'istante di tempo in cui il rilevamento viene effettuato (*timestamp*, $time_i$); ad ogni punto è associata anche una stima dell'altitudine (ele_i). Il numero complessivo di punti GPS contenuti nel campione raccolto è di 2028, costituenti 13 tracce. I dati sono strutturati in formato XML ed hanno estensione .gpx (*GPS eXchange Format*). Di seguito è riportata una traccia estratta dal campione (Fig. 3.4).

```

<trkseg>
  <trkpt lon="15.0818595886230469"
lat="37.5220413208007812">
    <ele>173.0</ele>
    <time>2015-11-09T11:10:43.000Z</time>
  </trkpt>
  <trkpt lon="15.0819158554077148"
lat="37.5219383239746094">
    <ele>175.0</ele>
    <time>2015-11-09T11:10:57.000Z</time>
  </trkpt>
  <trkpt lon="15.0820322036743164"
lat="37.5218353271484375">
    <ele>161.0</ele>
    <time>2015-11-09T11:11:01.000Z</time>
  </trkpt>
  <trkpt lon="15.0821647644042969"
lat="37.5217590332031250">
    <ele>156.0</ele>
    <time>2015-11-09T11:11:05.000Z</time>
  </trkpt>
  <trkpt lon="15.0822515487670898"
lat="37.5215797424316406">
    <ele>149.0</ele>
    <time>2015-11-09T11:11:09.000Z</time>
  </trkpt>

```

Figura 3.4: esempio di dati GPS contenuti in una traccia del campione raccolto, in formato .gpx

Anche la rete di trasporto della città di Catania è stata ricavata attraverso il download dal database di OpenStreetMap (Fig. 3.5).¹⁰⁵

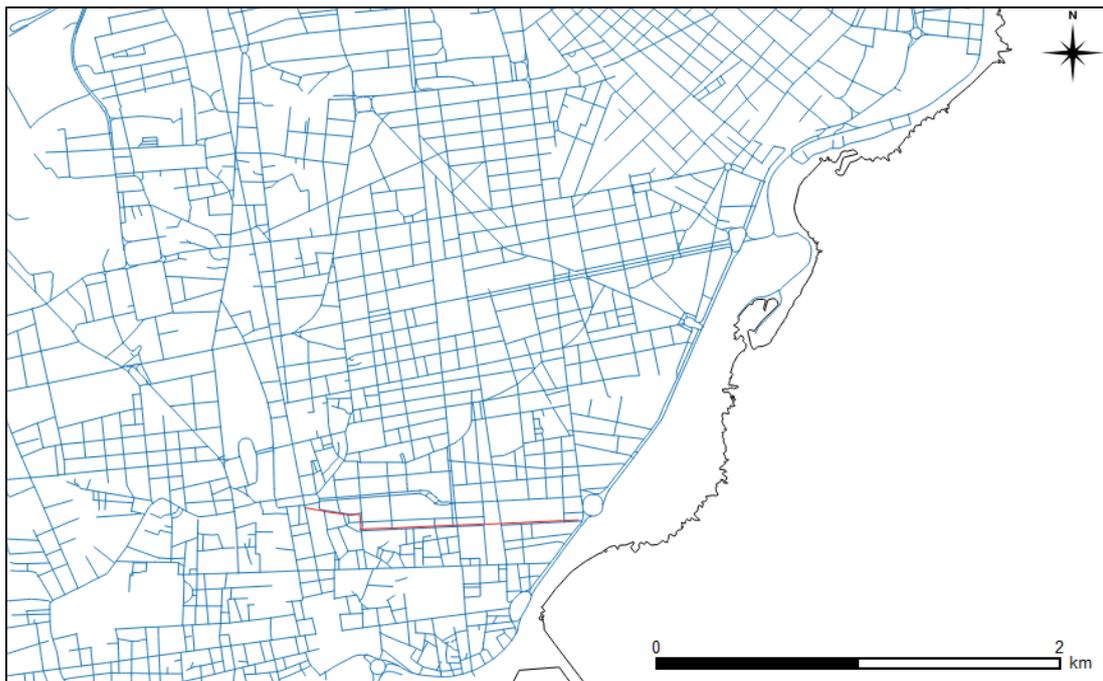


Figura 3.5: rete di Catania (centro)

Tutte le operazioni relative al map-matching sono state eseguite all'interno dell'ambiente SUMO¹⁰⁶ (*Simulation of Urban MObility*).¹⁰⁷ Esso è un pacchetto open source di micro-simulazione continua del traffico stradale progettato per la gestione di reti di grandi dimensioni, scritto in linguaggio Python¹⁰⁸ e sviluppato all'interno del Centro Aerospaziale Tedesco DLR (*Deutsches Zentrum für Luft - und Raumfahrt*) dall'Istituto dei Sistemi di Trasporto (*Institute of Transportation Systems*).

¹⁰⁵ OpenStreetMap, <http://www.openstreetmap.org>

¹⁰⁶ Behrisch M. et al., 2011.

¹⁰⁷ DLR and contributors, SUMO, <http://sumo.sourceforge.net/>

¹⁰⁸ <http://www.python.org/>

Nel presente lavoro, SUMO è stato utilizzato per convertire il database OSM in una rete stradale; per realizzare il map-matching e per una prima elaborazione dei risultati, sono stati impiegati alcuni moduli disponibili in Python: pyproj (per la proiezione dei punti GPS), shapely¹⁰⁹ (per l'analisi di oggetti geometrici nel piano cartesiano), numpy¹¹⁰ (per le operazioni matematiche più complesse), pylab (per il tracciamento dei grafici) e time (per il calcolo del tempo impiegato per l'esecuzione). Gli algoritmi sono stati scritti anch'essi in linguaggio Python tramite degli scripts.

Altri programmi utilizzati per l'elaborazione dei risultati sono Microsoft Office Excel 2013© per la parte grafica e Quantum GIS (QGIS)¹¹¹ versione 2.8.2 “Wien” per la parte su GIS.

I test sono stati eseguiti tutti su un PC a sistema operativo Windows© 7 Home Premium, con processore Intel® i7 a frequenza di clock pari a 1,70 GHz, e memoria RAM 4 GB.

2. PREPROCESSING E DATI DI OUTPUT

Nel presente lavoro sono state eseguiti complessivamente 43 test di map-matching sul database Endomondo, per un totale di 184 h e 48 minuti. I test sono stati realizzati con l'obiettivo di ottimizzare i valori dei parametri di ogni algoritmo, in modo da ottenere una buona accuratezza in un tempo di matching accettabile.

Prima di procedere al map-matching, allo scopo di filtrare le tracce degli spostamenti degli utenti (*clean*), si è resa necessaria una fase di *preprocessing*, che ne ha ridotto il numero a 5969 (Fig. 3.6). Sono state eliminate le tracce interamente al di fuori della zona della rete selezionata, quelle con durata minore di 30 secondi, quelle costituite da

¹⁰⁹ <http://toblerity.org/shapely/>

¹¹⁰ <http://www.numpy.org/>

¹¹¹ Quantum GIS (QGIS), <http://www.qgis.org/>

meno di 10 punti e quelle con una distanza percorsa minore di 300 metri, in quanto considerate non significative.

Inoltre, gli algoritmi 'Marchal' e 'NewMarchal' hanno reso necessaria l'ulteriore suddivisione delle tracce (*split*) nel caso di due punti successivi a distanza maggiore di 300 metri o con un intervallo di rilevamento maggiore di 30 secondi. Infine, sono stati divisi in più parti anche i percorsi pseudo-circolari, nel caso in cui uno dei punti della traccia si trovasse ad una distanza dal punto iniziale minore della metà della distanza massima dallo stesso. In questo caso la traccia è stata divisa in due parti, in corrispondenza del punto della traccia originale a massima distanza dal punto iniziale.

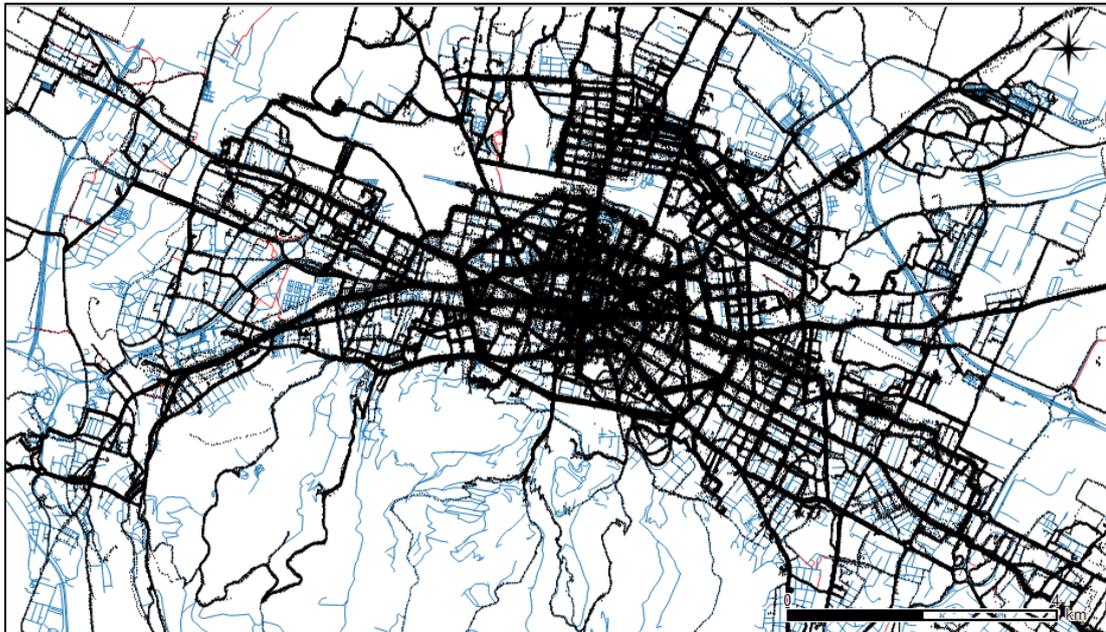


Figura 3.6: punti GPS del database Endomondo sulla rete di Bologna

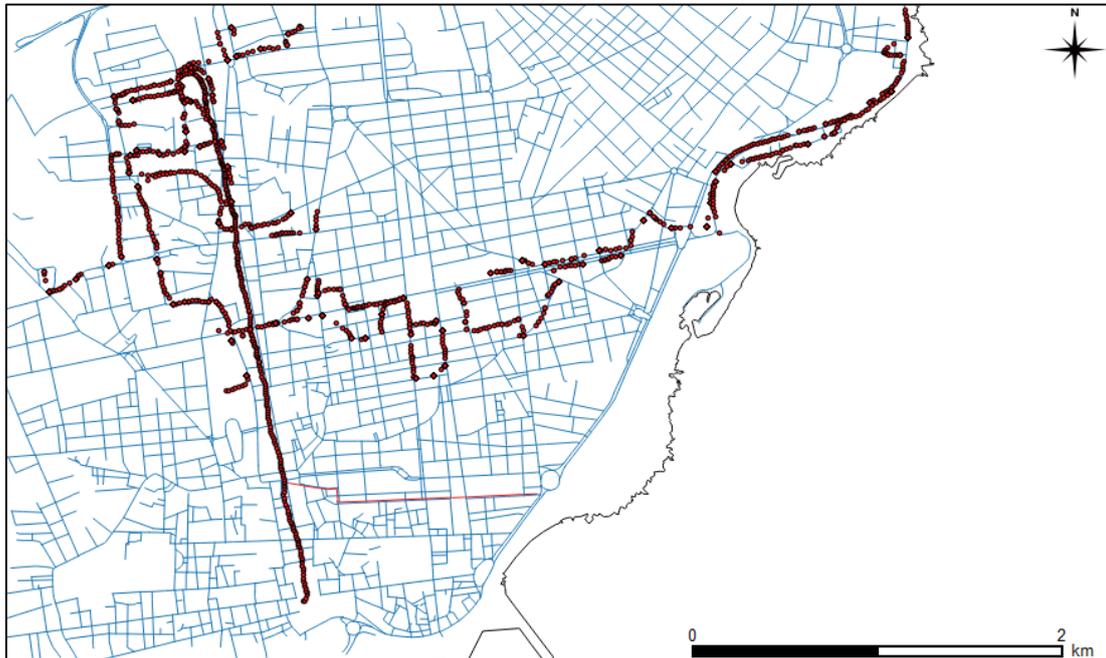
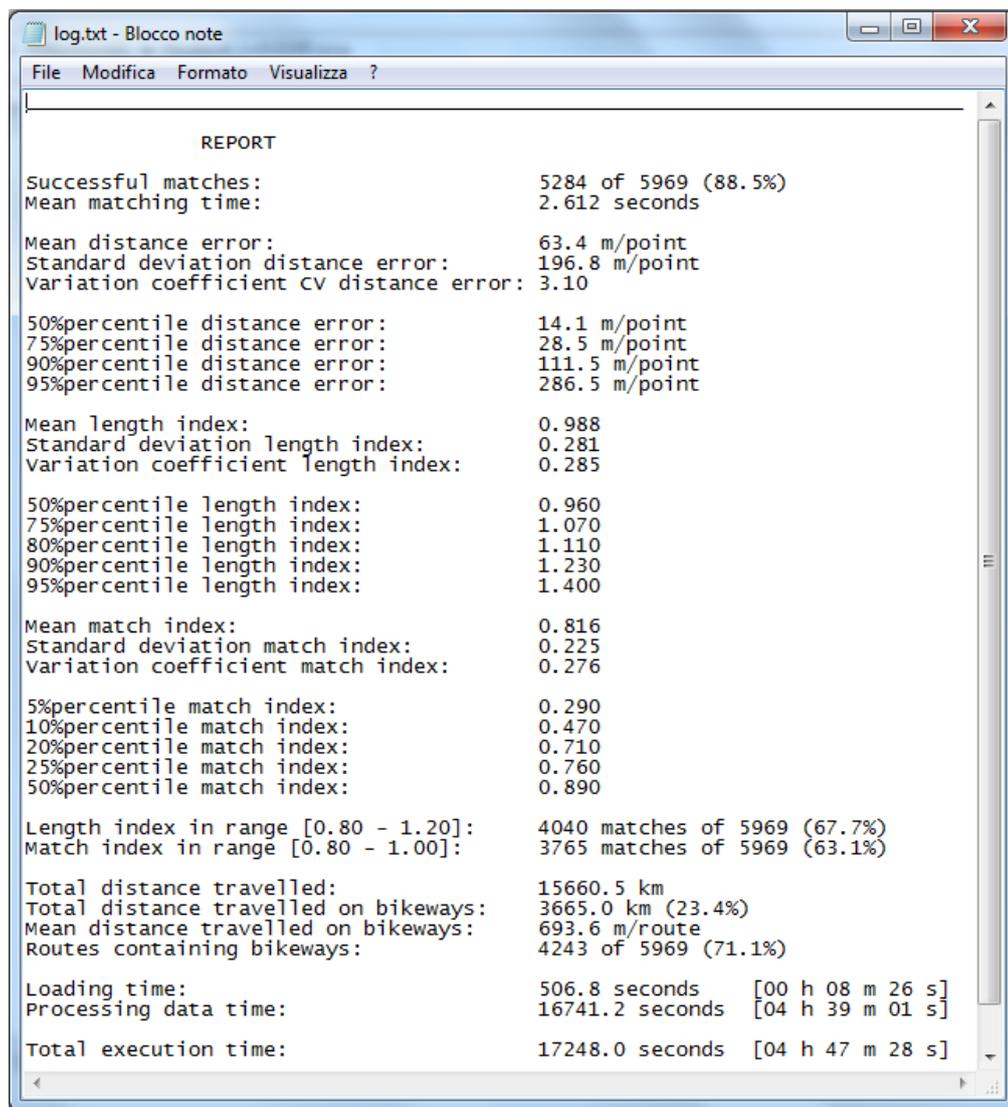


Figura 3.7: punti GPS del campione raccolto sulla rete di Catania

Per il salvataggio dei dati di output è stato adoperato uno script in linguaggio Python; al termine di ogni prova sono stati redatti un report in un file di testo (.txt) ed alcuni grafici per la valutazione dell'accuratezza del matching, e i punti GPS e gli itinerari risultanti sono stati salvati in formato *shapefiles*, rendendoli così esportabili e manipolabili su GIS.

L'analisi dei parametri più significativi di ogni algoritmo e la comparazione tra i differenti algoritmi sono state effettuate valutando i risultati di ogni test attraverso il numero di matches riusciti (*Successful matches*), il tempo medio impiegato per il matching di ciascuna traccia (*Mean matching time*) ed un'analisi statistica sia su alcuni indicatori che saranno descritti in seguito (*Length index* e *Match index*), sia sull'errore di distanza (*Distance error*), ovvero la distanza media misurata in [metri/punto] tra i punti GPS tracciati e gli archi costituenti il percorso identificato. Si è fatto riferimento in particolare alla frequenza, alla frequenza cumulata e ai percentili, ovvero i valori non superati in una certa percentuale dei casi, poiché considerati statisticamente più significativi di semplici medie, data la vastità del campione disponibile.

In occasione di ogni prova sono stati altresì calcolati la distanza totale percorsa (*Total distance travelled*), la distanza totale (*Total distance travelled on bikeways*) e media (*Mean distance travelled on bikeways*) percorsa su piste ciclabili, il numero di itinerari comprendenti piste ciclabili (*Routes containing bikeways*) ed il tempo di caricamento del database e della rete (*Loading time*). Di seguito è riportato un esempio del report generato al termine di ogni test (Fig. 3.8).



```

log.txt - Blocco note
File Modifica Formato Visualizza ?

REPORT

Successful matches:                5284 of 5969 (88.5%)
Mean matching time:                2.612 seconds

Mean distance error:               63.4 m/point
Standard deviation distance error: 196.8 m/point
Variation coefficient CV distance error: 3.10

50%percentile distance error:      14.1 m/point
75%percentile distance error:      28.5 m/point
90%percentile distance error:      111.5 m/point
95%percentile distance error:      286.5 m/point

Mean length index:                 0.988
Standard deviation length index:    0.281
Variation coefficient length index: 0.285

50%percentile length index:         0.960
75%percentile length index:         1.070
80%percentile length index:         1.110
90%percentile length index:         1.230
95%percentile length index:         1.400

Mean match index:                  0.816
Standard deviation match index:     0.225
Variation coefficient match index:  0.276

5%percentile match index:          0.290
10%percentile match index:         0.470
20%percentile match index:         0.710
25%percentile match index:         0.760
50%percentile match index:         0.890

Length index in range [0.80 - 1.20]: 4040 matches of 5969 (67.7%)
Match index in range [0.80 - 1.00]:  3765 matches of 5969 (63.1%)

Total distance travelled:           15660.5 km
Total distance travelled on bikeways: 3665.0 km (23.4%)
Mean distance travelled on bikeways: 693.6 m/route
Routes containing bikeways:         4243 of 5969 (71.1%)

Loading time:                       506.8 seconds   [00 h 08 m 26 s]
Processing data time:                16741.2 seconds [04 h 39 m 01 s]

Total execution time:                17248.0 seconds [04 h 47 m 28 s]

```

Figura 3.8: esempio di report al termine di un test

3. PRIMA METODOLOGIA: ALGORITMO ‘orig’

La prima metodologia di map-matching analizzata trae spunto dal lavoro di J.Schweizer e F.Rupi (2014).¹¹² Il primo algoritmo studiato è quello da loro ideato ed è classificabile come di tipo probabilistico. Esso può essere diviso in due passaggi essenziali, eseguiti per ogni traccia, dopo il *clean* iniziale:

- 1) Assegnazione di un costo o peso (*weight*) a tutti gli archi della rete: attorno ad ogni arco (*edge*) è creato un *buffer* di 30 metri ed è quindi rilevato il numero di punti GPS ricadenti all’interno di ogni buffer; queste operazioni sono le più gravose dal punto di vista computazionale. Se un punto GPS j si trova in m_j buffers diversi, il suo contributo al costo di ogni arco è diviso in parti uguali ed assunto pari a $\frac{1}{m_j}$. Il costo c_a di ogni arco a è quindi dato dalla formula seguente:

$$c_a = c \cdot L_a - \sum_{j \in P_a} \frac{1}{m_j} \quad (3.1)$$

con L_a lunghezza dell’arco a , e P_a insieme dei punti j ricadenti nel buffer dell’arco a . La costante c è assunta pari a 0,01 per gli archi stradali (c_{length}), e pari a 0,005 per gli archi di piste ciclabili (c_{bike}); in tal modo, si ipotizza che, a parità di lunghezza, i ciclisti preferiscano una pista ciclabile ad una strada normale. La funzione di costo è quindi data dalla somma algebrica di due termini: il primo, penalizzante, dipende dalla lunghezza dell’arco e ne aumenta il costo; il secondo - da sottrarre al primo - ne diminuisce invece il costo all’aumentare del numero di punti GPS ricadenti nel buffer dell’arco stesso.

- 2) Dati i costi calcolati al punto precedente, determinazione del percorso di minimo costo, per ogni coppia origine - destinazione, tramite l’algoritmo di Dijkstra. Come possibili coppie origine - destinazione sono presi in considerazione tutti gli archi il cui buffer contiene il primo punto GPS (arco di

¹¹² Schweizer J., Rupi F., 2014.

origine) e tutti gli archi il cui buffer contiene l'ultimo punto GPS (arco di destinazione); l'itinerario a costo totale minore tra questi è quello selezionato come *matched route*.

Tale approccio si fonda sulla supposizione secondo la quale l'itinerario dal costo totale minimo, che è quello selezionato, tende ad essere quello i cui buffers degli archi che lo costituiscono contengono il maggior numero di punti, i quali riducono i costi degli archi interessati. Difatti, si ritiene più probabile che, tra tutti i possibili percorsi, l'itinerario reale sia quello con più punti nelle vicinanze. In assenza di punti, invece, il secondo termine della funzione di costo dell'arco si annulla e di conseguenza il percorso seguirà lo *shortest path*, ovvero il percorso a lunghezza minore.

Nelle figure seguenti è esemplificato il procedimento dell'algoritmo (Fig. 3.9).

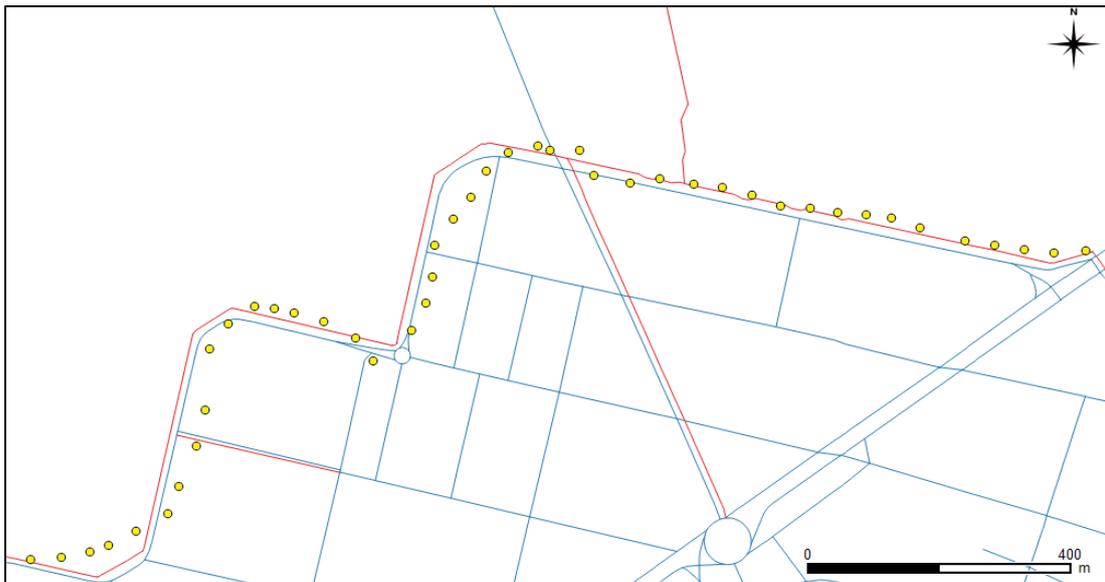


Figura 3.9 (a): esempio di map-matching: i punti GPS tracciati in giallo rappresentano un ciclista in movimento, in figura da sinistra verso destra; rete stradale in blu; rete ciclabile in rosso

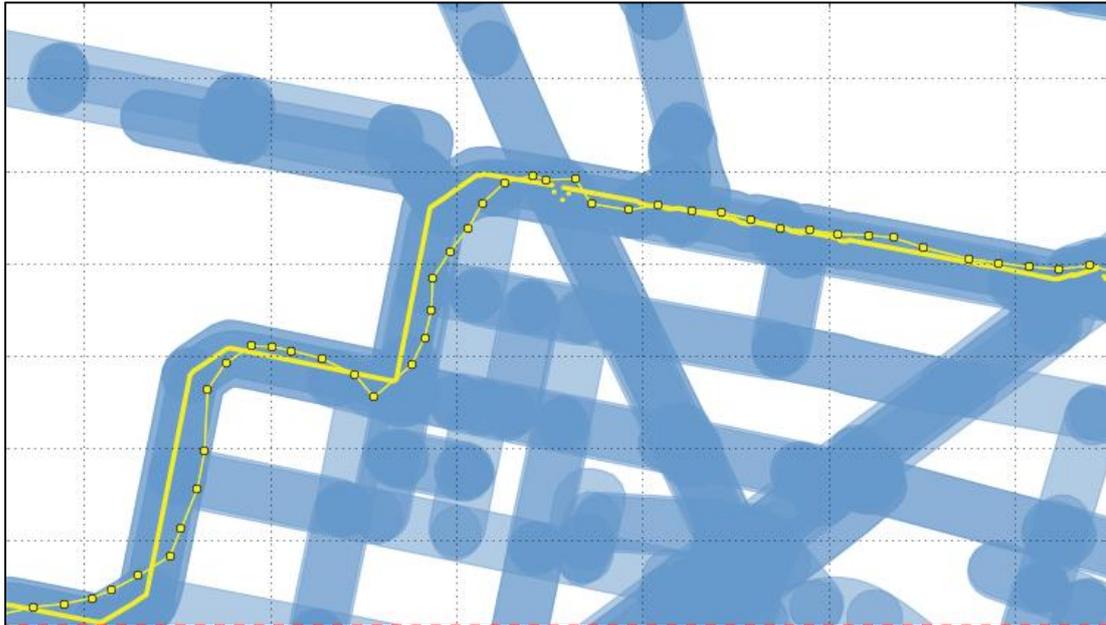


Figura 3.9 (b): creazione dei buffers per ogni arco di rete, in blu; essi in parte si sovrappongono, dove il blu è più scuro; in giallo, invece, l'itinerario ricostruito dall'algoritmo; si può notare come alla strada è preferita la pista ciclabile, grazie al minore valore attribuito alla costante C_{bike}

3.1 INDICI DI VALUTAZIONE

Al fine di valutare l'accuratezza del map-matching, ovvero la corrispondenza tra itinerario reale ed itinerario identificato dall'algoritmo, sono stati presi in considerazione due indici differenti. Tali indici, però, seppur significativi, sono in grado di dare soltanto un'indicazione di massima, poiché l'itinerario reale di ogni traccia del database Endomondo è sconosciuto e gli indici stessi dipendono dagli errori di rilevamento del GPS e della mappa.

Il primo indice utilizzato è il *Length Index*, definito come segue:

$$LengthIndex = \frac{L_{match}}{L_{GPS}} \quad (3.2)$$

Dove L_{match} rappresenta la lunghezza dell'itinerario ricostruito, mentre L_{GPS} rappresenta la lunghezza del percorso GPS, ovvero la somma di ogni singola distanza tra un punto GPS ed il successivo della traccia.

Tale indice varia teoricamente da 0 ad un valore $+\infty$. Se il *Length Index* assume valori nell'intorno dell'unità, le due lunghezze messe a rapporto tendono ad eguagliarsi, il che generalmente indica un matching abbastanza corretto - nel presente studio sono stati considerati accettabili i valori compresi tra 0,8 e 1,2. Se invece il *Length Index* assume valori minori di 0,8 o maggiori di 1,2, il percorso individuato è con tutta probabilità rispettivamente troppo breve o troppo lungo rispetto a quello reale. In entrambi i casi pertanto l'itinerario identificato non esprime con adeguata approssimazione il percorso reale.

Il secondo indice utilizzato è il *Match Index*, definito nel modo seguente:

$$MatchIndex = \frac{N_{buffer}}{N_{GPS}} \quad (3.3)$$

Dove N_{buffer} rappresenta il numero dei punti GPS ricadenti all'interno dei buffers degli archi dell'itinerario ricostruito, mentre N_{GPS} rappresenta il numero totale di punti GPS della traccia. Tale indice varia pertanto teoricamente da un minimo di 0 ad un massimo di 1. Se il *Match Index* assume valori vicino all'unità, la quasi totalità dei punti GPS ricade all'interno dei buffers degli archi del percorso identificato, il che generalmente indica un matching corretto - nel presente studio sono stati considerati accettabili i valori compresi tra 0,8 e 1. Se invece il *Match Index* assume valori bassi, minori di 0,8, più del 20% dei punti GPS tracciati si trova ad oltre 30 metri (valore della larghezza del buffer) di distanza dagli archi del percorso individuato, il che suggerisce un matching errato.

In tutte le analisi parametriche effettuate sono stati comunque tenuti in considerazione tutti gli itinerari ricostruiti, anche se i valori degli indici non erano accettabili; soltanto successivamente è stato eseguito il map-matching eliminando i risultati considerati errati (paragrafo 10).

3.2 ANALISI DEI PARAMETRI

Al fine di individuare il valore del parametro c_{length} ottimale sono stati eseguiti tre test sul database Endomondo, utilizzando tre diversi valori, rispettivamente pari a 0,001, 0,01 e 0,1; in ogni test la costante c_{bike} è stata fissata pari alla metà del valore assunto dalla c_{length} , quindi rispettivamente 0,0005, 0,005 e 0,05. Di seguito è riportata una tabella riassuntiva dei risultati ottenuti.

	c_length('orig')	0,001	c_length('orig')	0,01	c_length('orig')	0,1
Successful matches	5263	88,2%	5284	88,5%	3291	55,1%
Mean matching time [seconds]	2,728 s		2,612 s		2,592 s	
Mean Distance Error	56,9 m/point		63,4 m/point		89,8 m/point	
Standard deviation distance error	186,0 m/point		196,8 m/point		119,8 m/point	
CV distance error	3,27		3,10		1,33	
50%percentile distance error	13,6 m/point		14,1 m/point		41,1 m/point	
75%percentile distance error	25,3 m/point		28,5 m/point		120,1 m/point	
90%percentile distance error	89,5 m/point		111,5 m/point		228,7 m/point	
95%percentile distance error	246,4 m/point		286,5 m/point		329,4 m/point	
Mean Length Index	1,410		0,988		0,999	
Standard deviation length index	0,484		0,281		0,249	
CV length index	0,343		0,285		0,250	
50%percentile length index	1,330		0,960		0,930	
75%percentile length index	1,690		1,070		1,020	
80%percentile length index	1,790		1,110		1,060	
90%percentile length index	2,090		1,230		1,220	
95%percentile length index	2,310		1,400		1,390	
Mean Match Index	0,824		0,816		0,652	
Standard deviation match index	0,212		0,225		0,276	
CV match index	0,257		0,276		0,424	
5%percentile match index	0,310		0,290		0,140	
10%percentile match index	0,520		0,470		0,250	
20%percentile match index	0,730		0,710		0,380	
25%percentile match index	0,780		0,760		0,450	
50%percentile match index	0,890		0,890		0,690	
Length index in range [0,80 - 1,20]	1685	28,2%	4040	67,7%	2948	49,4%
Match index in range [0,80 - 1,00]	3838	64,3%	3765	63,1%	1216	20,4%
Total distance travelled [km]	23024,3 km		15660,5 km		9087,2 km	
Total distance travelled on bikeways [km]	4515,3	19,6%	3665,0	23,4%	3383,1	37,2%
Mean distance travelled on bikeways	857,9 m/route		693,6 m/route		1028,0 m/route	
Routes containing bikeways	4301	72,1%	4243	71,1%	2706	45,3%
Loading time [seconds]	406,3 s		506,8 s		408,2 s	
Processing data time [seconds]	17590,6 s		16741,2 s		16560,1 s	
Total execution time [seconds]	17996,9 s		17248,0 s		16968,4 s	

Tabella 3.1: risultati dei test sul database Endomondo, algoritmo 'orig', al variare di c_{length}

Dalla tabella riassuntiva appare chiaro che la scelta di c_{length} pari 0,01 è quella più corretta, confermando quanto affermato dagli autori: difatti, assumendo questo valore, si ottiene un numero di itinerari, i cui indici ricadono in un intervallo accettabile (*Length index in range [0,80 – 1,20]* e *Match index in range [0,80 – 1,00]*), decisamente superiore ai risultati delle altre due prove (Fig. 3.10). Anche i valori dei percentili degli indici confermano tale affermazione.

Il tempo di matching (*mean matching time*), invece, si mantiene pressoché costante, assumendo un valore decisamente alto, attorno a 2,6 - 2,7 secondi/traccia, e l'intero processo di map-matching impiega quasi 5 ore.

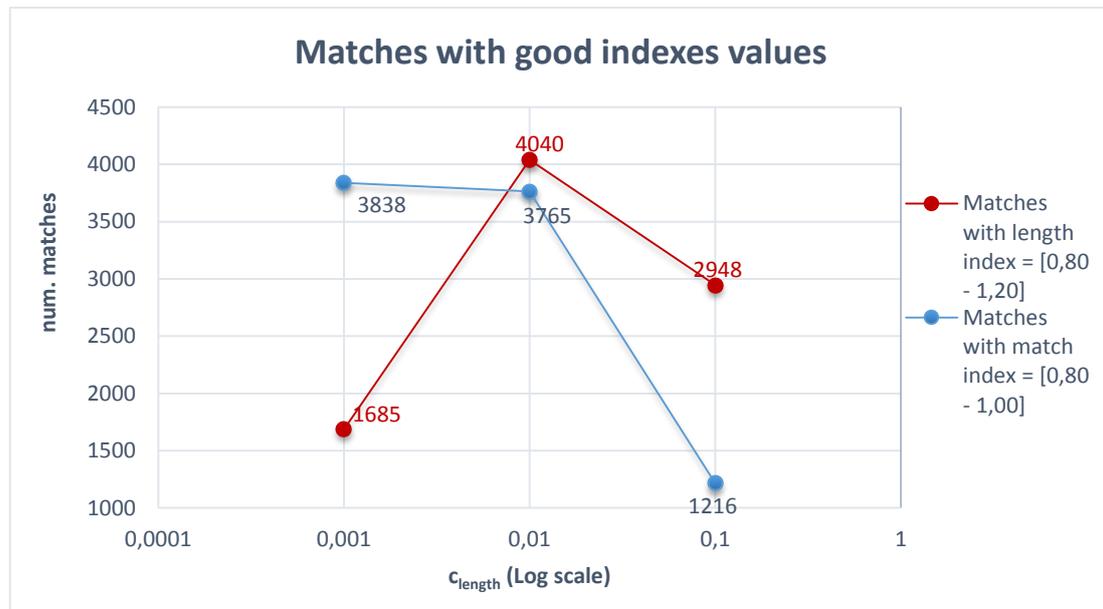


Figura 3.10: numero di matches aventi buoni valori degli indici, al variare di c_{length} (rappresentata in scala logaritmica)

Risulta comunque evidente la forte dipendenza dell'algoritmo dal parametro c_{length} , dato che i risultati appaiono alquanto imprecisi sia per un valore del parametro pari a 0,001 (itinerari spesso decisamente troppo lunghi, come appare evidente dal *Length Index*), sia per un valore pari 0,1 (in tal modo la lunghezza dell'arco diventa

preponderante nel calcolo del costo del medesimo, rispetto al numero di punti GPS ricadenti nel suo buffer, e l'algoritmo di Dijkstra tende quindi verso il cammino più breve tenendo in scarsa considerazione i punti).

3.3 RISULTATI E CONSIDERAZIONI

Di seguito è riportata una tabella riassuntiva dei risultati ottenuti eseguendo l'algoritmo di map-matching denominato 'orig' sul database di tracce Endomondo, posto il parametro c_{length} pari a 0,01.

	'orig'	
Successful matches	5284	88,5%
Mean matching time [seconds]	2,612 s	
Mean Distance Error	63,4 m/point	
Standard deviation distance error	196,8 m/point	
CV distance error	3,10	
50%percentile distance error	14,1 m/point	
75%percentile distance error	28,5 m/point	
90%percentile distance error	111,5 m/point	
95%percentile distance error	286,5 m/point	
Mean Length Index	0,988	
Standard deviation length index	0,281	
CV length index	0,285	
50%percentile length index	0,960	
75%percentile length index	1,070	
80%percentile length index	1,110	
90%percentile length index	1,230	
95%percentile length index	1,400	
Mean Match Index	0,816	
Standard deviation match index	0,225	
CV match index	0,276	
5%percentile match index	0,290	
10%percentile match index	0,470	
20%percentile match index	0,710	
25%percentile match index	0,760	
50%percentile match index	0,890	
Length index in range [0,80 - 1,20]	4040	67,7%
Match index in range [0,80 - 1,00]	3765	63,1%
Total distance travelled [km]	15660,5 km	
Total distance travelled on bikeways [km]	3665,0	23,4%
Mean distance travelled on bikeways	693,6 m/route	
Routes containing bikeways	4243	71,1%
Loading time [seconds]	506,8 s	
Processing data time [seconds]	16741,2 s	
Total execution time [seconds]	17248,0 s	

Tabella 3.2: risultati del map-matching dell'algoritmo 'orig' sul database Endomondo

L'algoritmo 'orig', per la sua struttura intrinseca, associa sempre un itinerario ad ogni traccia a cui riesce a collegare un arco di partenza ed uno di arrivo, grazie all'azione dell'algoritmo di Dijkstra al suo interno, il quale ricerca il percorso più breve sulla rete, anche in tratti in assenza di punti GPS, qualunque siano i costi. Difatti, ben l'88,5% di matches sono andati a buon fine (*successful matches*); le tracce per cui non è stato possibile identificare un itinerario sono quelle che si trovano interamente al di fuori della rete considerata, o quelle per le quali non è possibile collegare gli eventuali archi di partenza e di arrivo individuati. Tuttavia, occorre analizzare attentamente i risultati: errori molto grandi possono derivare dalle anomalie della rete, se ad esempio i punti GPS passano per due archi rappresentati non connessi sulla rete, anche se in realtà essi lo sono. In questo caso, l'algoritmo di Dijkstra cercherà comunque un percorso attorno a tali archi, seguendo la più breve alternativa sulla rete, spesso molto più lunga dell'itinerario reale e contenente archi non facenti parte del percorso reale. Di conseguenza, questi *false routing* presenteranno un *Length Index* molto superiore a 1 ed un *Match Index* molto inferiore a 1. Un'altra fonte di notevoli imprecisioni può scaturire dall'uso da parte dei ciclisti di strade pedonali o di parchi non riprodotti sulla rete: anche in questo caso l'algoritmo di Dijkstra cercherà il percorso più breve alternativo, ma poco rappresentativo (Fig. 3.11).

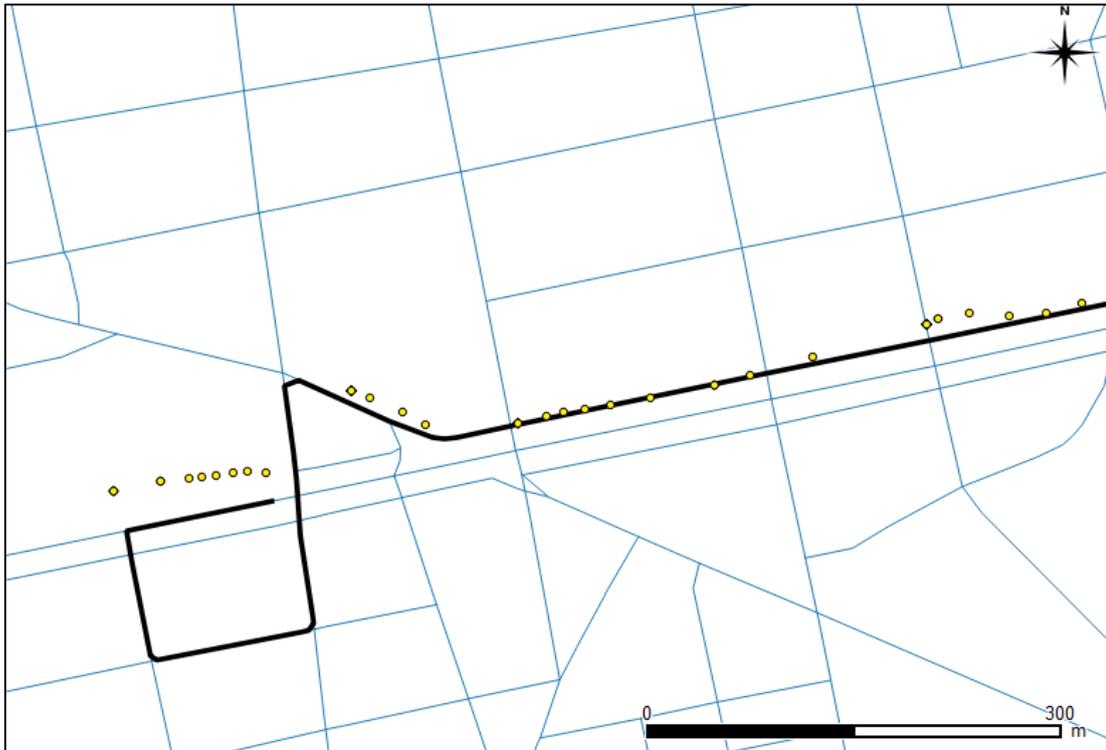


Figura 3.11: esempio di map-matching effettuato con l'algoritmo 'orig', tratto dal campione raccolto; in questo caso un arco non è riportato sulla rete e, tramite l'algoritmo di Dijkstra è ricercato il percorso alternativo a minima lunghezza

L'errore di distanza (*distance error*) ha una media molto alta (circa 63 m/punto) ed è molto distribuito (coefficiente di variazione di 3,1); il percentile al 50% però evidenzia che circa la metà dei matches presentano un errore (14,1 m/punto) paragonabile agli errori di misurazione, per cui ampiamente accettabili.

Anche il *Length Index* mostra dei valori ottimi, molto concentrati nell'intorno di 1 (media di 0,988, e coefficiente di variazione di appena 0,285): poco più dei $\frac{2}{3}$ dei matches presentano un indice compreso in un range accettabile tra 0,8 e 1,2.

Infine anche il *Match Index* offre dei buoni risultati: una media non lontana dall'unità (0,816) ed una distribuzione non troppo dispersa (coefficiente di variazione di 0,276), con il 63,1% di matches aventi un indice maggiore di 0,8.

Il principale difetto di quest'algoritmo è il notevole costo computazionale: ben 2,6 secondi è il tempo medio impiegato per il matching di ogni traccia (*mean matching time*). In totale l'intero processo di map-matching ha richiesto 4 ore, 47 minuti e 28 secondi (*total execution time*), un lasso di tempo poco accettabile, dovuto per la maggior parte alla fase di assegnazione dei costi a tutti gli archi della rete, da effettuare per ogni traccia; di conseguenza, una rete contenente un elevato numero di archi penalizza pesantemente la velocità di esecuzione.

Nella figura seguente sono raffigurati tutti gli itinerari risultanti (Fig. 3.12).

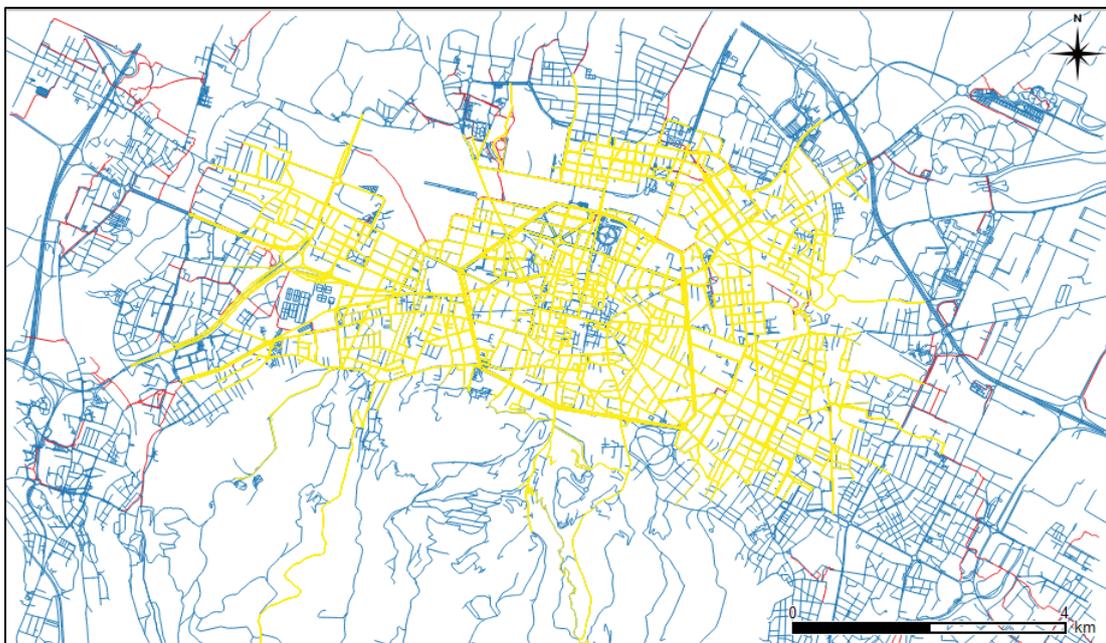


Figura 3.12: itinerari risultanti dal map-matching effettuato dall'algoritmo 'orig', in giallo; rete stradale in blu, rete ciclabile in rosso

Di seguito sono riportati i risultati del map-matching effettuato mediante l'algoritmo 'orig' sul campione di tracce raccolte.

ID_ROUTE	REAL LEN [m]	'orig'					
		LEN_ROUTE [m]	REAL LENGTH INDEX	MATCHIND	LENGTHIND	TIME_MATCH [s]	ERR_DIST [m]
11	992,420	789,800	0,80	0,88	0,77	0,574	20,7
10	2709,370	3108,120	1,15	0,94	1,09	1,246	9,9
12	2084,170	2168,880	1,04	0,98	0,95	1,460	15,1
7#	1128,730	1235,340	1,09	1,00	1,02	0,555	6,8
1	525,350	574,550	1,09	1,00	1,06	1,038	4,9
0	956,870	967,860	1,01	1,00	1,11	1,160	4,9
3	1297,040	1252,910	0,97	0,98	0,99	2,827	7,4
2	1596,590	1824,510	1,14	0,85	1,13	0,759	12,6
5	1266,520	1336,520	1,06	1,00	0,94	1,032	5,9
4	1514,570	1785,150	1,18	0,98	1,07	0,773	6,9
6	735,970	792,730	1,08	1,00	1,13	1,351	7,3
9	1101,030	1180,830	1,07	1,00	1,06	3,811	6,8
8	1064,700	1133,230	1,06	1,00	0,97	3,953	8,0
mean		1396,19	1,06	0,97	1,02	1,580	9,0

Tabella 3.3: risultati del matching dell'algoritmo 'orig' sul campione di tracce raccolte

Per le tracce del campione è presentato un altro indice, il *Real Length Index*, definito come segue:

$$RealLengthIndex = \frac{L_{match}}{L_{real}} \quad (3.4)$$

Dove L_{match} rappresenta la lunghezza dell'itinerario ricostruito, mentre L_{real} rappresenta la lunghezza dell'itinerario reale, noto per le tracce del campione. Tale indice rappresenta una versione raffinata del *Length Index* e ne assume un significato molto simile, con l'unica differenza di riferirsi al percorso reale piuttosto che ai punti GPS.

I risultati sovraesposti sono soddisfacenti ad eccezione della traccia 11 (*Length Index* = 0,77, *Real Length Index* = 0,80, valori non accettabili, come sopra affermato) e sostanzialmente confermano quanto osservato in precedenza per il database Endomondo.

4. PRIMA METODOLOGIA: ALGORITMO PROPOSTO 'new'

Il secondo algoritmo studiato, denominato 'new', si propone come possibile aggiornamento di quello studiato in precedenza; infatti, esso utilizza la medesima metodologia, seppur con degli accorgimenti. Gli scopi principali delle modifiche apportate sono di eliminare in buona parte la dipendenza dal parametro c_{length} e di ridurre il tempo di esecuzione, problematiche evidenziate nel paragrafo 3.3. Il funzionamento è del tutto analogo a quello dell'algoritmo 'orig', ad eccezione del calcolo del costo per ogni arco, fase più impegnativa dal punto di vista del costo computazionale. Dopo il *clean* dei dati, anch'esso può essere diviso in due momenti principali, eseguiti per ogni traccia:

- 1) Assegnazione di un costo o peso (*weight*) a tutti gli archi della rete: attorno ad ogni arco (*edge*) è creato un *buffer*, la cui larghezza ottimale è stata determinata ed è quindi rilevato il numero di punti GPS ricadenti all'interno di ogni buffer. Queste operazioni sono le più gravose dal punto di vista computazionale. Se un punto GPS j si trova in m_j buffers diversi, il suo contributo al costo di ogni arco è diviso in parti uguali ed assunto pari a $\frac{1}{m_j}$. Il costo c_a di ogni arco a è quindi valutato in maniera differente a seconda della presenza o meno di punti GPS all'interno del suo buffer, ed è dato dalle formule seguenti:

$$\begin{cases} c_a = \frac{c_{bike} \cdot L_a}{\sum_{j \in P_a} \frac{1}{m_j}} & \text{if } j \in \text{buffer}_a \\ c_a = \frac{c_{bike} \cdot L_a}{c_{length} \cdot L_a} & \text{elsewhere} \end{cases} \quad (3.5)$$

con L_a lunghezza dell'arco a , P_a insieme dei punti ricadenti nel buffer dell'arco a , ed i parametri c_{length} , il principale, e c_{bike} assunto minore di 1 nel caso l'arco sia ad uso esclusivamente ciclabile, 1 per gli altri archi. In tal modo si ipotizza che, a parità di lunghezza, i ciclisti preferiscano una pista ciclabile ad una strada normale.

La funzione di costo è quindi diversa in base alla presenza di punti GPS all'interno del buffer dell'arco. Nel caso ve ne ricadano, essi ne diminuiscono il costo, il quale rimane comunque direttamente proporzionale alla lunghezza dell'arco. Inoltre, l'effetto della sommatoria, che nella presente funzione di costo si colloca al denominatore, è maggiore di quello dell'algoritmo 'orig', ove la sommatoria opera una semplice differenza al termine dipendente dalla lunghezza dell'arco. Nel caso in cui non vi siano punti GPS all'interno del buffer dell'arco, invece, il costo dello stesso assume un valore dipendente dalla c_{length} ; in tal modo si cerca di evitare che l'algoritmo di Dijkstra, applicato successivamente, possa allontanarsi dalle zone dove sono stati tracciati i punti GPS.

Come già sottolineato con riguardo al primo algoritmo (paragrafo 3), si ritiene difatti più probabile che, tra tutti i possibili percorsi, l'itinerario reale sia quello con più punti nelle vicinanze.

- 2) Dati i costi calcolati al punto precedente, determinazione del percorso di minimo costo, per ogni coppia origine - destinazione, tramite l'algoritmo di Dijkstra. Come possibili coppie origine - destinazione sono presi in considerazione tutti gli archi il cui buffer contiene il primo punto GPS (arco di origine) e tutti gli archi il cui buffer contiene l'ultimo punto GPS (arco di destinazione); l'itinerario a costo totale minore tra questi è quello selezionato come *matched route*.

4.1 ANALISI DEI PARAMETRI

Al fine di individuare il valore ottimale dei parametri dell'algoritmo sono stati eseguiti diversi test, i cui risultati sono stati confrontati tramite i valori degli indici *Length Index* e *Match Index*, degli errori di distanza (*error distance*), del numero di matches e del tempo impiegato. I tre parametri analizzati sono c_{length} , c_{bike} ed il valore di larghezza del buffer $width_{buffer}$.

Per lo studio del parametro c_{length} , sicuramente il più importante ed influente, sono stati eseguiti otto test sul database Endomondo, utilizzando otto diversi valori, compresi tra 0,00001 e 0,02. Di seguito è riportata una tabella riassuntiva dei risultati ottenuti.

	c_length	0,00001	c_length	0,0001	c_length	0,0005	c_length	0,001	c_length	0,005	c_length	0,01	c_length	0,015	c_length	0,02
Successful matches	5266	88,2%	5266	88,2%	5264	88,2%	5261	88,1%	5156	86,4%	4949	82,9%	4792	80,3%	4573	76,6%
Mean matching time [seconds]	2,084 s		2,130 s		2,201 s		2,072 s		2,126 s		2,135 s		2,203 s		2,167 s	
Mean Distance Error	73,4 m/point		73,5 m/point		74,8 m/point		79,9 m/point		116,6 m/point		176,2 m/point		216,6 m/point		246,8 m/point	
Standard deviation distance error	212,5 m/point		213,1 m/point		214,4 m/point		222,4 m/point		241,8 m/point		259,3 m/point		263,4 m/point		264,2 m/point	
CV distance error	2,90		2,90		2,87		2,78		2,07		1,47		1,22		1,07	
50%percentile distance error	16,2 m/point		16,2 m/point		16,3 m/point		17,0 m/point		29,9 m/point		84,5 m/point		146,7 m/point		193,2 m/point	
75%percentile distance error	36,0 m/point		36,1 m/point		37,3 m/point		40,4 m/point		103,1 m/point		247,2 m/point		303,8 m/point		334,4 m/point	
90%percentile distance error	148,9 m/point		149,3 m/point		152,0 m/point		172,2 m/point		323,9 m/point		437,0 m/point		487,8 m/point		523,0 m/point	
95%percentile distance error	353,5 m/point		353,5 m/point		362,0 m/point		433,8 m/point		555,2 m/point		633,1 m/point		651,3 m/point		657,9 m/point	
Mean Length Index	0,964		0,963		0,963		0,956		0,947		0,968		1,003		1,037	
Standard deviation length index	0,301		0,301		0,303		0,305		0,320		0,335		0,360		0,379	
CV length index	0,312		0,312		0,315		0,319		0,338		0,347		0,359		0,365	
50%percentile length index	0,950		0,950		0,940		0,940		0,940		0,950		0,980		1,010	
75%percentile length index	1,060		1,060		1,060		1,050		1,050		1,090		1,160		1,220	
80%percentile length index	1,100		1,100		1,100		1,090		1,080		1,150		1,220		1,270	
90%percentile length index	1,230		1,230		1,250		1,240		1,250		1,310		1,399		1,450	
95%percentile length index	1,380		1,380		1,390		1,390		1,400		1,486		1,580		1,680	
Mean Match Index	0,787		0,787		0,784		0,777		0,690		0,553		0,444		0,366	
Standard deviation match index	0,241		0,241		0,242		0,248		0,290		0,327		0,326		0,307	
CV match index	0,306		0,307		0,309		0,319		0,420		0,593		0,734		0,839	
5%percentile match index	0,230		0,220		0,220		0,210		0,110		0,050		0,040		0,040	
10%percentile match index	0,390		0,390		0,383		0,360		0,215		0,100		0,070		0,060	
20%percentile match index	0,650		0,640		0,640		0,620		0,400		0,200		0,130		0,100	
25%percentile match index	0,710		0,710		0,710		0,690		0,490		0,250		0,160		0,120	
50%percentile match index	0,870		0,870		0,870		0,860		0,770		0,550		0,350		0,260	
Length index in range [0,80 - 1,20]	3877	65,0%	3883	65,1%	3876	64,9%	3858	64,6%	3631	60,8%	3267	54,7%	2936	49,2%	2570	43,1%
Match index in range [0,80 - 1,00]	3405	57,0%	3403	57,0%	3381	56,6%	3318	55,6%	2422	40,6%	1570	26,3%	1036	17,4%	701	11,7%
Total distance travelled [km]	15240,6 km		15232,5 km		15187,4 km		15006,4 km		14494,7 km		14328,7 km		14456,0 km		14421,0 km	
Total distance travelled on bikeways [km]	3169,5	20,8%	3163,2	20,8%	3157,4	20,8%	3134,9	20,9%	2975,9	20,5%	2772,1	19,3%	2661,5	18,4%	2569,7	17,8%
Mean distance travelled on bikeways	601,9 m/route		600,7 m/route		599,8 m/route		595,9 m/route		577,2 m/route		560,1 m/route		555,4 m/route		561,9 m/route	
Routes containing bikeways	3777	63,3%	3774	63,2%	3757	62,9%	3729	62,5%	3395	60,2%	3466	58,1%	3366	56,4%	3265	54,7%
Loading time [seconds]	409,2 s		534,6 s		439,0 s		432,7 s		591,5 s		471,2 s		498,0 s		513,5 s	
Processing data time [seconds]	13813,6 s		14139,6 s		14591,4 s		13751,0 s		14020,5 s		13868,6 s		14186,3 s		13616,3 s	
Total execution time [seconds]	14222,8 s		14674,2 s		15030,4 s		14183,7 s		14611,9 s		14339,8 s		14684,3 s		14129,8 s	

Tabella 3.4: risultati dei test sul database Endomondo, algoritmo ‘new’, al variare di

*C*length

Dalla tabella riassuntiva appare chiaro che il parametro c_{length} influenza i risultati per valori compresi tra 0,005 e 0,02, diminuendo il numero dei matches riusciti (Fig. 3.13, Fig. 3.14) ed aumentandone via via gli errori; invece, per valori inferiori a 0,001, ben quattro prove eseguite (per valori di c_{length} pari rispettivamente a 0,001, 0,0005, 0,0001, e 0,00001) presentano dei risultati praticamente identici in tutti gli aspetti: i valori medi e percentili di *Length Index* e *Match Index*, quelli di *distance error* e il numero di matches aventi caratteristiche accettabili. Ciò evidenzia come l’algoritmo ‘new’ sia quindi poco dipendente dal valore del parametro c_{length} , e di conseguenza sicuramente più affidabile e replicabile in diverse situazioni. Il tempo medio di matching (*mean matching time*), invece, si mantiene pressoché costante, assumendo un valore ancora alto, attorno a 2,0 - 2,2 secondi/traccia, e l’intero processo di map-matching impiega all’incirca 4 ore.



Figura 3.13: numero di matches riusciti al variare di c_{length} (rappresentato in scala logaritmica)

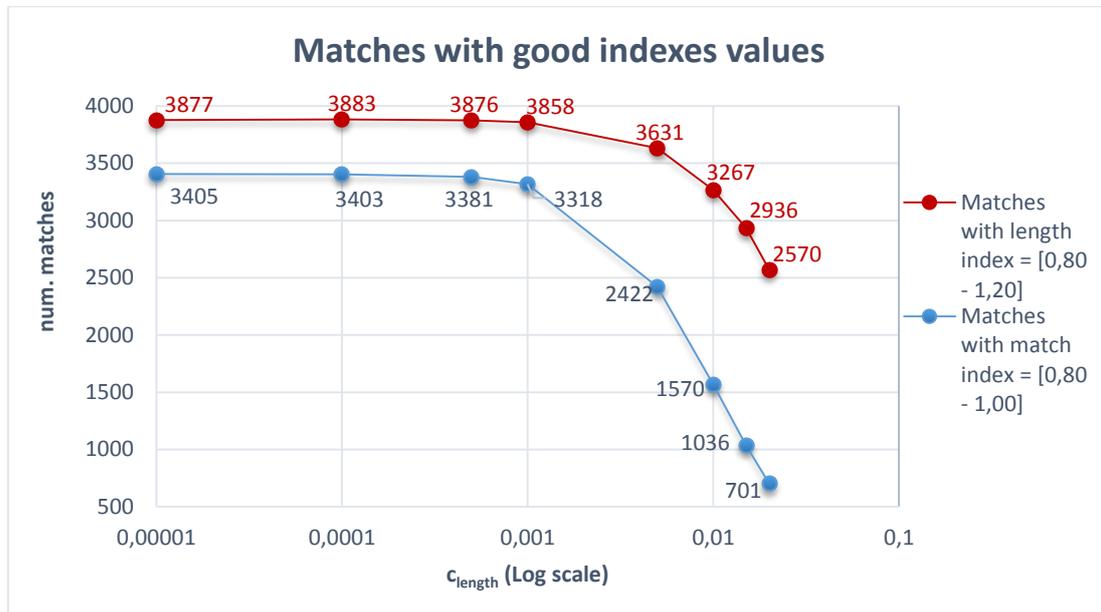


Figura 3.14: numero di matches aventi buoni valori degli indici, al variare di c_{length} (rappresentato in scala logaritmica)

Il valore scelto come ottimale è quindi $c_{length} = 0,0001$.

Per lo studio del parametro c_{bike} sono stati eseguiti cinque test sul database Endomondo, utilizzando cinque diversi valori, compresi tra 0,1 e 0,9. Di seguito è riportata una tabella riassuntiva dei risultati ottenuti.

	c_bike	0,1	c_bike	0,3	c_bike	0,5	c_bike	0,7	c_bike	0,9
Successful matches	5260	88,1%	5267	88,2%	5265	88,2%	5265	88,2%	5266	88,2%
Mean matching time [seconds]	2,069	s	2,132	s	2,054	s	2,007	s	2,130	s
Mean Distance Error	81,4	m/point	76,4	m/point	74,6	m/point	73,8	m/point	73,5	m/point
Standard deviation distance error	219,0	m/point	216,3	m/point	213,9	m/point	213,3	m/point	213,1	m/point
CV distance error	2,69		2,83		2,87		2,89		2,90	
50%percentile distance error	17,2	m/point	16,4	m/point	16,2	m/point	16,2	m/point	16,2	m/point
75%percentile distance error	42,0	m/point	38,4	m/point	37,6	m/point	36,9	m/point	36,1	m/point
90%percentile distance error	206,3	m/point	161,3	m/point	150,8	m/point	149,6	m/point	149,3	m/point
95%percentile distance error	432,0	m/point	406,4	m/point	372,5	m/point	354,4	m/point	353,5	m/point
Mean Length Index	0,970		0,965		0,964		0,964		0,963	
Standard deviation length index	0,306		0,301		0,300		0,301		0,301	
CV length index	0,315		0,312		0,311		0,312		0,312	
50%percentile length index	0,950		0,950		0,950		0,950		0,950	
75%percentile length index	1,070		1,070		1,060		1,060		1,060	
80%percentile length index	1,110		1,110		1,100		1,100		1,100	
90%percentile length index	1,250		1,230		1,230		1,230		1,230	
95%percentile length index	1,410		1,390		1,380		1,380		1,380	
Mean Match Index	0,772		0,782		0,785		0,786		0,787	
Standard deviation match index	0,251		0,245		0,242		0,242		0,241	
CV match index	0,326		0,313		0,308		0,307		0,307	
5%percentile match index	0,200		0,210		0,220		0,222		0,220	
10%percentile match index	0,350		0,370		0,390		0,390		0,390	
20%percentile match index	0,600		0,630		0,640		0,640		0,640	
25%percentile match index	0,680		0,700		0,710		0,710		0,710	
50%percentile match index	0,860		0,870		0,870		0,870		0,870	
Length index in range [0,80 - 1,20]	3823	64,0%	3884	65,1%	3878	65,0%	3882	65,0%	3883	65,1%
Match index in range [0,80 - 1,00]	3253	54,5%	3346	56,1%	3379	56,6%	3387	56,7%	3403	57,0%
Total distance travelled [km]	15314,5	km	15256,8	km	15248,1	km	15242,8	km	15232,5	km
Total distance travelled on bikeways [km]	3904,6	25,5%	3625,4	23,8%	3483,4	22,8%	3341,7	21,9%	3163,2	20,8%
Mean distance travelled on bikeways	742,3	m/route	688,3	m/route	661,6	m/route	634,7	m/route	600,7	m/route
Routes containing bikeways	4048	67,8%	3984	66,7%	3922	65,7%	3838	64,3%	3774	63,2%
Loading time [seconds]	444,8	s	451,9	s	659,7	s	402,8	s	534,6	s
Processing data time [seconds]	13720,7	s	14153,8	s	13622,0	s	13304,8	s	14139,6	s
Total execution time [seconds]	14165,5	s	14605,7	s	14281,8	s	13707,6	s	14674,2	s

Tabella 3.5: risultati dei test sul database Endomondo, algoritmo ‘new’, al variare di

C_{bike}

Dalla tabella riassuntiva appare chiaro che il parametro *c_{bike}* ha un effetto praticamente nullo sui risultati in termini di *Length index*, *Match index* o *distance error*; esso non influenza nemmeno il tempo di esecuzione, che rimane sui 2,0 - 2,1 secondi/traccia. Come ci si aspettava, però, il parametro *c_{bike}* ha una notevole importanza nella scelta dei percorsi ciclabili (Fig. 3.15): la distanza percorsa sulla rete ciclabile, difatti, aumenta al diminuire del parametro, che riduce il costo associato all’arco, rendendolo più appetibile. Di conseguenza, la distanza percorsa su rete ciclabile passa da 600,7

m/itinerario (20,8% del totale), per un valore di c_{bike} pari a 0,9, a 742,3 m/itinerario (25,5% del totale), per un valore di c_{bike} pari a 0,1.

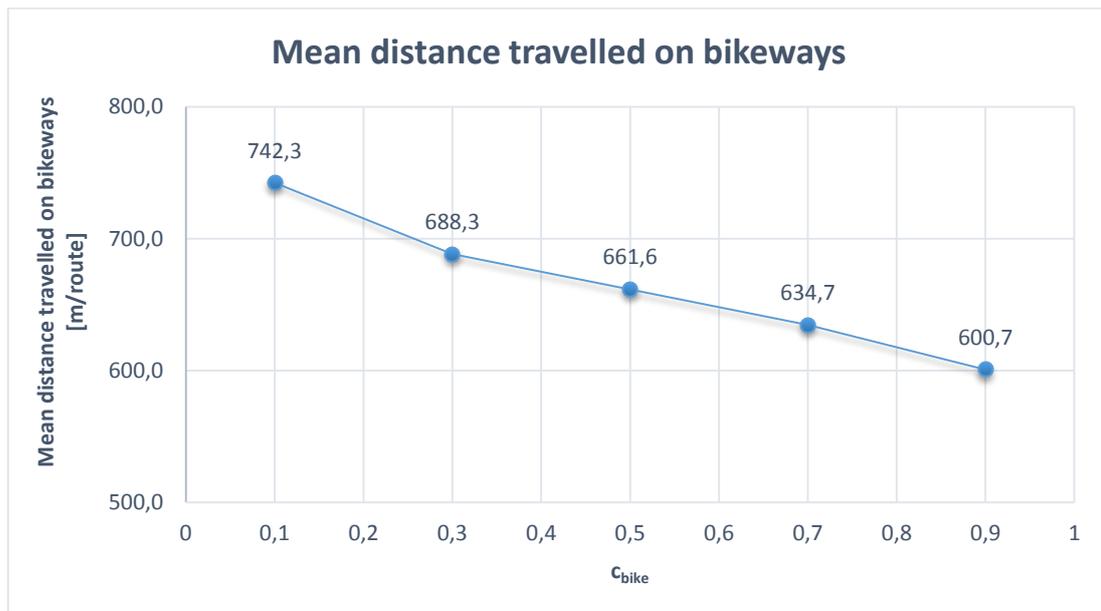


Figura 3.15: distanza media percorsa su rete ciclabile al variare di c_{bike}

Al fine di non forzare il percorso nel passaggio sulla rete ciclabile, comunque, ipotizzando in ogni caso una preferenza per la stessa del ciclista, è stato scelto come ottimale un valore di c_{bike} pari a 0,9.

Per lo studio del parametro *width_buffer* (larghezza del buffer creato attorno all'arco, con lo scopo di valutare se un punto GPS possa essere assegnato ad esso) sono stati eseguiti sei test sul database Endomondo, utilizzando sei diversi valori, compresi tra 10 e 40 metri. Di seguito è riportata una tabella riassuntiva dei risultati ottenuti.

	width_buffer [m]	10	width_buffer [m]	20	width_buffer [m]	25	width_buffer [m]	30	width_buffer [m]	35	width_buffer [m]	40
Successful matches	4889	81,9%	5196	87,0%	5553	88,0%	5266	88,2%	5295	88,7%	5324	89,2%
Mean matching time [seconds]	1,507 s		1,812 s		1,919 s		2,130 s		2,236 s		2,241 s	
Mean Distance Error	81,8 m/point		73,0 m/point		73,9 m/point		73,5 m/point		72,9 m/point		73,4 m/point	
Standard deviation distance error	217,9 m/point		210,8 m/point		213,8 m/point		213,1 m/point		208,6 m/point		210,0 m/point	
CV distance error	2,66		2,89		2,89		2,90		2,86		2,86	
50%percentile distance error	18,3 m/point		16,2 m/point		16,1 m/point		16,2 m/point		16,4 m/point		16,7 m/point	
75%percentile distance error	50,5 m/point		38,3 m/point		36,9 m/point		36,1 m/point		37,2 m/point		37,6 m/point	
90%percentile distance error	182,5 m/point		144,2 m/point		150,4 m/point		149,3 m/point		144,5 m/point		144,8 m/point	
95%percentile distance error	415,8 m/point		352,7 m/point		360,5 m/point		353,5 m/point		352,5 m/point		351,2 m/point	
Mean Length Index	0,998		0,980		0,971		0,963		0,956		0,951	
Standard deviation length index	0,323		0,307		0,307		0,301		0,300		0,299	
CV length index	0,323		0,313		0,316		0,312		0,314		0,314	
50%percentile length index	0,980		0,960		0,950		0,950		0,940		0,930	
75%percentile length index	1,100		1,070		1,060		1,060		1,050		1,050	
80%percentile length index	1,140		1,120		1,110		1,100		1,100		1,090	
90%percentile length index	1,270		1,250		1,240		1,230		1,220		1,220	
95%percentile length index	1,470		1,430		1,410		1,380		1,370		1,360	
Mean Match Index	0,588		0,740		0,769		0,787		0,799		0,809	
Standard deviation match index	0,243		0,244		0,243		0,241		0,239		0,237	
CV match index	0,413		0,329		0,316		0,307		0,299		0,293	
5%percentile match index	0,120		0,190		0,210		0,220		0,240		0,250	
10%percentile match index	0,220		0,340		0,370		0,390		0,410		0,420	
20%percentile match index	0,370		0,570		0,620		0,640		0,660		0,680	
25%percentile match index	0,430		0,640		0,690		0,710		0,730		0,740	
50%percentile match index	0,620		0,810		0,850		0,870		0,880		0,900	
Length index in range [0,80 - 1,20]	3521	59,0%	3804	63,7%	3856	64,6%	3883	65,1%	3890	65,2%	3909	65,5%
Match index in range [0,80 - 1,00]	1119	18,7%	2772	46,4%	3174	53,2%	3403	57,0%	3533	59,2%	3654	61,2%
Total distance travelled [km]	14077,7 km		15124,7 km		15228,4 km		15232,5 km		15315,7 km		15408,3 km	
Total distance travelled on bikeways [km]	2826,3	20,1%	3101,7	20,5%	3135,0	20,6%	3163,2	20,8%	3234,9	21,1%	3289,2	21,3%
Mean distance travelled on bikeways	578,1 m/route		596,9 m/route		596,8 m/route		600,7 m/route		610,9 m/route		617,8 m/route	
Routes containing bikeways	3499	58,6%	3744	62,7%	3785	63,4%	3774	63,2%	3793	63,5%	3823	64,0%
Loading time [seconds]	376,2 s		513,7 s		381,4 s		534,6 s		486,0 s		399,4 s	
Processing data time [seconds]	10310,7 s		12281,0 s		12873,4 s		14139,6 s		14771,7 s		27507,0 s	
Total execution time [seconds]	10869,9 s		12794,6 s		13254,7 s		14674,2 s		15257,7 s		27906,4 s	

Tabella 3.6: risultati dei test sul database Endomondo, algoritmo 'new', al variare di *width_buffer*

Dalla tabella riassuntiva risulta che la larghezza del buffer ha un'influenza trascurabile sia sul *distance error* che sul *Length index*. I valori dei *Match index*, invece, variano notevolmente, aumentando per valori di *width_buffer* maggiori; ciò è abbastanza ovvio, in quanto dei buffers più grandi catturano un maggior numero di punti. Questo può portare ad un peggioramento dei risultati nel caso di reti dense, quali quelle urbane, poiché il contributo di uno stesso punto che appartiene ad una moltitudine di buffers diventa irrilevante ed in tal modo l'informazione perde di significato. Inoltre, buffers molto larghi determinano l'aumento esponenziale del tempo di esecuzione (Fig. 3.16), in quanto sia il primo che l'ultimo punto GPS vengono associati ad un numero maggiore di possibili archi, rispettivamente di origine e di destinazione, e l'algoritmo viene eseguito per ogni coppia origine - destinazione; di conseguenza il tempo medio di esecuzione (*mean matching time*) passa da 1,5 secondi/traccia, corrispondenti a circa 3 h totali (buffers larghi 10 metri), a 4,2 secondi/traccia, corrispondenti a ben 8 h totali (buffers larghi 40 metri).

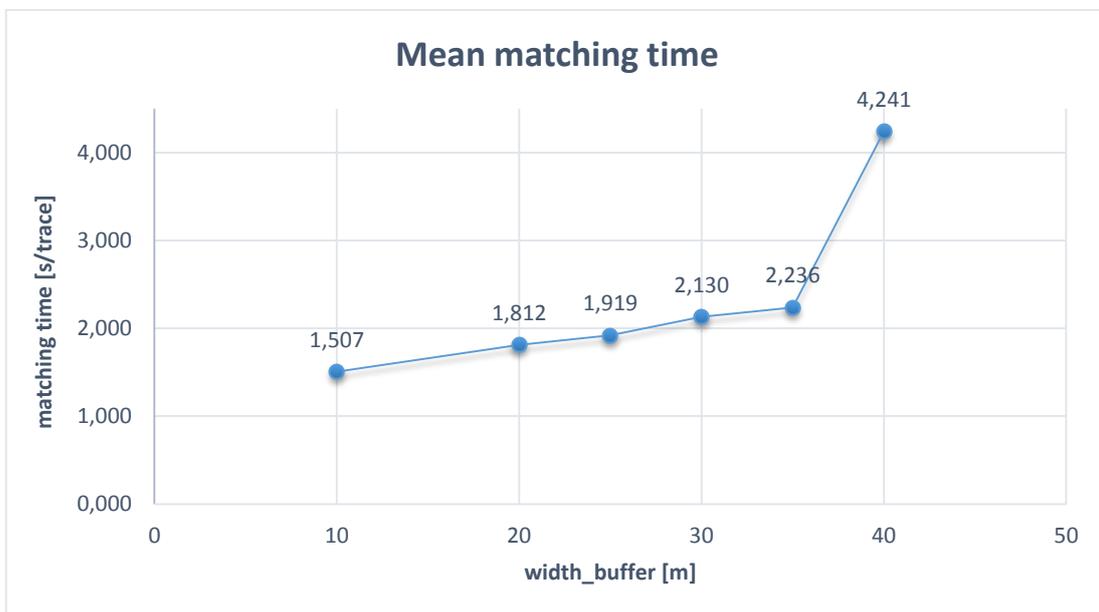


Figura 3.16: tempo medio di matching al variare di *width_buffer*

In base a tali considerazioni è stato quindi scelto un valore ottimale di larghezza dei buffer (*width_buffer*) pari a 30 metri.

4.2 RISULTATI E CONSIDERAZIONI

Dopo aver realizzato l'analisi dei parametri, nella tabella riassuntiva che segue si riportano i risultati ottenuti eseguendo l'algoritmo di map-matching denominato 'new' sul database di tracce Endomondo, posti i parametri c_{length} pari a 0,0001, c_{bike} pari a 0,9 e *width_buffer* pari a 30 metri.

	'new'	
Successful matches	5266	88,2%
Mean matching time [seconds]	2,130	s
Mean Distance Error	73,5	m/point
Standard deviation distance error	213,1	m/point
CV distance error	2,90	
50%percentile distance error	16,2	m/point
75%percentile distance error	36,1	m/point
90%percentile distance error	149,3	m/point
95%percentile distance error	353,5	m/point
Mean Length Index	0,963	
Standard deviation length index	0,301	
CV length index	0,312	
50%percentile length index	0,950	
75%percentile length index	1,060	
80%percentile length index	1,100	
90%percentile length index	1,230	
95%percentile length index	1,380	
Mean Match Index	0,787	
Standard deviation match index	0,241	
CV match index	0,307	
5%percentile match index	0,220	
10%percentile match index	0,390	
20%percentile match index	0,640	
25%percentile match index	0,710	
50%percentile match index	0,870	
Length index in range [0,80 - 1,20]	3883	65,1%
Match index in range [0,80 - 1,00]	3403	57,0%
Total distance travelled [km]	15232,5	km
Total distance travelled on bikeways [km]	3163,2	20,8%
Mean distance travelled on bikeways	600,7	m/route
Routes containing bikeways	3774	63,2%
Loading time [seconds]	534,6	s
Processing data time [seconds]	14139,6	s
Total execution time [seconds]	14674,2	s

Tabella 3.7: risultati del map-matching dell'algorithmo 'new' sul database Endomondo

Esattamente come l'algoritmo 'orig', l'algoritmo 'new', per la sua struttura intrinseca, associa sempre un itinerario ad ogni traccia a cui riesce a collegare un arco di partenza ed uno di arrivo, grazie all'azione dell'algoritmo di Dijkstra al suo interno, il quale ricerca il percorso più breve sulla rete, anche in tratti in assenza di punti GPS, qualunque siano i costi. Anche in questo caso, difatti, ben l'88,2% di matches sono andati a buon fine (*successful matches*), e le tracce per cui non è stato possibile identificare un itinerario sono quelle che si trovano interamente al di fuori della rete considerata, o quelle per le quali non è possibile collegare gli eventuali archi di partenza e di arrivo individuati. L'errore di distanza (*distance error*) ha una media molto alta (circa 73 m/punto) ed è molto distribuito (coefficiente di variazione di 2,9); tuttavia, anche in questo caso, il percentile al 50% evidenzia che circa la metà dei matches presentano un errore (16,2 m/punto) paragonabile agli errori di misurazione, per cui ampiamente accettabili.

Il *Length Index* mostra dei buoni valori, molto concentrati nell'intorno di 1 (media di 0,963, e coefficiente di variazione di 0,312): poco meno dei $\frac{2}{3}$ dei matches presentano un indice compreso in un range accettabile tra 0,8 e 1,2.

Anche il *Match Index* offre dei risultati accettabili: una media non lontana dall'unità (0,787), ed una distribuzione non troppo dispersa (coefficiente di variazione di 0,307), con il 57% di matches aventi un indice maggiore di 0,8.

Così come per l'algoritmo 'orig', il problema principale è il rilevante costo computazionale: 2,1 secondi/traccia è il tempo medio impiegato per il matching delle tracce (*mean matching time*). In totale l'intero processo di map-matching ha richiesto 4 ore, 4 minuti e 34 secondi (*total execution time*), anche in questo caso dovuto per la maggior parte alla fase di assegnazione dei costi a tutti gli archi della rete, da effettuare per ogni traccia.

Nella figura seguente sono raffigurati tutti gli itinerari risultanti (Fig. 3.17).

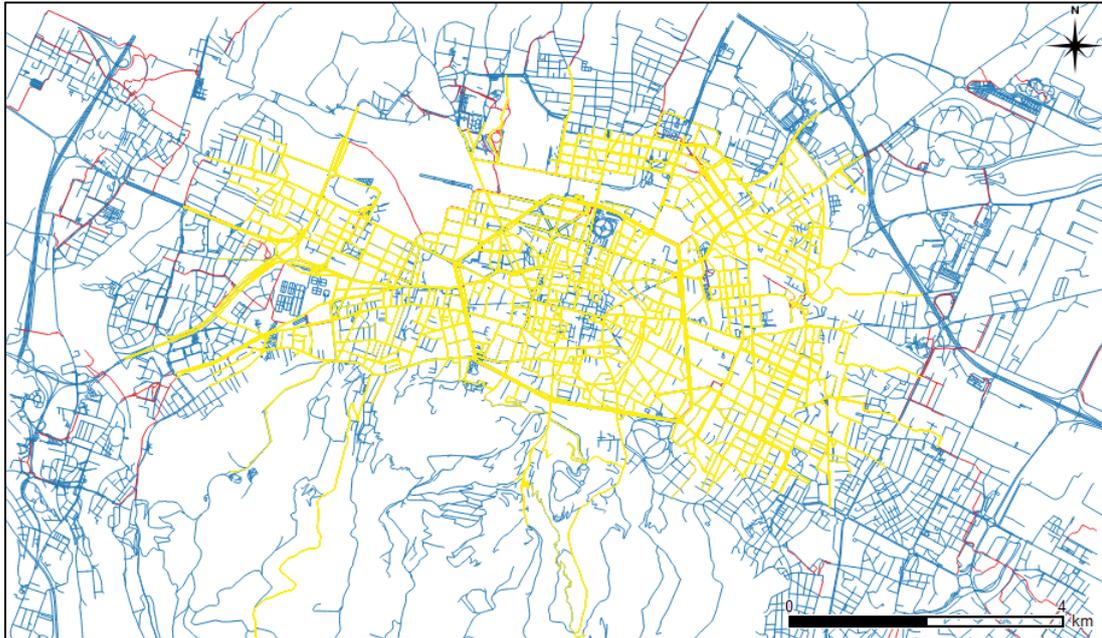


Figura 3.17: itinerari risultanti dal map-matching effettuato dall'algorithmo 'new', in giallo; rete stradale in blu, rete ciclabile in rosso

Di seguito sono riportati i risultati del map-matching effettuato mediante l'algorithmo 'new' sul campione di tracce raccolte:

ID_ROUTE	REAL LEN [m]	'new'					
		LEN_ROUTE [m]	REAL LENGTH INDEX	MATCHIND (%)	LENGTHIND (%)	TIME_MATCH [s]	ERR_DIST [m]
11	992,420	735,120	0,74	0,88	0,72	1,262	21,8
10	2709,370	2693,320	0,99	0,88	0,95	1,288	12,2
12	2084,170	2084,170	1,00	0,98	0,91	2,305	15,1
7#	1128,730	1083,010	0,96	0,83	0,90	0,434	14,9
1	525,350	495,700	0,94	1,00	0,91	1,585	5,3
0	956,870	929,790	0,97	1,00	1,07	1,443	4,9
3	1297,040	1297,040	1,00	1,00	1,02	1,804	6,1
2	1596,590	1266,960	0,79	0,66	0,79	1,017	33,6
5	1266,520	1278,830	1,01	1,00	0,90	1,034	5,8
4	1514,570	1514,570	1,00	1,00	0,91	1,278	6,4
6	735,970	no match					
9	1101,030	1101,030	1,00	1,00	0,99	2,403	6,8
8	1064,700	1074,100	1,01	1,00	0,92	3,172	8,0
	mean	1296,14	0,95	0,94	0,92	1,585	11,7

Tabella 3.8: risultati del matching dell'algorithmo 'new' sul campione di tracce raccolte

I risultati sovraesposti sono soddisfacenti, ad eccezione delle tracce 2 (*Length Index* = 0,79 *Real Length Index* = 0,79, *Match Index* = 0,66) e 11 (*Length Index* = 0,72, *Real Length Index* = 0,74), mentre la traccia 6 non è stata ricostruita. Essi sostanzialmente confermano quanto osservato in precedenza per il database Endomondo.

5. CONFRONTO TRA GLI ALGORITMI ‘orig’ E ‘new’

Dopo avere analizzato singolarmente gli algoritmi denominati ‘orig’ e ‘new’, si procede ad una comparazione dei risultati ottenuti eseguendo il map-matching sul database di tracce Endomondo tramite i due algoritmi. Gli esiti dell’analisi si riportano nella tabella riassuntiva seguente.

	'orig'		'new'	
Traces number (cleaning)	5969		5969	
Successful matches	5284	88,5%	5266	88,2%
Mean matching time [seconds]	2,612	s	2,130	s
Mean Distance Error	63,4	m/point	73,5	m/point
Standard deviation distance error	196,8	m/point	213,1	m/point
CV distance error	3,10		2,90	
50%percentile distance error	14,1	m/point	16,2	m/point
75%percentile distance error	28,5	m/point	36,1	m/point
90%percentile distance error	111,5	m/point	149,3	m/point
95%percentile distance error	286,5	m/point	353,5	m/point
Mean Length Index	0,988		0,963	
Standard deviation length index	0,281		0,301	
CV length index	0,285		0,312	
50%percentile length index	0,960		0,950	
75%percentile length index	1,070		1,060	
80%percentile length index	1,110		1,100	
90%percentile length index	1,230		1,230	
95%percentile length index	1,400		1,380	
Mean Match Index	0,816		0,787	
Standard deviation match index	0,225		0,241	
CV match index	0,276		0,307	
5%percentile match index	0,290		0,220	
10%percentile match index	0,470		0,390	
20%percentile match index	0,710		0,640	
25%percentile match index	0,760		0,710	
50%percentile match index	0,890		0,870	
Length index in range [0,80 - 1,20]	4040	67,7%	3883	65,1%
Match index in range [0,80 - 1,00]	3765	63,1%	3403	57,0%
Total distance travelled [km]	15660,5	km	15232,5	km
Total distance travelled on bikeways [km]	3665,0	23,4%	3163,2	20,8%
Mean distance travelled on bikeways	693,6	m/route	600,7	m/route
Routes containing bikeways	4243	71,1%	3774	63,2%
Loading time [seconds]	506,8	s	534,6	s
Processing data time [seconds]	16741,2	s	14139,6	s
Total execution time [seconds]	17248,0	s	14674,2	s

Tabella 3.9: confronto tra i risultati del map-matching degli algoritmi 'orig' e 'new' sul database Endomondo

Il secondo algoritmo studiato, denominato ‘new’, si proponeva come possibile aggiornamento dell’algoritmo ‘orig’, in quanto, sfruttando la medesima metodologia, apportava dei correttivi al fine di risolvere alcune problematiche di quest’ultimo, evidenziate nel paragrafo 3.3: in particolare, perseguiva il duplice obiettivo di eliminare in buona parte la dipendenza dal parametro C_{length} e di ridurre il tempo di esecuzione.

In ordine al primo punto, come già riportato nel paragrafo 4.1, i risultati delle prove effettuate al variare del parametro C_{length} sono equivalenti; ciò evidenzia come l’algoritmo ‘new’ sia quindi poco dipendente dal valore del parametro C_{length} , e di conseguenza più affidabile e replicabile in diverse situazioni.

In ordine al secondo punto, come appare chiaro dalla tabella suesposta e dalla figura 3.18(b), il tempo medio di matching (*mean matching time*) si è ridotto di circa il 20%, generando un risparmio di tempo di 43 minuti circa sul tempo di esecuzione totale (*total execution time*).

Infine, si può notare come i risultati ottenuti applicando i due diversi algoritmi siano del tutto simili, senza significative perdite di informazioni, come mostrato dal numero di matches (Fig. 3.18 a) e dai valori medi e dei percentili di *distance error*, *length index* e *match index*, compreso il numero di matches aventi indici ritenuti accettabili (Fig. 3.19).

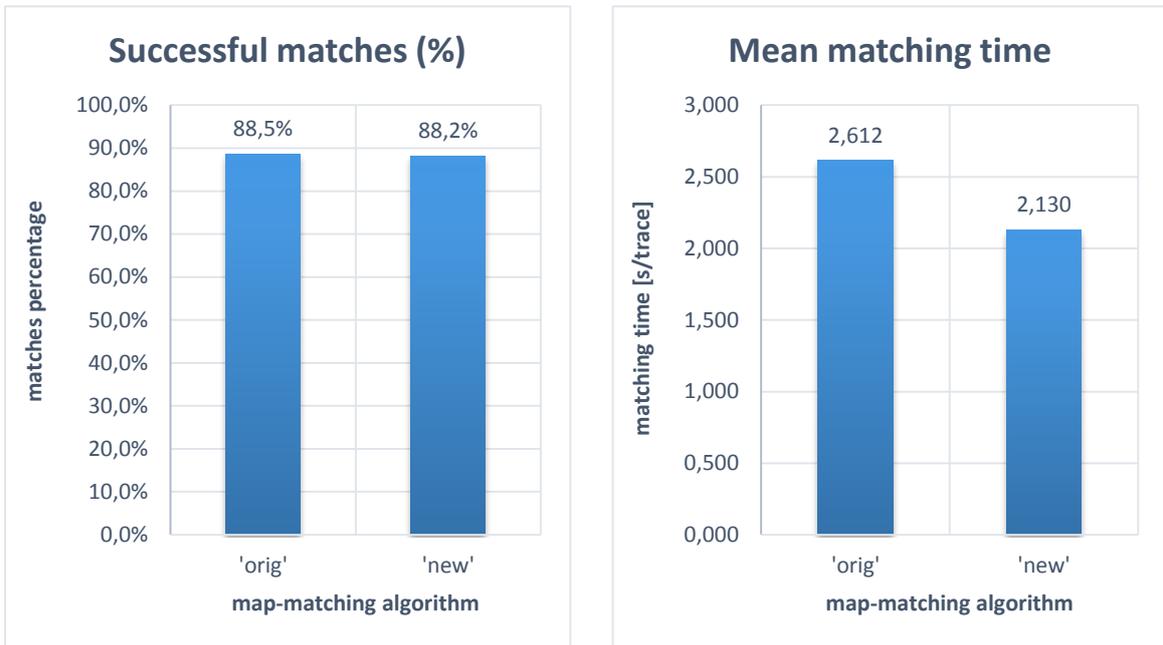


Figura 3.18: percentuale di matches riusciti (a) e tempo medio di matching (b)

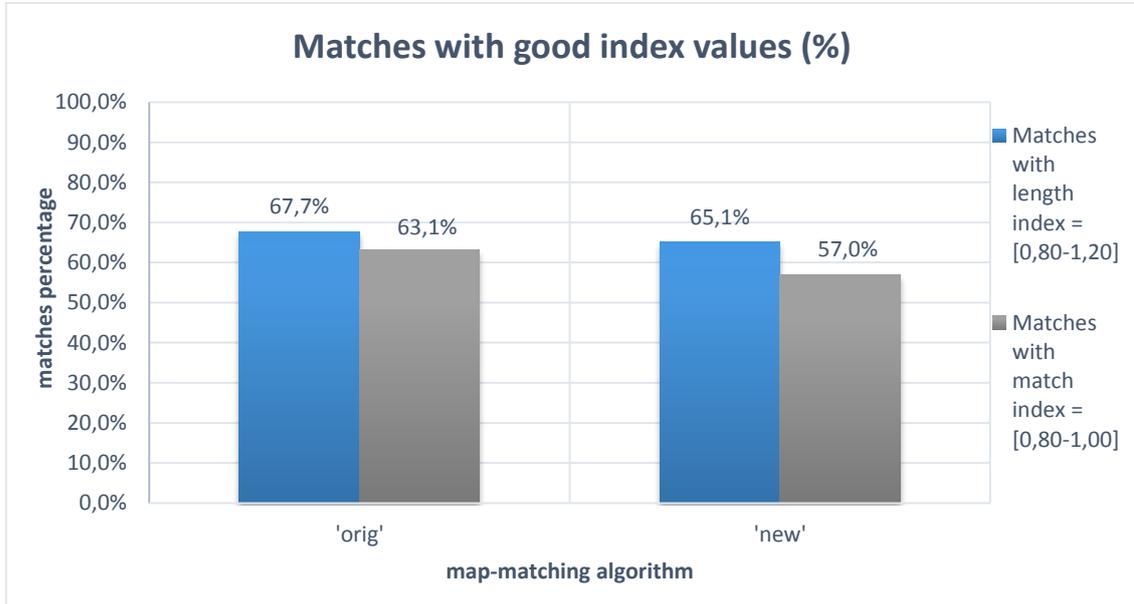


Figura 3.19: percentuale di matches aventi buoni valori degli indici

Per quanto riguarda il campione di tracce raccolte, i risultati si sono dimostrati sovrapponibili tra le due versioni studiate; è invece interessante notare il diverso comportamento nel caso di assenza di punti GPS (Fig. 3.20).

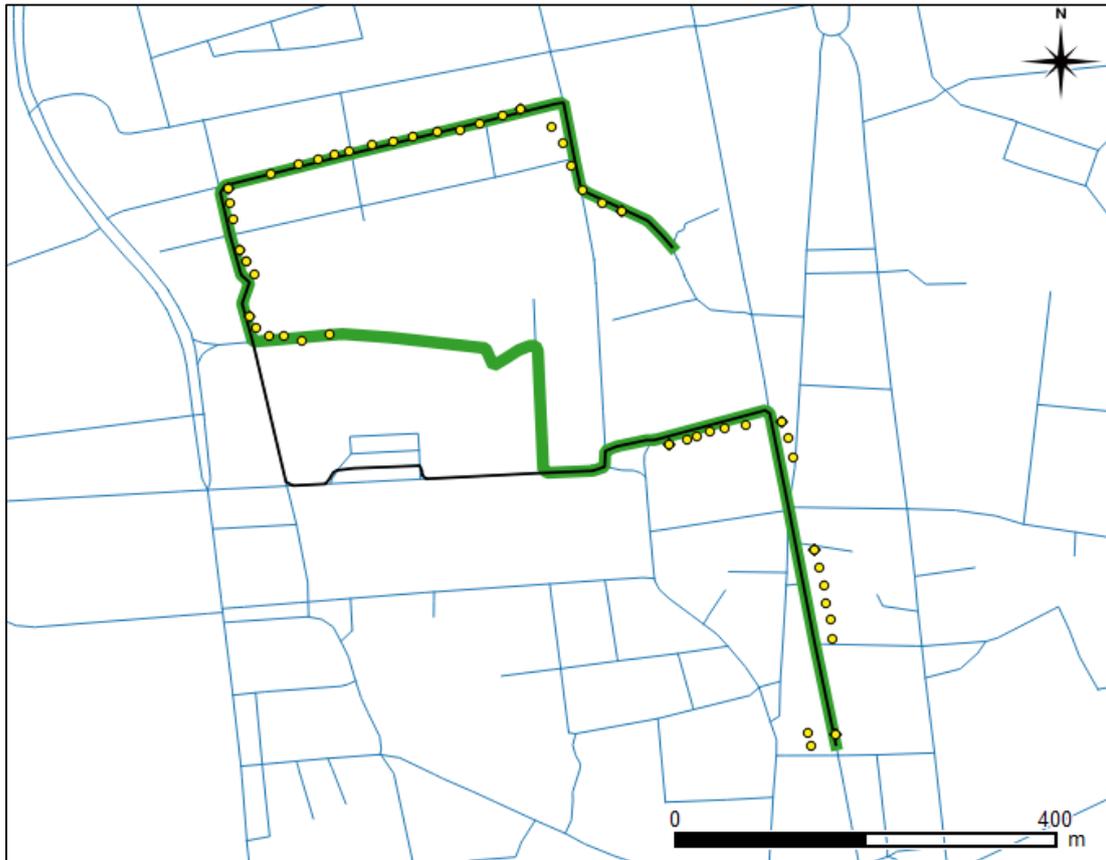


Figura 3.20 a): esempio di map-matching effettuato con l'algoritmo 'orig', sovrapposto al percorso reale (in verde), tratto dal campione raccolto; nel tratto in assenza di punti GPS l'itinerario ricostruito segue il percorso più breve sulla rete, allontanandosi dai punti seguenti (*Length Index* = 0,99; *Real Length Index* = 0,97; *Match Index* = 0,98; *error distance* = 7,4 m)

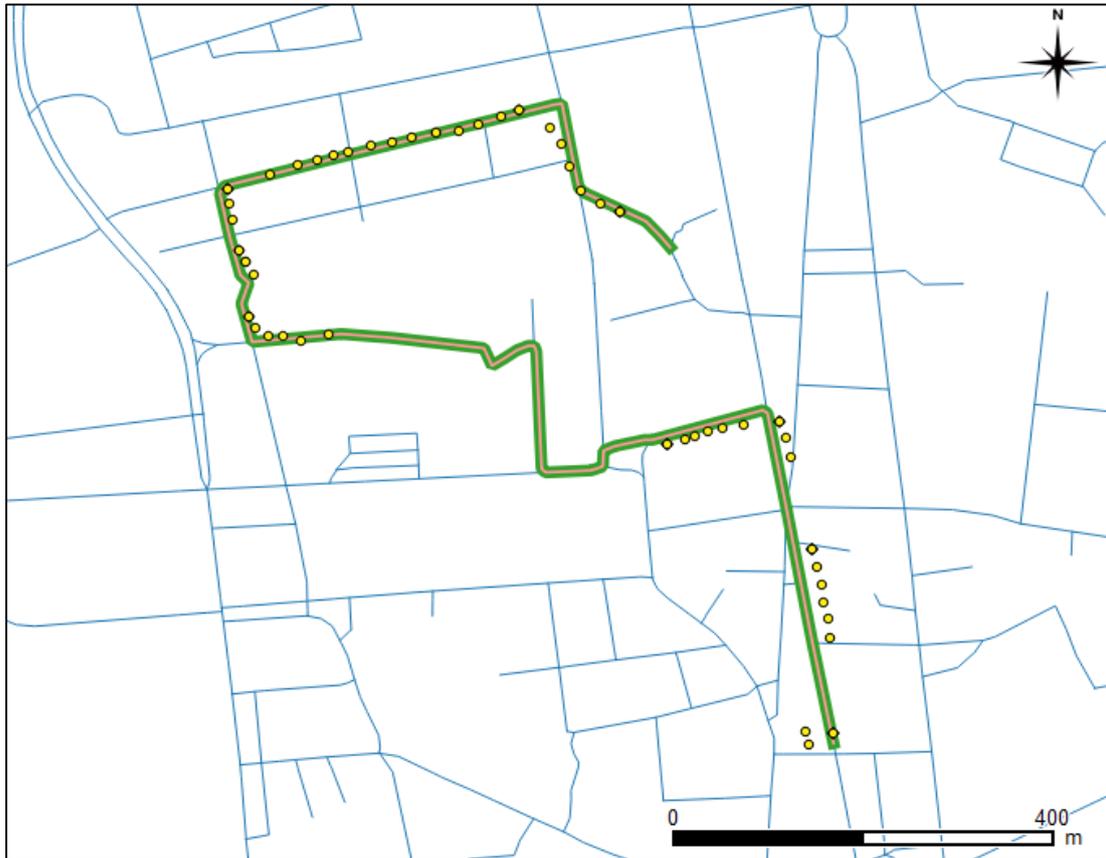


Figura 3.20 b): esempio di map-matching effettuato con l'algoritmo 'new' (in arancione), sovrapposto al percorso reale (in verde), tratto dal campione raccolto; nel tratto in assenza di punti GPS, l'itinerario ricostruito segue il percorso più breve di ricongiunzione tra i due punti successivi. È evidente l'influenza maggiore dei punti tracciati sulla ricostruzione del percorso, ed il matching è perfetto ($Length Index = 1,02$; $Real Length Index = 1,00$; $Match Index = 1,00$; $error distance = 6,1$ m)

6. SECONDA METODOLOGIA: ALGORITMO ‘Marchal’

La seconda metodologia di map-matching analizzata trae spunto dal lavoro di F.Marchal, J.Hackney e K.W.Axhausen (2004).¹¹³

Il terzo algoritmo studiato è quello da loro ideato ed è classificabile come di tipo topologico - probabilistico. Esso è descritto dagli autori come idoneo a trattare set di dati GPS a larga scala, su una rete ad alta risoluzione, in un tempo computazionale accettabile.¹¹⁴ Il suo funzionamento è basato sulla tecnica delle ipotesi multiple (*MHT, multiple hypothesis technique*), ovvero la formulazione di una serie di ipotesi sul percorso sulla base della topologia della rete (connessioni tra gli archi ai nodi) e la conservazione in memoria di un set di possibili alternative per il percorso effettivo, a cui è assegnato un punteggio (*score*): ad ogni passo vengono eliminate le alternative aventi il punteggio peggiore e, tra le rimanenti, è scelta infine quella con il migliore punteggio. La misura fondamentale presa in considerazione è la distanza punto - arco. Dato che l'algoritmo processa i punti sequenzialmente uno dopo l'altro, secondo il loro ordine cronologico, è essenziale che un punto ed il suo successivo non siano troppo distanti nel tempo o nello spazio: per tale motivo la fase di *clean* dei dati è più lunga e complessa, in quanto le tracce aventi tra un punto ed il precedente una distanza maggiore di 300 metri, o un lasso di tempo intercorrente maggiore di 30 secondi, vengono divise (*split*) in corrispondenza di tali punti; ciò aumenta notevolmente il numero delle tracce del database Endomondo, le quali diventano 10971. L'approccio metodologico prevede due fasi fondamentali, da eseguire per ogni traccia:

- 1) Inizializzazione. Il primo passo consiste nel selezionare tutti i possibili archi di partenza, ovvero l'insieme di archi più vicini al primo punto GPS tracciato; nella pratica si traccia un buffer, circolare, di dato raggio (assunto pari a 50 metri) e di centro coincidente con il primo punto GPS, e si selezionano tutti gli archi ricadenti all'interno di esso. Essi vengono memorizzati come archi

¹¹³ Marchal F. et al., 2004.

¹¹⁴ Marchal F. et al., 2004.

iniziali di percorsi diversi ed è assegnato loro un punteggio pari alla rispettiva distanza perpendicolare dal punto Q_i , definita come segue:

$$d(Q_i, \overrightarrow{AB}) = \begin{cases} d_e(Q_i, Q'_i) & \text{if } Q'_i \in [\overrightarrow{AB}] \\ \min \{d_e(Q_i, A); d_e(Q_i, B)\} & \text{elsewhere} \end{cases} \quad (3.6)$$

Con Q'_i proiezione del punto Q_i sull'arco AB : se essa ricade sull'arco stesso, la distanza perpendicolare coincide con la distanza euclidea d_e tra il punto Q_i e la sua proiezione Q'_i ; in caso contrario, la distanza perpendicolare è definita come il minimo tra le distanze eucldee tra il punto Q_i ed i punti estremi dell'arco A e B (Fig. 3.21).

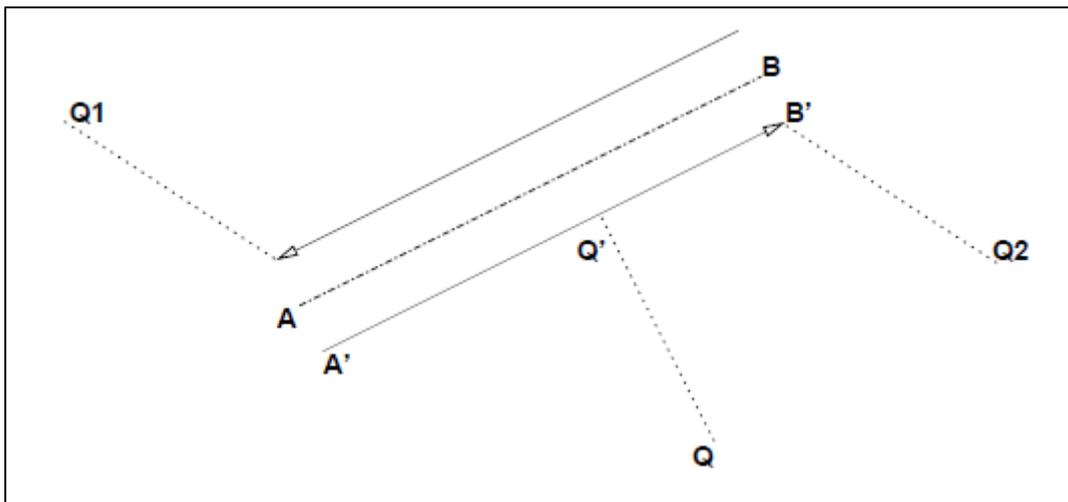


Figura 3.21: distanza perpendicolare dei punti Q , Q_1 e Q_2 (Marchal F. et al., 2004)¹¹⁵

- 2) Avanzamento. Ad ogni punto Q_i successivo al primo, in ordine sequenziale cronologico, per ogni percorso P mantenuto in memoria vengono eseguite le seguenti operazioni:

¹¹⁵ Marchal F. et al., 2004.

- 2.1) Assunto che Q_{i-1} sia stato assegnato all'ultimo arco E_P del percorso P , si controlla se è stata raggiunta l'intersezione seguente o il nodo finale di E_P . Gli autori considerano questa condizione verificata se la sommatoria delle distanze tra i punti GPS assegnati ad uno stesso arco E_P copre una certa percentuale della lunghezza dell'arco stesso, suggerendo un valore del 50%; tale percentuale (α) è un parametro dell'algoritmo. Espressa in termini matematici, la condizione è la seguente:

$$\sum_{k=p1}^i d_e(Q_{k-1}, Q_k) > \alpha \cdot L(E_P) \quad (3.7)$$

con Q_{P0} primo punto assegnato all'arco E_P , $L(E_P)$ lunghezza dell'arco E_P , ed $\alpha = 0,5$, valore suggerito dagli autori.¹¹⁶ Se la condizione è verificata, si assume che l'intersezione è stata superata.

- 2.2) In caso affermativo, mediante una ricerca topologica vengono creati ed aggiunti in memoria una serie di nuovi percorsi (*children paths*) P_{FS} , tanti quanti sono gli archi uscenti dal nodo finale di E_P , i quali differiscono dall'originario percorso P poiché contengono un arco in più, uscente dal nodo finale di E_P , e diverso per ogni P_{FS} (Fig. 3.22); in altre parole essi rappresentano le alternative di percorso, una volta raggiunta l'intersezione. Il punto Q_i viene quindi assegnato all'arco E_{P+1} per ogni percorso P_{FS} . In caso negativo, il punto Q_i è assegnato all'ultimo arco E_P del percorso P , e viene semplicemente aggiornato il punteggio del percorso P , come al punto successivo 2.3.
- 2.3) Lo score di ogni percorso, compresi quelli eventualmente creati, viene aggiornato tramite l'equazione seguente:

$$S_P = \sum_{j=1}^p \sum_{i=1}^N d(Q_i, E_j) \cdot \delta_{ij} \quad (3.8)$$

¹¹⁶ Marchal F. et al., 2004.

con d distanza perpendicolare tra il punto Q_i e l'arco E_j , facente parte del percorso P , e $\delta_{ij} = 1$ se Q_i è associato all'arco E_j , $\delta_{ij} = 0$ altrimenti. Il percorso P è formato da una successione di archi $\{ E_1, E_2, \dots, E_j, \dots, E_p \}$, ed il set di punti GPS $\{ Q_1, Q_2, \dots, Q_i, \dots, Q_N \}$ è formato da N punti.

- 2.4) I percorsi vengono ordinati in base al loro punteggio e vengono eliminati quelli aventi i peggiori punteggi (i più alti), in relazione al numero di percorsi da mantenere in memoria, il quale rappresenta un altro parametro dell'algoritmo (n_routes_follow).

Al termine della sequenza di punti, il percorso che minimizza lo score è scelto come itinerario effettivamente seguito, dopo aver superato un'ultima verifica: è infatti controllato che esso sia arrivato a destinazione, ovvero che l'ultimo arco dell'itinerario ricada almeno in parte all'interno di un buffer circolare, di raggio 50 metri, centrato nell'ultimo punto GPS tracciato, similmente a quanto fatto per il primo punto GPS. Ciò al fine di evitare di fornire risultati errati.

Nella figura seguente è esemplificato il procedimento dell'algoritmo (Fig. 3.22).

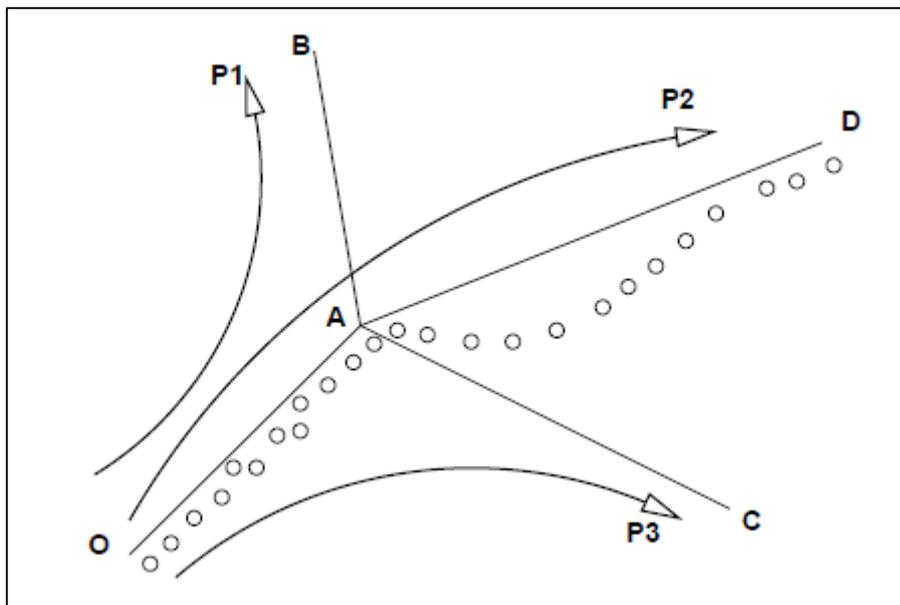


Figura 3.22: creazione dei tre *children paths* (P1, P2, P3) all'intersezione (nodo A), corrispondenti ai tre archi uscenti dal nodo A (AB, AD, AC); l'uso della tecnica delle

ipotesi multiple ed il mantenimento in memoria dei possibili percorsi permette di scegliere il percorso esatto (P2), sebbene dai primi punti GPS dopo l'intersezione lo score migliore sia quello del percorso P3

6.1 ANALISI DEI PARAMETRI

Al fine di individuare il valore ottimale dei parametri, sono stati eseguiti diversi test, i cui risultati sono stati confrontati tramite i valori dell'indice *Length index*, degli errori di distanza (*error distance*), del numero di matches e del tempo impiegato. In questo caso perde di significato l'indice *Match index*, in quanto scompare il concetto di buffer attorno agli archi, preso in considerazione nel primo approccio.

I due parametri analizzati sono il numero di percorsi mantenuti in memoria, *n_routes follow*, e la percentuale α della lunghezza dell'arco da coprire al fine di considerare superata l'intersezione o il nodo finale dell'arco.

Per lo studio del parametro *n_routes follow*, sicuramente il più influente dal punto di vista del costo computazionale, sono stati eseguiti sei test sul database Endomondo, utilizzando sei diversi valori, compresi tra 10 e 100. Di seguito è riportata una tabella riassuntiva dei risultati ottenuti.

	n_routes follow	10	n_routes follow	20	n_routes follow	30	n_routes follow	40	n_routes follow	50	n_routes follow	100
Successful matches	5724	52,2%	6057	55,2%	6182	56,3%	6293	57,4%	6341	57,8%	6466	58,9%
Mean matching time [seconds]	0,482 s		0,637 s		0,776 s		0,891 s		1,005 s		1,626 s	
Matches/matching time differential			2148 s ⁻¹		899 s ⁻¹		965 s ⁻¹		421 s ⁻¹		201 s ⁻¹	
Mean Distance Error	12,2 m/point		12,6 m/point		12,8 m/point		12,7 m/point		12,8 m/point		12,8 m/point	
Standard deviation distance error	19,4 m/point		20,7 m/point		20,8 m/point		20,4 m/point		20,8 m/point		18,8 m/point	
CV distance error	1,60		1,64		1,63		1,61		1,62		1,47	
50%percentile distance error	7,9 m/point		8,0 m/point		8,0 m/point		8,1 m/point		8,1 m/point		8,1 m/point	
75%percentile distance error	12,5 m/point		12,8 m/point		12,9 m/point		13,0 m/point		13,1 m/point		13,2 m/point	
90%percentile distance error	22,0 m/point		23,1 m/point		23,7 m/point		23,4 m/point		23,7 m/point		24,1 m/point	
95%percentile distance error	34,5 m/point		35,0 m/point		35,3 m/point		34,8 m/point		35,2 m/point		36,1 m/point	
Mean Length Index	1,956		1,957		1,956		1,956		1,955		1,955	
Standard deviation length index	0,485		0,477		0,474		0,474		0,472		0,468	
CV length index	0,248		0,244		0,242		0,242		0,241		0,240	
50%percentile length index	1,890		1,900		1,900		1,900		1,900		1,890	
75%percentile length index	2,130		2,130		2,130		2,120		2,120		2,130	
80%percentile length index	2,200		2,200		2,200		2,190		2,190		2,190	
90%percentile length index	2,460		2,454		2,440		2,440		2,440		2,430	
95%percentile length index	2,820		2,790		2,790		2,804		2,800		2,770	
Mean Marchal distance error	12,2 m/point		28,0 m/point		27,8 m/point		27,5 m/point		27,6 m/point		27,0 m/point	
Standard deviation Marchal distance error	37,8 m/point		39,9 m/point		38,4 m/point		37,5 m/point		37,9 m/point		34,5 m/point	
CV Marchal distance error	1,35		1,42		1,38		1,37		1,37		1,28	
50%percentile Marchal distance error	17,0 m/point		16,8 m/point		16,6 m/point		16,5 m/point		16,5 m/point		16,4 m/point	
75%percentile Marchal distance error	28,4 m/point		28,0 m/point		27,6 m/point		27,3 m/point		27,1 m/point		27,0 m/point	
90%percentile Marchal distance error	58,1 m/point		56,8 m/point		56,7 m/point		55,7 m/point		55,8 m/point		55,8 m/point	
95%percentile Marchal distance error	82,4 m/point		82,8 m/point		82,8 m/point		81,5 m/point		82,1 m/point		81,8 m/point	
Length index in range [0,80 - 1,20]	116	1,1%	108	1,0%	109	1,0%	107	1,0%	109	1,0%	109	1,0%
Total distance travelled [km]	15114,5 km		16442,5 km		16820,0 km		17293,9 km		17453,9 km		17942,4 km	
Total distance travelled on bikeways [km]	2725,2	18,0%	2946,4	17,9%	3024,9	18,0%	3122,4	18,1%	3136,5	18,0%	3234,8	18,0%
Mean distance travelled on bikeways	476,1 m/route		486,4 m/route		489,3 m/route		496,2 m/route		494,6 m/route		500,3 m/route	
Routes containing bikeways	3167	28,9%	3447	31,4%	3533	32,2%	3620	33,0%	3654	33,3%	3743	34,1%
Loading time [seconds]	450,9 s		542,0 s		425,1 s		462,6 s		464,6 s		397,2 s	
Processing data time [seconds]	8014,9 s		9721,5 s		10914,1 s		12235,1 s		13359,4 s		19710,3 s	
Total execution time [seconds]	8465,8 s		10263,5 s		11339,2 s		12697,7 s		13824,0 s		20107,5 s	

Tabella 3.10: risultati dei test sul database Endomondo, algoritmo ‘Marchal’, al variare di *n_routes follow*

Dall'esame della tabella riassuntiva è di tutta evidenza che, utilizzando quest'approccio, la percentuale di matches di esito positivo (*Successful matches*) è minore rispetto a quella ottenuta utilizzando la prima metodologia ('orig' e 'new'), attorno al 50-60%. I valori dei *Length index* si distribuiscono attorno ad un valore medio di 2, quindi sono praticamente raddoppiati rispetto al valore ideale, ovvero 1, ed i matches aventi un valore accettabile, compreso tra 0,8 e 1,2, sono appena un centinaio, circa l'1%. Tuttavia, sia gli errori di distanza calcolati al termine del matching (*error distance*) che quelli calcolati durante il processo di associazione dei punti agli archi (*Marchal distance error*) presentano dei valori bassi, dell'ordine di grandezza delle decine di metri. Tali caratteristiche sono presenti per qualsiasi valore di *n_routes follow*: esse sono dunque specifiche dell'algoritmo 'Marchal' e saranno discusse in seguito nel paragrafo 6.2.

L'effetto più importante della scelta del numero di percorsi da mantenere in memoria è invece sul tempo di esecuzione: esso mostra un valore decisamente basso, che ovviamente aumenta all'aumentare di *n_routes follow*, dovendosi compiere un maggior numero di operazioni, passando da un valore di 0,482 secondi/traccia ($n = 10$), corrispondenti a circa 2 h e mezza totali, a 1,626 secondi/traccia ($n = 100$), corrispondenti a circa 5 h e mezzo, rivelando una tendenza di tipo approssimativamente lineare (Fig. 3.23).

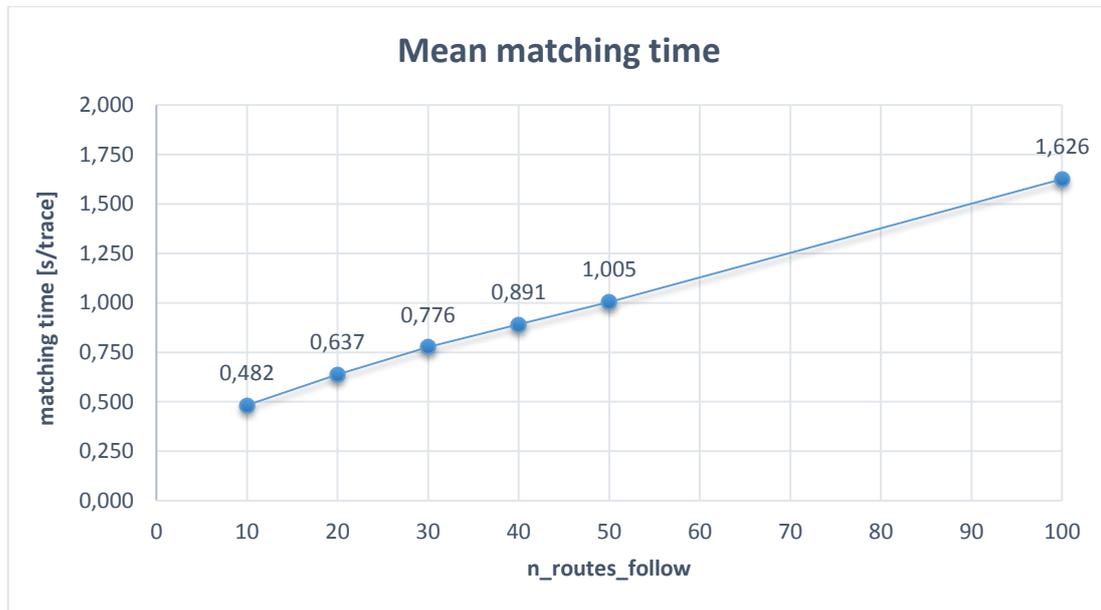


Figura 3.23: tempo medio di matching al variare del numero di percorsi in memoria

Considerando fondamentale per l'algoritmo Marchal il vantaggio in termini di tempo di esecuzione, è stato costruito il rapporto incrementale *Diff* (Equazione 3.9) tra il numero di matches riusciti (*successful matches*) ed il tempo medio di matching (*mean matching time*), al variare di *n_routes_follow* (Fig. 3.24): sulla base di tale rapporto, si è stabilito che il valore ottimale sia pari a 40. Tuttavia, tale scelta, oltre che dipendere dal grado di risoluzione della rete, dipende anche dall'importanza che si vuole attribuire alla velocità di esecuzione piuttosto che all'accuratezza del matching stesso. Difatti, si potrebbe ritenere che l'incremento del numero di matches riusciti, del tutto marginale (13% massimo, per $n = 100$, rispetto al caso in cui $n = 10$), non sia tale da giustificare l'adozione di un *n_routes_follow* maggiore, che genera invece tempi di esecuzione molto più dilatati (aumento del 137%, quindi tempi più che raddoppiati, per $n = 100$, rispetto a $n = 10$).

$$Diff = \frac{\Delta(\text{successful matches})}{\Delta(\text{mean matching time})} \quad (3.9)$$

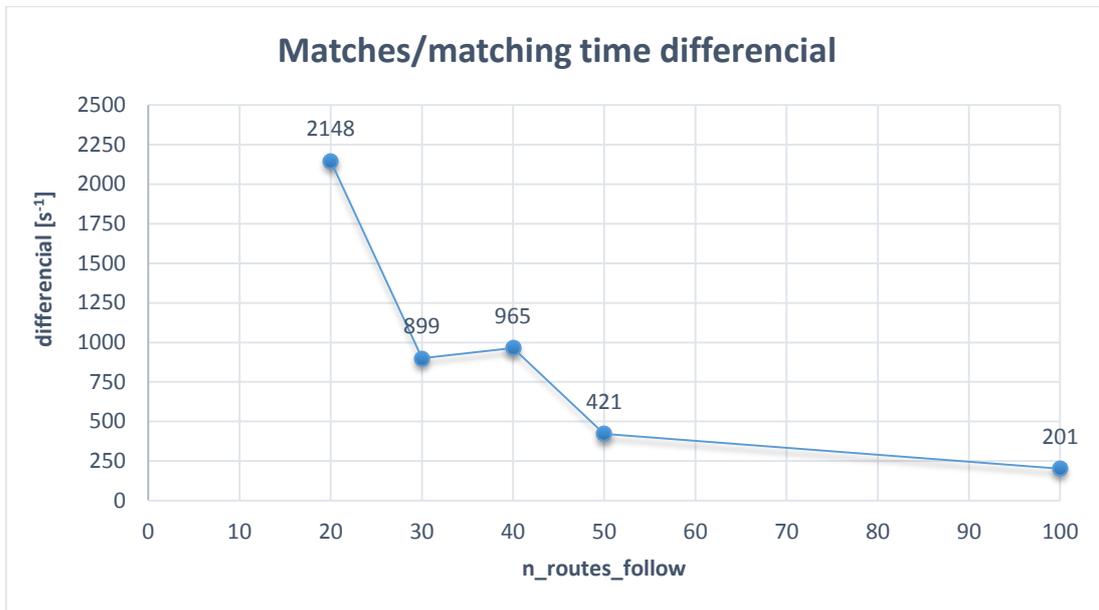


Figura 3.24: rapporto incrementale tra il numero di matches riusciti ed il tempo medio di matching, al variare del numero di percorsi mantenuto in memoria

Per lo studio del parametro α , determinante per stabilire se l'intersezione o il nodo finale dell'arco siano stati raggiunti, e di conseguenza per la ricostruzione del percorso, sono stati eseguiti quattro test sul database Endomondo, utilizzando quattro diversi valori, compresi tra 0,3 e 0,9. Di seguito è riportata una tabella riassuntiva dei risultati ottenuti.

	α	0,3	α	0,5	α	0,7	α	0,9
Successful matches	5859	53,4%	6293	57,4%	6491	59,2%	6532	59,5%
Mean matching time [seconds]	0,959	s	0,891	s	0,924	s	0,967	s
Mean Distance Error	13,5	m/point	12,7	m/point	12,6	m/point	12,6	m/point
Standard deviation distance error	22,8	m/point	20,4	m/point	20,5	m/point	20,9	m/point
CV distance error	1,69		1,61		1,62		1,65	
50%percentile distance error	8,3	m/point	8,1	m/point	7,9	m/point	7,8	m/point
75%percentile distance error	13,3	m/point	13,0	m/point	12,8	m/point	12,7	m/point
90%percentile distance error	24,6	m/point	23,4	m/point	23,5	m/point	23,5	m/point
95%percentile distance error	37,1	m/point	34,8	m/point	35,6	m/point	36,6	m/point
Mean Length Index	2,016		1,956		1,923		1,889	
Standard deviation length index	0,553		0,474		0,444		0,439	
CV length index	0,274		0,242		0,231		0,232	
50%percentile length index	1,930		1,900		1,880		1,840	
75%percentile length index	2,170		2,120		2,090		2,060	
80%percentile length index	2,250		2,190		2,170		2,130	
90%percentile length index	2,530		2,440		2,390		2,359	
95%percentile length index	2,950		2,804		2,680		2,660	
Mean Marchal distance error	31,9	m/point	27,5	m/point	25,0	m/point	23,4	m/point
Standard deviation Marchal distance error	42,8	m/point	37,5	m/point	36,9	m/point	36,5	m/point
CV Marchal distance error	1,34		1,37		1,47		1,56	
50%percentile Marchal distance error	18,8	m/point	16,5	m/point	14,5	m/point	12,8	m/point
75%percentile Marchal distance error	31,9	m/point	27,3	m/point	23,5	m/point	21,0	m/point
90%percentile Marchal distance error	67,6	m/point	55,7	m/point	49,7	m/point	47,8	m/point
95%percentile Marchal distance error	102,6	m/point	81,5	m/point	83,2	m/point	80,2	m/point
Length index in range [0,80 - 1,20]	75	0,7%	107	1,0%	136	1,2%	203	1,9%
Total distance travelled [km]	16150,9	km	17293,9	km	17707,4	km	17741,2	km
Total distance travelled on bikeways [km]	2752,2	17,0%	3122,4	18,1%	3304,4	18,7%	3431,9	19,3%
Mean distance travelled on bikeways	469,7	m/route	496,2	m/route	509,1	m/route	525,4	m/route
Routes containing bikeways	3281	29,9%	3620	33,0%	3751	34,2%	3805	34,7%
Loading time [seconds]	406,6	s	462,6	s	415,1	s	418,4	s
Processing data time [seconds]	13085,6	s	12235,1	s	12575,6	s	12994,6	s
Total execution time [seconds]	13492,2	s	12697,7	s	12990,7	s	13413,0	s

Tabella 3.11: risultati dei test sul database Endomondo, algoritmo ‘Marchal’, al variare di α

Dalla tabella riassuntiva è possibile notare che il numero di matches (*Successful matches*) rimane relativamente basso, attorno al 50-60%; i valori di *Length Index*, *distance error* e *Marchal distance error* sono comparabili, nonostante vi sia un leggero miglioramento per valori maggiori di α . Anche il tempo di matching non subisce delle variazioni significative.

Nella pratica il parametro α sembra dare risultati migliori per valori vicini a 1; tuttavia, valori troppo prossimi all’unità forzano l’algoritmo ad attendere che venga coperta dai punti GPS l’intera lunghezza dell’arco prima di passare agli archi successivi, il che può causare errori notevoli, perdendo l’esatta associazione tra il punto e l’arco, specialmente nel caso in cui i punti non siano numerosi, né abbastanza vicini. Valori

prossimi allo 0, invece, suggeriscono che l'intersezione a fine arco è stata raggiunta quasi immediatamente, e quindi conducono ad un notevole accumulo di percorsi diversi creati, con il rischio notevole di perdere il percorso più probabile.

Per tali motivi, è stato ritenuto che il valore ottimale sia $\alpha = 0,7$.

6.2 RISULTATI E CONSIDERAZIONI

Dopo aver realizzato l'analisi dei parametri, nella tabella riassuntiva che segue si riportano i risultati ottenuti eseguendo l'algoritmo di map-matching denominato 'Marchal' sul database di tracce Endomondo, posti i parametri *n_routes follow* pari a 40 e α pari a 0,7.

	'Marchal'	
Traces number (cleaning)	10971	
Successful matches	6491	59,2%
Mean matching time [seconds]	0,924 s	
Mean Distance Error	12,6 m/point	
Standard deviation distance error	20,5 m/point	
CV distance error	1,62	
50%percentile distance error	7,9 m/point	
75%percentile distance error	12,8 m/point	
90%percentile distance error	23,5 m/point	
95%percentile distance error	35,6 m/point	
Mean Length Index	1,923	
Standard deviation length index	0,444	
CV length index	0,231	
50%percentile length index	1,880	
75%percentile length index	2,090	
80%percentile length index	2,170	
90%percentile length index	2,390	
95%percentile length index	2,680	
Length index in range [0,80 - 1,20]	136	1,2%
Total distance travelled [km]	17707,4 km	
Total distance travelled on bikeways [km]	3304,4	18,7%
Mean distance travelled on bikeways	509,1 m/route	
Routes containing bikeways	3751	34,2%
Loading time [seconds]	415,1 s	
Processing data time [seconds]	12575,6 s	
Total execution time [seconds]	12990,7 s	

Tabella 3.12: risultati del map-matching dell'algoritmo 'Marchal' sul database Endomondo

Dalla tabella riassuntiva si può notare che i matches riusciti (*successful matches*) sono circa il 60%, con un tempo medio di matching pari a 0,924 secondi/traccia (*mean matching time*) ed un tempo totale di esecuzione di 3 h, 36 minuti e 30 secondi (*total execution time*). L'aspetto del costo computazionale è senza dubbio il principale elemento a sostegno di tale algoritmo, in quanto il numero di operazioni da eseguire è

decisamente minore, e di conseguenza il tempo di matching si mantiene su valori molto più bassi rispetto a quelli ottenuti utilizzando la metodologia degli algoritmi 'orig' e 'new'. Sebbene i valori della media (12,6 m/punto) e dei percentili degli errori di distanza (*distance error*) siano ottimi, dell'ordine di grandezza comparabile all'errore del GPS stesso, i valori dei *Length Index* sono distribuiti attorno al valore medio di 1,923; sono quindi praticamente raddoppiati rispetto a quelli che dovrebbero essere i valori significativi. Difatti, soltanto 136 itinerari (1,2% del totale) presentano un indice compreso tra 0,8 e 1,2, in un range ritenuto accettabile.

Ciò è dovuto al punteggio (*score*) associato ad ogni percorso, il quale dipende esclusivamente dalla distanza punto - arco; in una rete urbana molto densa, come quella oggetto di studio, specialmente nel centro di Bologna, gli archi si trovano in posizioni abbastanza ravvicinate e non è raro che i punti GPS alle intersezioni si localizzino proprio in prossimità di archi vicini a quello effettivamente percorso, il che determina l'attribuzione di un punteggio quasi nullo ad un percorso errato. Gli errori ricorrono in particolar modo nelle situazioni in cui vi sia un incrocio con un arco bidirezionale: l'algoritmo attribuisce un punteggio (distanza) quasi nullo all'arco - errato - in prossimità del quale vi è il punto GPS, quindi compie nell'itinerario una deviazione errata all'intersezione e torna infine indietro all'intersezione stessa, essendo l'arco di deviazione bidirezionale; da lì riprende il cammino (Fig. 3.25).

Pertanto, a causa di tali deviazioni dal percorso principale, la lunghezza dell'itinerario cresce notevolmente, all'incirca raddoppiando, ed i *Length Index* presentano dei valori distribuiti nell'intorno di 2; conseguentemente, gli itinerari ricostruiti sono poco rappresentativi.

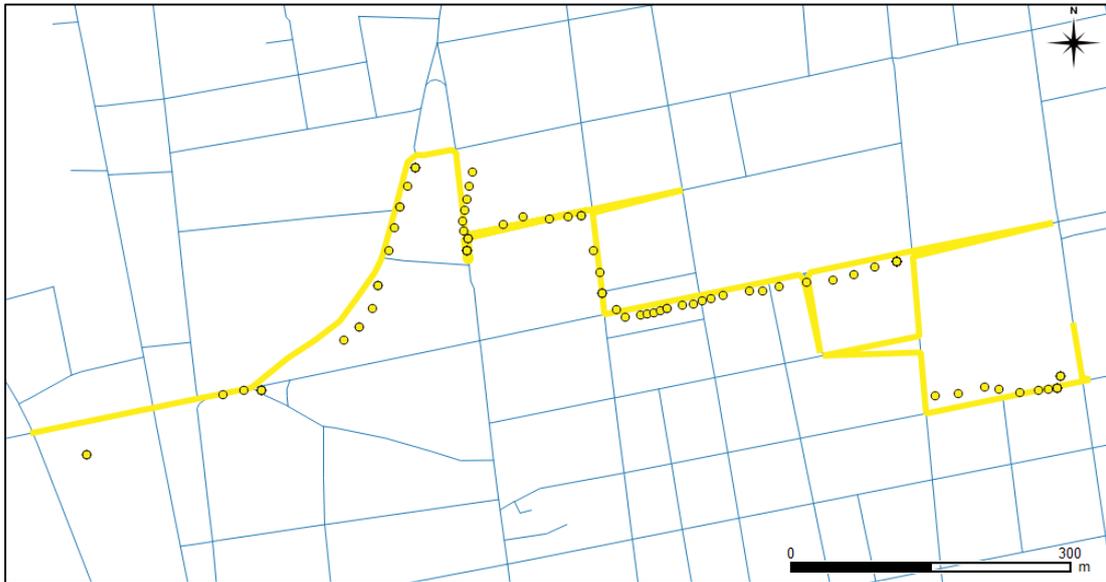


Figura 3.25 a): esempio di errori rilevati nel map-matching effettuato tramite l'algoritmo 'Marchal'; il percorso in figura è tratto dal campione raccolto (*Length Index* = 1,49; *Real Length Index* = 1,60; *error distance* = 5,7 m/point)

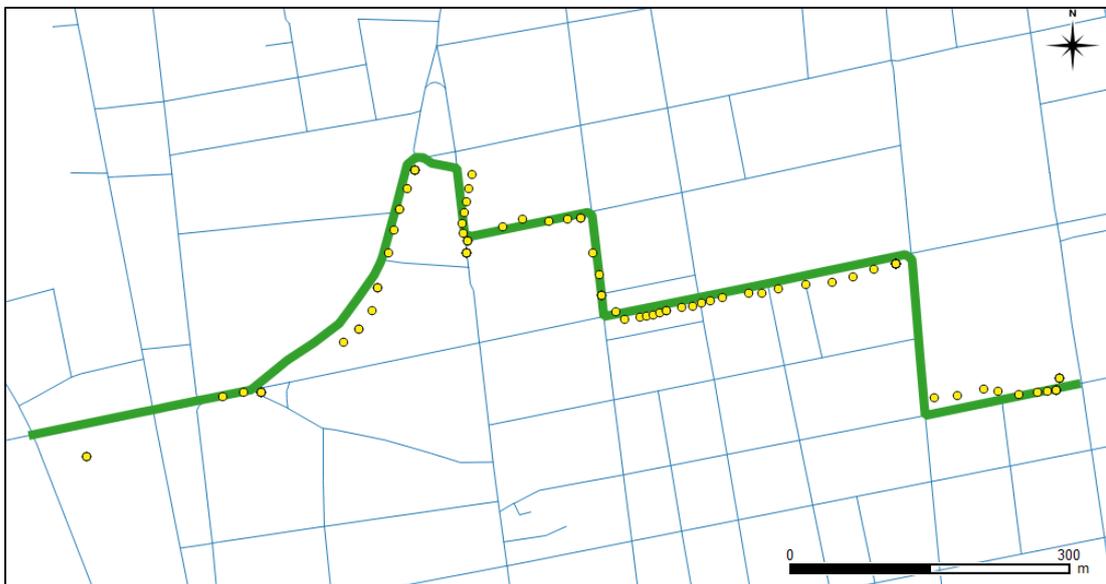


Figura 3.25 b): percorso reale; sono evidenti le differenze con l'itinerario ricostruito dal map-matching effettuato tramite l'algoritmo 'Marchal'

Nella figura seguente sono raffigurati tutti gli itinerari risultanti (Fig. 3.26).

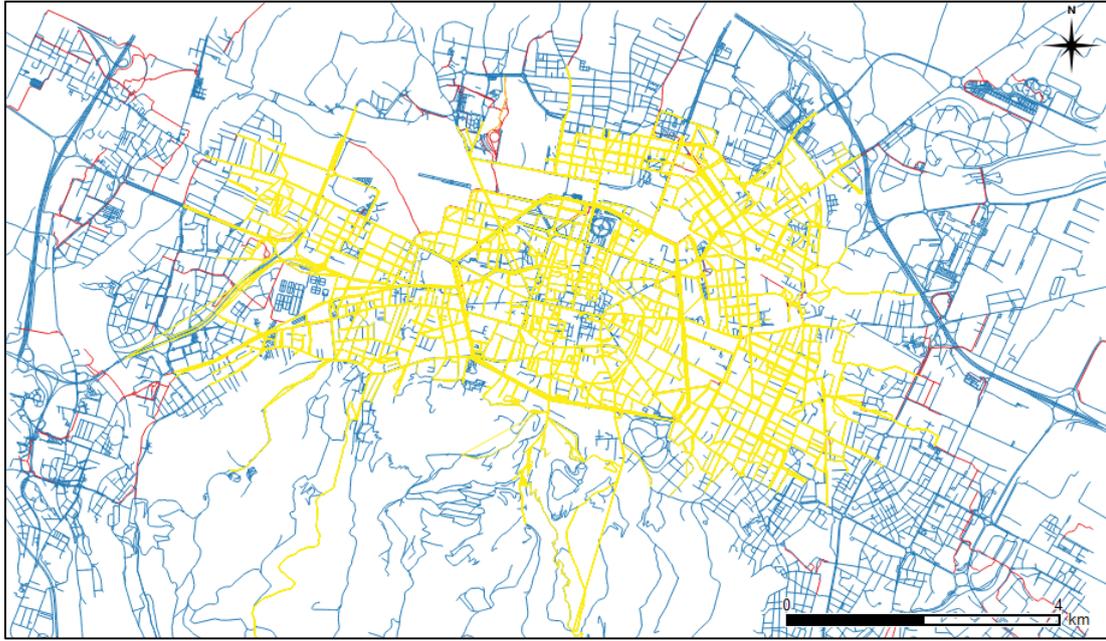


Figura 3.26: itinerari risultanti dal map-matching effettuato dall'algoritmo 'Marchal', in giallo; rete stradale in blu, rete ciclabile in rosso

Di seguito sono riportati i risultati del map-matching effettuato mediante l'algoritmo 'Marchal' sul campione di tracce raccolte:

		'Marchal'				
ID_ROUTE	REAL LEN [m]	LEN_ROUTE [m]	REAL LENGTH INDEX	LENGTHIND	TIME_MATCH [s]	ERR_DIST [m]
11	992,420	1189,330	1,20	1,17	0,593	7,6
10	2709,370	3692,810	1,36	1,30	1,496	21,1
12	2084,170	2806,590	1,35	1,14	1,590	17,1
7#	1128,730	1801,590	1,60	1,49	0,608	5,7
1	525,350	739,450	1,41	1,36	0,326	2,3
0	956,870	1000,680	1,05	1,15	0,531	4,9
3*	1297,040	1247,360	0,96	1,30	0,763	4,9
2	1596,590	2624,950	1,64	1,67	0,998	5,4
5	1266,520	1933,170	1,53	1,36	2,387	5,7
4	1514,570	2169,430	1,43	1,29	0,919	6,5
6	735,970	870,970	1,18	1,24	0,328	10,6
9	1101,030	1906,110	1,73	1,71	0,702	5,6
8	1064,700	1848,040	1,74	1,59	0,514	4,0
	mean	1833,11	1,40	1,37	0,904	7,8

Tabella 3.13: risultati del matching dell'algoritmo 'Marchal' sul campione di tracce raccolte

I risultati suesposti confermano i problemi relativi alle intersezioni e alla lunghezza degli itinerari osservati in precedenza per il database Endomondo: difatti, i *Length Index* e i *Real Length Index* sono decisamente elevati, nonostante gli errori sulla distanza siano bassi. La traccia 3* è evidenziata in quanto è stata divisa durante la fase di filtraggio per la corretta esecuzione dell'algoritmo.

7. SECONDA METODOLOGIA: ALGORITMO PROPOSTO 'NewMarchal'

Il quarto algoritmo studiato, denominato 'NewMarchal', è proposto come possibile perfezionamento di quello studiato in precedenza, ideato da Marchal, dato che questo ha dimostrato notevoli potenzialità, in particolar modo nell'efficienza computazionale. L'algoritmo presentato, infatti, sfrutta la medesima metodologia, basata sulla tecnica

delle ipotesi multiple, sul mantenimento in memoria di un certo numero di itinerari alternativi e sulla ricerca topologica degli archi susseguentisi in ogni percorso, seppur con delle modifiche. Gli scopi principali di tali cambiamenti sono aumentare il numero di matches riusciti e migliorare la loro accuratezza, in particolare limitando il più possibile gli errori alle intersezioni con archi bidirezionali, evidenziati nel paragrafo 6.2.

Il funzionamento è del tutto analogo a quello dell'algoritmo 'Marchal', ad eccezione di due elementi.

La prima modifica è stata realizzata sul calcolo del punteggio (*score*) di ogni percorso, in quanto si sono tenuti in considerazione due concetti: la distanza punto - arco, la quale è stata comunque modificata imponendo una distanza minima, e la similitudine tra la direzione dell'arco e la traiettoria dei punti GPS consecutivi, che è stata individuata mediante spezzate che uniscono i suddetti punti.

L'idea di base è di premiare il parallelismo e penalizzare la perpendicolarità tra il segmento che unisce due punti GPS successivi e l'arco, in quanto si presume che tali segmenti siano indicativi della direzione effettiva dell'utente, e quanto più essi sono paralleli ad un arco, tanto più è probabile che l'utente abbia attraversato proprio tale arco. In aggiunta, si è notato sperimentalmente dalle tracce raccolte che, in genere, i punti GPS, pur non essendo perfettamente localizzati, tendono ad allinearsi lungo la stessa direzione dell'arco e ad addensarsi sullo stesso lato (Fig. 3.27), confermando così la bontà dell'affermazione precedente. Inoltre, al fine di evitare l'associazione del punto ad archi errati, determinata dall'errore di misurazione della posizione GPS, sono state apportate delle modifiche alla distanza punto - arco da considerare, introducendo la *pseudo distanza*, che tiene conto del suddetto errore di misurazione.

La seconda principale modifica è stata apportata alle condizioni per le quali si considera raggiunta l'intersezione o il nodo finale di un arco, aggiungendo un'ulteriore verifica sulla posizione relativa del punto rispetto all'arco.

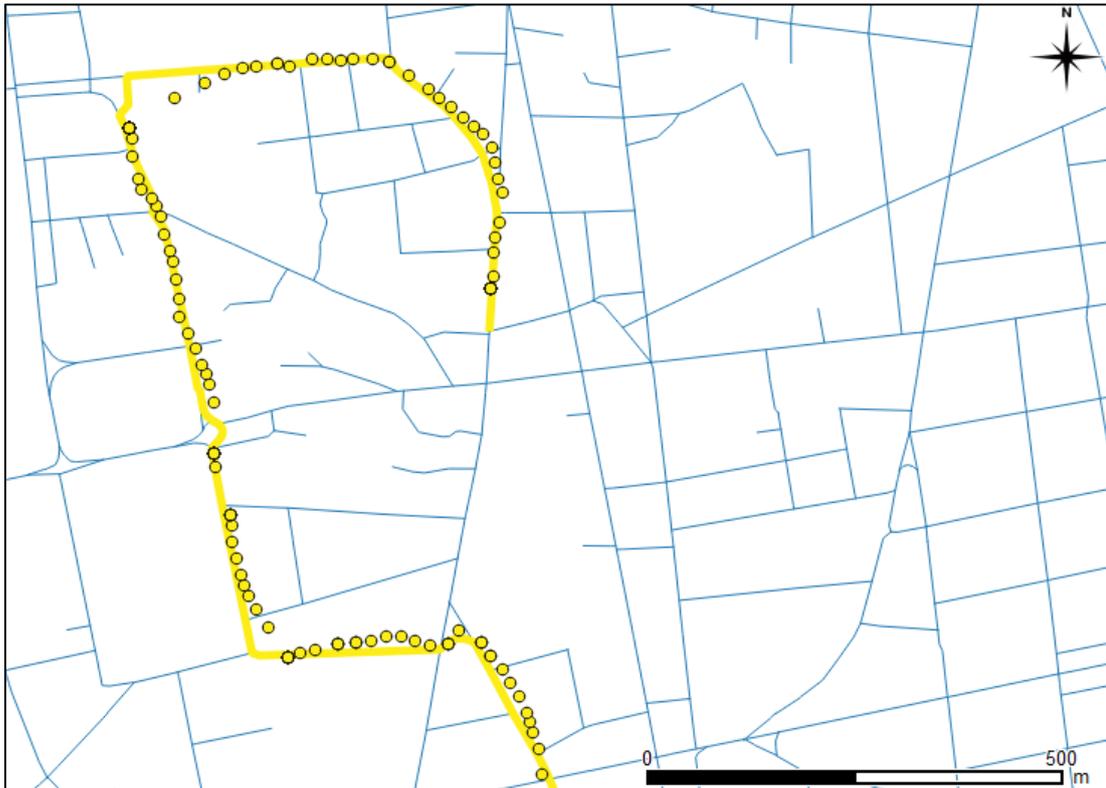


Figura 3.27: punti GPS e percorso reale, tratti dal campione raccolto; i punti si allineano lungo la stessa direzione dell'arco e si addensano tutti sullo stesso lato

Dopo il *clean* dei dati, effettuato in modo analogo al caso precedente (paragrafo 6) - ossia frazionando le tracce aventi due punti consecutivi distanti più di 300 metri o registrati dopo più di 30 secondi - anche l'algoritmo 'NewMarchal' può essere diviso in due fasi principali, eseguite per ogni traccia:

- 1) Inizializzazione. Il primo passo consiste nel selezionare tutti i possibili archi di partenza, ovvero l'insieme di archi più vicini al primo punto GPS tracciato; nella pratica si traccia un buffer, circolare, di dato raggio (assunto pari a 50 metri) e di centro coincidente con il primo punto GPS, e si selezionano tutti gli archi ricadenti all'interno di esso. Essi vengono memorizzati come archi iniziali di percorsi diversi, a cui è assegnato un punteggio iniziale S_{0P}

(equazione 3.11), pari alla rispettiva distanza perpendicolare dal punto, definita in modo analogo al caso precedente (paragrafo 6):

$$d(Q_i, \overrightarrow{AB}) = \begin{cases} d_e(Q_i, Q_i') & \text{if } Q_i' \in [\overrightarrow{AB}] \\ \min \{d_e(Q_i, A); d_e(Q_i, B)\} & \text{elsewhere} \end{cases} \quad (3.10)$$

Con Q_i' proiezione del punto Q_i sull'arco AB . Se essa ricade sull'arco stesso, la distanza perpendicolare coincide con la distanza euclidea d_e tra il punto Q_i e la sua proiezione Q_i' ; in caso contrario, la distanza perpendicolare è definita come il minimo tra le distanze euclidee tra il punto Q_i ed i punti estremi dell'arco A e B (Fig. 3.28).

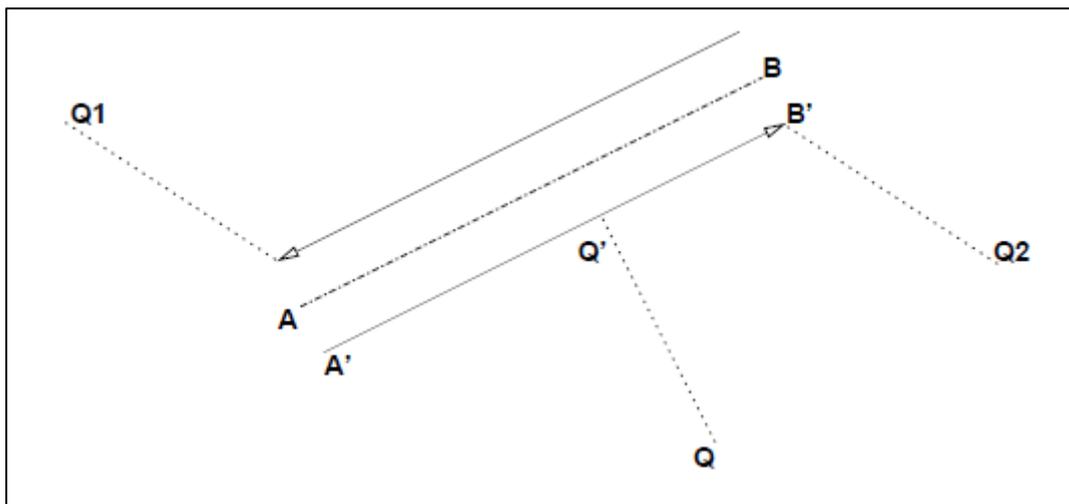


Figura 3.28: distanza perpendicolare dei punti Q , Q_1 e Q_2 (Marchal F. et al., 2004)¹¹⁷

$$S_{0P} = d(Q_1, E_1) \quad (3.11)$$

¹¹⁷ Marchal F. et al., 2004.

2) Avanzamento. Ad ogni punto Q_i successivo al primo, in ordine sequenziale cronologico, per ogni percorso P mantenuto in memoria vengono eseguite le seguenti operazioni:

2.1) Assunto che Q_{i-1} sia stato assegnato all'ultimo arco E_P del percorso P , si controlla se è stata raggiunta l'intersezione seguente o il nodo finale di E_P . Nella presente versione si considera tale condizione soddisfatta se è verificata almeno una delle due condizioni seguenti:

- la sommatoria delle distanze tra i punti GPS assegnati ad uno stesso arco E_P copre una certa percentuale della lunghezza dell'arco stesso (Fig. 3.29a); tale percentuale (α) è un parametro dell'algoritmo. Espressa in termini matematici, la condizione è la seguente:

$$\sum_{k=p1}^i d_e(Q_{k-1}, Q_k) > \alpha \cdot L(E_P) \quad (3.12)$$

con Q_{p0} primo punto assegnato all'arco E_P , $L(E_P)$ lunghezza dell'arco E_P , ed α parametro da stabilire. Se la condizione è verificata, si assume che l'intersezione è stata superata.

- la proiezione Q_i' del punto GPS processato Q_i sull'arco E_P - associato al punto precedente Q_{i-1} - ricade al di fuori dell'arco stesso (Fig. 3.29b), ovvero se

$$Q_i' \notin [E_P]$$

Se la condizione è verificata, si assume che l'intersezione è stata superata.

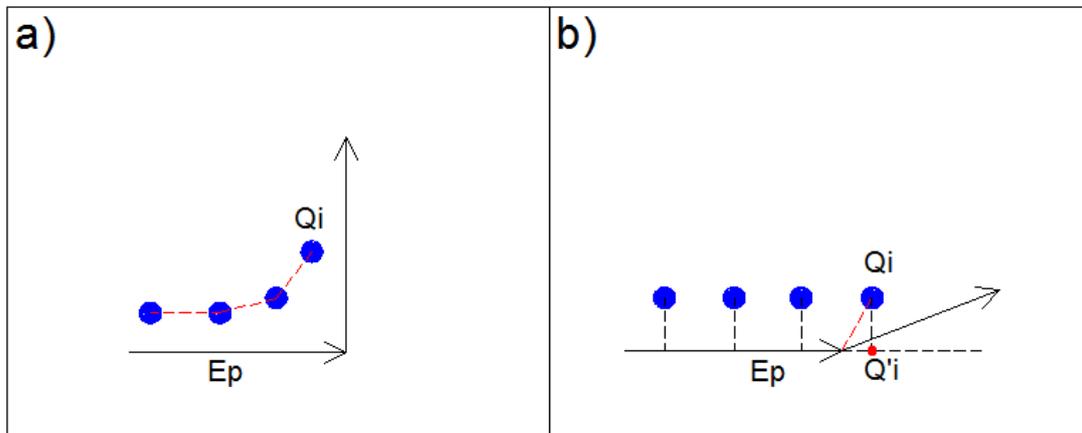


Figura 3.29: condizioni di superamento dell'intersezione

- 2.2) Nel caso in cui almeno una delle due condizioni di cui al punto 2.1 sia verificata, mediante una ricerca topologica degli archi uscenti dal nodo raggiunto, vengono creati ed aggiunti in memoria una serie di nuovi percorsi (*children paths*) P_{FS} , tanti quanti sono gli archi uscenti dal nodo finale di E_p . Essi differiscono dall'originario percorso P poiché contengono un arco in più, uscente dal nodo finale di E_p e diverso per ogni P_{FS} ; in altre parole, essi rappresentano le alternative di percorso, una volta raggiunta l'intersezione (Fig. 3.31). Il punto Q_i viene quindi assegnato all'arco E_{P+1} per ogni percorso P_{FS} .

Nel caso in cui nessuna delle due condizioni si verifichi, il punto Q_i è assegnato all'ultimo arco E_p del percorso P , e viene semplicemente aggiornato il punteggio del percorso P , come al punto successivo 2.3.

- 2.3) Lo score di ogni percorso, compresi quelli eventualmente creati, viene aggiornato tramite l'equazione seguente:

$$S_P = S_{0P} + \sum_{j=1}^P \sum_{i=2}^N d_{ps}(Q_i, E_j) \cdot \delta_{ij} \cdot \left| \sin(\angle(\overrightarrow{Q_{i-1}Q_i}, \vec{E}_j)) \right| \quad (3.13)$$

con d_{ps} *pseudo distanza* tra il punto Q_i e l'arco E_j , $\delta_{ij} = 1$ se Q_i è associato all'arco E_j e $\delta_{ij} = 0$ altrimenti. L'angolo $(Q_{i-1}, Q_i; E_j)$ è formato dal segmento congiungente il precedente punto Q_{i-1} con il punto processato Q_i e l'arco E_j (Fig. 3.30); se esso è un angolo orientato ottuso, ovvero minore di $-\frac{\pi}{2}$ o maggiore di $\frac{\pi}{2}$, viene convertito rispettivamente in $-\frac{\pi}{2}$ e $\frac{\pi}{2}$, in cui il valore della funzione $|\sin(x)|$ è pari a 1, il suo massimo (Fig. 3.32).

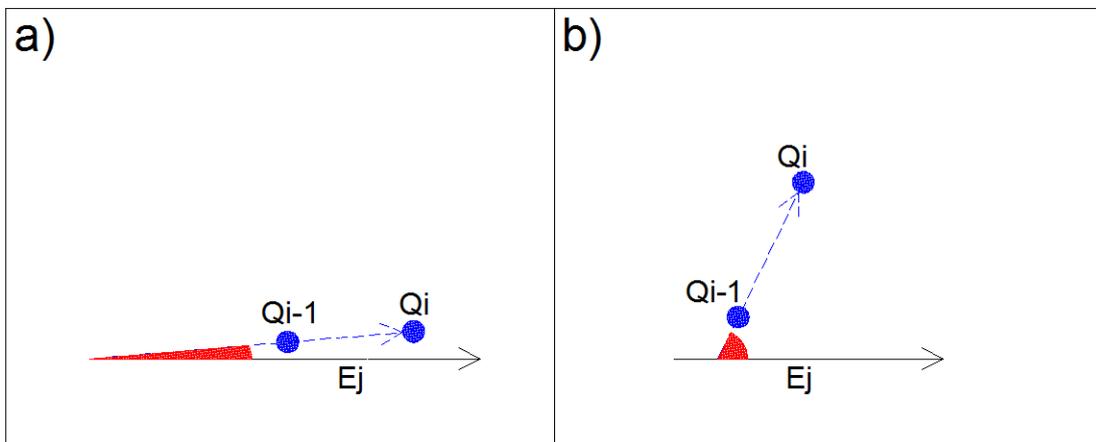


Figura 3.30: angolo formato dal segmento congiungente il precedente punto Q_{i-1} con il punto processato Q_i e l'arco E_j . In a) essi sono convergenti e l'angolo tende a 0; in b) essi sono divergenti e l'angolo tende a $\frac{\pi}{2}$.

Il percorso P è formato da una successione di archi $\{ E_1, E_2, \dots, E_j, \dots, E_p \}$, ed il set di punti GPS $\{ Q_1, Q_2, \dots, Q_i, \dots, Q_N \}$ è formato da N punti.

La *pseudo distanza* introdotta tra il generico punto Q_i ed il generico arco E_j è definita come segue:

$$d_{ps}(Q_i, E_j) = \max \{ d_0; d(Q_i, E_j) \} \quad (3.14)$$

con d distanza perpendicolare definita come sopra (equazione 3.10), e d_0 distanza base (*base distance*); essa rappresenta una distanza minima da considerare tra il punto e l'arco, al fine di evitare sovrapposizioni errate (Fig. 3.33, Fig. 3.34)

- 2.4) I percorsi vengono ordinati in base al loro punteggio e vengono eliminati quelli aventi i peggiori punteggi (i più alti), in relazione al numero di percorsi da mantenere in memoria, il quale rappresenta un altro parametro dell'algoritmo (*n_routes_follow*).

Al termine della sequenza di punti, il percorso che minimizza lo score è scelto come itinerario effettivamente seguito, dopo aver superato un'ultima verifica: è infatti controllato che esso sia arrivato a destinazione, ovvero che l'ultimo arco dell'itinerario ricada almeno in parte all'interno di un buffer circolare, di raggio 50 metri, centrato nell'ultimo punto GPS tracciato, similmente a quanto fatto per il primo punto GPS. Ciò al fine di evitare di fornire risultati errati.

Nelle figure seguenti è esemplificato il procedimento dell'algoritmo (Fig. 3.31), e ne sono approfonditi alcuni punti riguardanti il calcolo del punteggio (Fig. 3.32) e la *base distance* (Fig. 3.33).

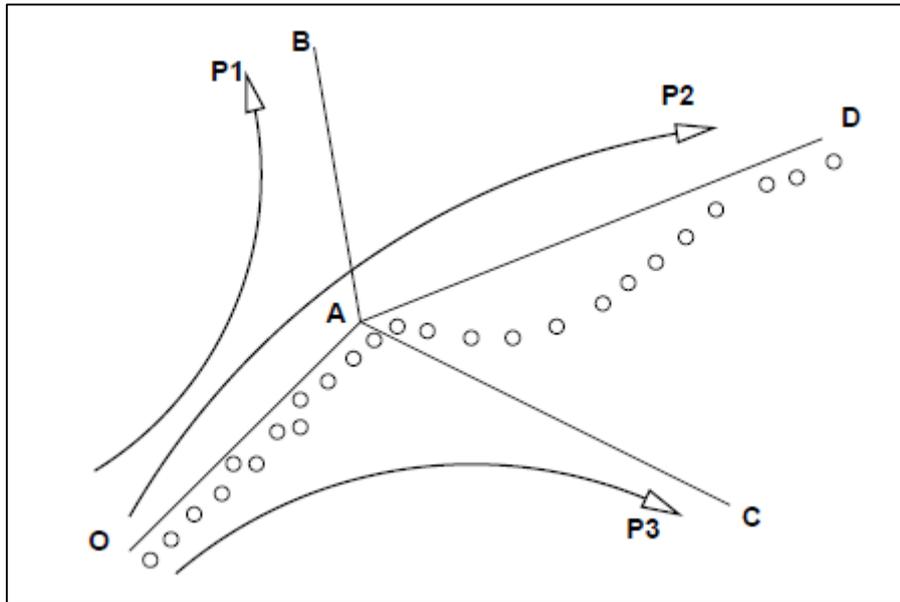
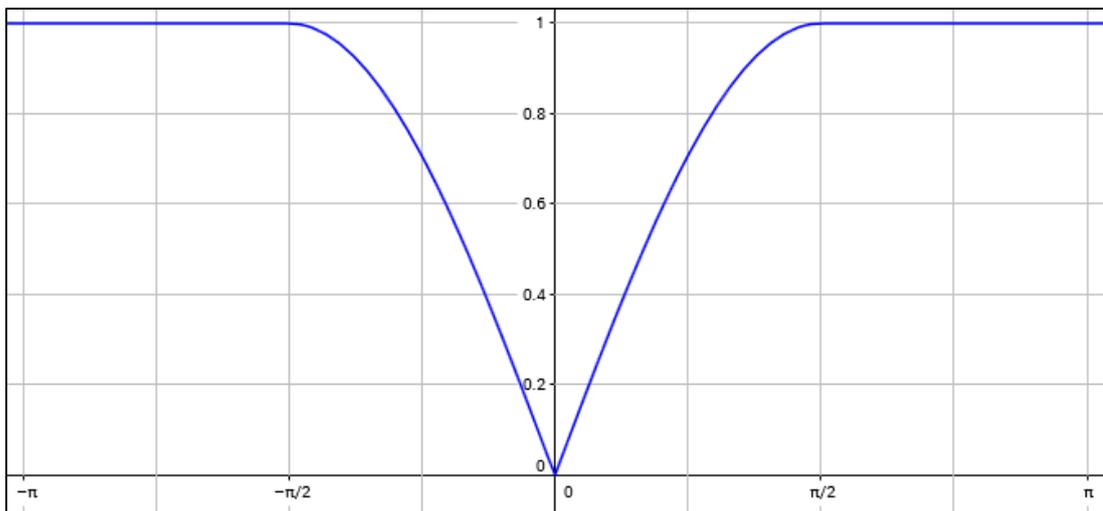


Figura 3.31: creazione dei tre *children paths* (P1, P2, P3) all'intersezione (nodo A), corrispondenti ai tre archi uscenti dal nodo A (AB, AD, AC); l'uso della tecnica delle ipotesi multiple ed il mantenimento in memoria dei possibili percorsi permette di scegliere il percorso esatto (P2), sebbene dai primi punti GPS dopo l'intersezione lo score migliore sia quello del percorso P3

La funzione seno, presa in valore assoluto, è stata inserita nella formula per il calcolo del punteggio (equazione 3.13) con lo scopo di premiare il parallelismo tra il segmento che unisce due punti GPS successivi e l'arco, e di penalizzare la perpendicolarità tra i medesimi elementi, in quanto si presuppone che quest'informazione sia indicativa della direzione di viaggio reale dell'utente.

L'assunto di partenza è infatti che quanto più essi sono paralleli, tanto più è probabile che l'itinerario attraversi proprio tale arco. Difatti, nel caso in cui l'angolo preso in considerazione sia prossimo a 0, e quindi i punti GPS e l'arco viaggino in direzione pressoché parallela, la funzione restituisce un valore vicino a 0, minimizzando il punteggio del punto relativo all'arco stesso; nel caso in cui, invece, l'angolo sia prossimo a $\frac{\pi}{2}$, quindi il segmento collegante i punti GPS e l'arco siano divergenti -

essendo all'incirca perpendicolari tra loro -, la funzione restituisce un valore prossimo a 1, massimizzando il punteggio. Nei casi intermedi tra 0 e $\frac{\pi}{2}$ e tra 0 e $-\frac{\pi}{2}$ la funzione assume valori crescenti compresi tra 0 e 1 , allontanandosi progressivamente dalla situazione di parallelismo. Infine, per valori dell'angolo maggiori di $\frac{\pi}{2}$ o minori di $-\frac{\pi}{2}$, la funzione è posta pari a 1 , il suo valore massimo, dato che la traiettoria dei punti GPS tende ad allontanarsi in direzione opposta a quella dell'arco. Nella figura seguente è mostrato l'andamento della funzione utilizzata per valutare la similitudine tra la traiettoria dei punti GPS e l'arco (Fig. 3.32).



$$f(x) = \begin{cases} |\sin(x)| & : \left(-\frac{\pi}{2} < x\right) \wedge \left(x < \frac{\pi}{2}\right) \\ 1 & : \text{altrove} \end{cases}$$

Figura 3.32: andamento della funzione $|\sin(x)|$ in $[-\frac{\pi}{2}; \frac{\pi}{2}]$; 1 per gli altri valori dell'angolo x

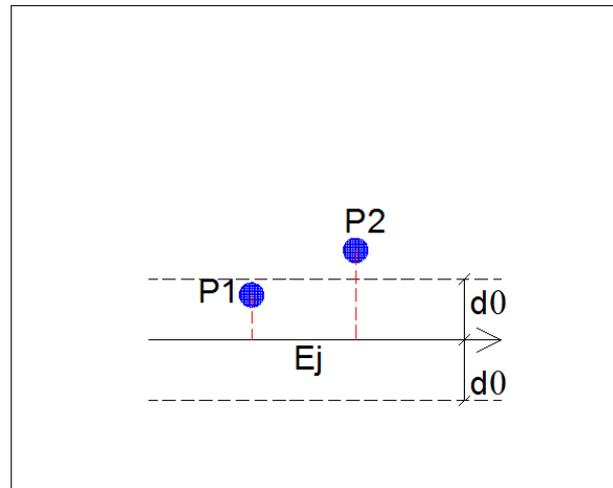


Figura 3.33: *base distance* d_0 . Il punto P1 ha una distanza da E_j $d(P1, E_j) < d_0$, per cui ha *pseudo distanza* $d_{ps}(P1, E_j) = d_0$. Il punto P2 invece ha una distanza $d(P2, E_j) > d_0$, per cui $d_{ps}(P2, E_j) = d(P2, E_j)$.

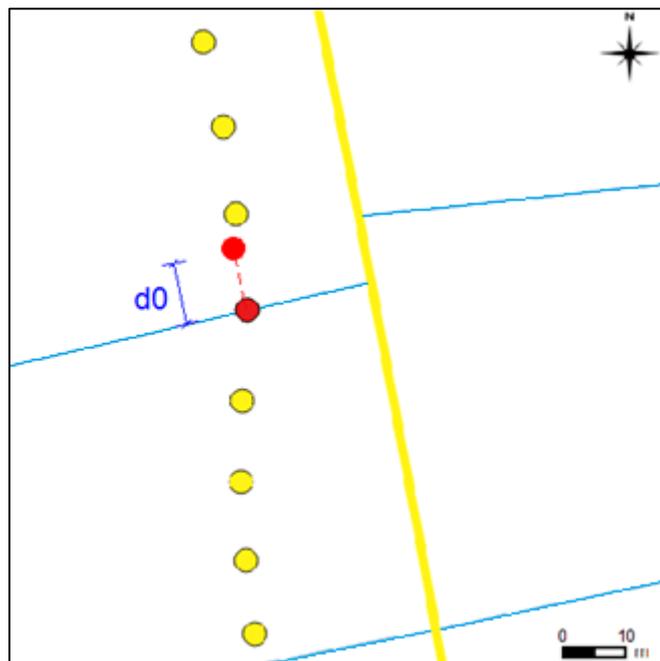


Figura 3.34: senza la *base distance* il punto evidenziato in rosso ricadrebbe su un arco errato, avendo da esso una distanza praticamente nulla, e in tal modo minimizzerebbe

lo score di un percorso errato che devia su tale arco non corretto (il percorso reale è in giallo). Applicando la *base distance*, invece, al punto evidenziato si impone una distanza dall'arco considerato pari alla *base distance* stessa.

7.1 ANALISI DEI PARAMETRI

Al fine di individuare il valore ottimale dei parametri, sono stati eseguiti diversi test, i cui risultati sono stati confrontati tramite i valori del *Length Index*, degli errori di distanza (*error distance*), del numero di matches (*successful matches*) e del tempo impiegato. In questo caso perde di significato il *Match Index*, in quanto scompare il concetto di buffer attorno agli archi, preso in considerazione nell'approccio metodologico degli algoritmi 'orig' e 'new'.

I tre parametri analizzati sono la distanza base d_0 (*base distance*), il numero di percorsi mantenuti in memoria $n_routes\ follow$ e la percentuale α della lunghezza dell'arco da coprire al fine di considerare superata l'intersezione o il nodo finale dell'arco.

Per lo studio del parametro d_0 sono stati eseguiti quattro test sul database Endomondo, utilizzando quattro diversi valori, rispettivamente di 3, 10, 15 e 20 metri; tali valori sono stati scelti in quanto si è ipotizzato di applicare ai punti una distanza minima dapprima di una corsia stradale (3 metri) ed in seguito di ordine di grandezza comparabile all'errore GPS (10, 15, 20 metri). Teoricamente d_0 dovrebbe dipendere dalla precisione dei dati GPS e dal tipo di rete: in una zona urbana ad alta densità di rete o con dei dati di posizionamento poco precisi, essa dovrebbe assumere dei valori maggiori, mentre in zone poco dense o con dati di posizionamento ad alta precisione dovrebbe assumere dei valori minori. Di seguito è riportata una tabella riassuntiva dei risultati ottenuti.

	d_0	3	d_0	10	d_0	15	d_0	20
Successful matches	7041	64,2%	7022	64,0%	7016	64,0%	6931	63,2%
Mean matching time [seconds]	0,643 s		0,629 s		0,642 s		0,640 s	
Mean Distance Error	17,2 m/point		17,2 m/point		17,3 m/point		17,4 m/point	
Standard deviation distance error	26,3 m/point		26,2 m/point		26,1 m/point		26,0 m/point	
CV distance error	1,53		1,52		1,50		1,49	
50%percentile distance error	9,4 m/point		9,6 m/point		9,8 m/point		9,9 m/point	
75%percentile distance error	16,8 m/point		16,8 m/point		17,0 m/point		17,2 m/point	
90%percentile distance error	35,1 m/point		35,0 m/point		34,7 m/point		35,0 m/point	
95%percentile distance error	56,6 m/point		56,6 m/point		56,5 m/point		55,9 m/point	
Mean Length Index	1,258		1,251		1,247		1,243	
Standard deviation length index	0,389		0,389		0,388		0,386	
CV length index	0,309		0,311		0,311		0,311	
50%percentile length index	1,140		1,130		1,130		1,130	
75%percentile length index	1,290		1,280		1,280		1,280	
80%percentile length index	1,350		1,330		1,330		1,320	
90%percentile length index	1,540		1,510		1,505		1,490	
95%percentile length index	1,910		1,910		1,912		1,870	
Length index in range [0,80 - 1,20]	4448	40,5%	4524	41,2%	4566	41,6%	4535	41,3%
Total distance travelled [km]	12395,3 km		12268,6 km		12276,9 km		12129,9 km	
Total distance travelled on bikeways [km]	2354,5	19,0%	2269,5	18,5%	2275,3	18,5%	2218,7	18,3%
Mean distance travelled on bikeways	334,4 m/route		323,2 m/route		324,3 m/route		320,1 m/route	
Routes containing bikeways	3034	27,7%	2978	27,1%	2970	27,1%	2889	26,3%
Loading time [seconds]	426,1 s		533,6 s		409,8 s		566,7 s	
Processing data time [seconds]	9250,1 s		9556,5 s		9123,7 s		9606,1 s	
Total execution time [seconds]	9676,2 s		10090,1 s		9533,4 s		10172,8 s	

Tabella 3.14: risultati dei test sul database Endomondo, algoritmo ‘NewMarchal’, al variare di d_0

Dalla tabella riassuntiva emerge immediatamente come i risultati siano comparabili per tutti i valori di d_0 : il numero di matches (*Successful matches*, attorno al 64%), i valori medi (circa 1,25) e percentili del *Length Index*, con poco più del 40% di matches aventi valori accettabili di *Length Index*; i valori medi (circa 17 m/punto) e percentili degli errori di distanza (*distance error*), dell’ordine delle decine di metri, lo stesso ordine di grandezza dell’errore GPS. Infine, anche il valore del tempo medio di matching (*Mean matching time*), poco meno di 0,65 secondi/traccia, corrispondente a poco più di 2 h e mezza, non presenta variazioni significative.

In definitiva è stato scelto un valore di *base distance* d_0 pari a 10 metri, in modo da mantenere una distanza minima comparabile all’errore base del GPS.

Per lo studio del parametro *n_routes follow*, sicuramente il più influente dal punto di vista del costo computazionale, sono stati eseguiti quattro test sul database Endomondo, utilizzando quattro diversi valori, compresi tra 10 e 40. Di seguito è riportata una tabella riassuntiva dei risultati ottenuti.

	<i>n_routes follow</i>	10	<i>n_routes follow</i>	20	<i>n_routes follow</i>	30	<i>n_routes follow</i>	40
Successful matches	7022	64,0%	7242	66,0%	7342	66,9%	7381	67,3%
Mean matching time [seconds]	0,629 s		0,910 s		1,162 s		1,527 s	
Matches/matching time differencial			783 s ⁻¹		397 s ⁻¹		107 s ⁻¹	
Mean Distance Error	17,2 m/point		17,3 m/point		17,2 m/point		17,2 m/point	
Standard deviation distance error	26,2 m/point		25,3 m/point		25,1 m/point		25,0 m/point	
CV distance error	1,52		1,46		1,46		1,45	
50%percentile distance error	9,6 m/point		9,7 m/point		9,7 m/point		9,7 m/point	
75%percentile distance error	16,8 m/point		17,2 m/point		17,2 m/point		17,1 m/point	
90%percentile distance error	35,0 m/point		35,3 m/point		34,9 m/point		35,5 m/point	
95%percentile distance error	56,6 m/point		56,0 m/point		55,4 m/point		55,4 m/point	
Mean Length Index	1,251		1,245		1,242		1,241	
Standard deviation length index	0,389		0,384		0,381		0,380	
CV length index	0,311		0,308		0,307		0,307	
50%percentile length index	1,130		1,130		1,130		1,130	
75%percentile length index	1,280		1,280		1,280		1,270	
80%percentile length index	1,330		1,330		1,320		1,320	
90%percentile length index	1,510		1,500		1,490		1,490	
95%percentile length index	1,910		1,890		1,879		1,870	
Length index in range [0,80 - 1,20]	4524	41,2%	4715	43,0%	4812	43,9%	4854	44,2%
Total distance travelled [km]	12268,6 km		12805,3 km		12997,1 km		13081,1 km	
Total distance travelled on bikeways [km]	2269,5	18,5%	2369,0	18,5%	2418,6	18,6%	2436,9	18,6%
Mean distance travelled on bikeways	323,2 m/route		327,1 m/route		329,4 m/route		330,2 m/route	
Routes containing bikeways	2978	27,1%	3127	28,5%	3177	29,0%	3182	29,0%
Loading time [seconds]	533,6 s		565,5 s		402,3 s		464,7 s	
Processing data time [seconds]	9556,5 s		12206,0 s		14288,4 s		18761,5 s	
Total execution time [seconds]	10090,1 s		12771,6 s		14690,7 s		19226,3 s	

Tabella 3.15: risultati dei test sul database Endomondo, algoritmo ‘NewMarchal’, al variare di *n_routes follow*

Dalla tabella riassuntiva si può notare che il numero di matches (*Successful matches*) è del tutto simile al variare del parametro (64 - 67%); anche i valori medi e percentili dei *Length index* e degli errori (*distance error*) presentano dei valori comparabili, senza variazioni significative. L’effetto più importante della scelta del numero di percorsi da mantenere in memoria è invece sul tempo medio di matching (*mean matching time*): esso mostra un valore decisamente basso, che ovviamente aumenta all’aumentare di *n_routes follow*, dovendosi compiere un maggior numero di operazioni, passando da un valore di 0,629 secondi/traccia (n = 10), corrispondenti a quasi 3 h totali, a 1,527 secondi/traccia (n = 40), corrispondenti a circa 5 h e mezza,

rivelando anche in questo caso una tendenza approssimativamente di tipo lineare (Fig. 3.35).

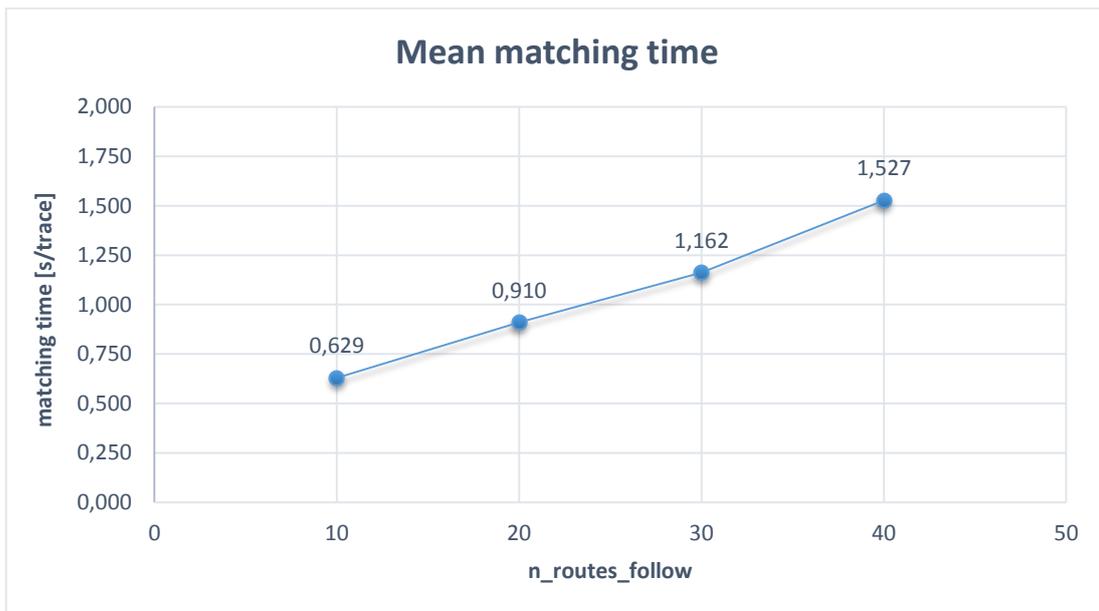


Figura 3.35: tempo medio di matching al variare del numero di percorsi in memoria

In figura 3.36 è mostrato il rapporto incrementale tra il numero dei matches riusciti (*successful matches*) ed il tempo medio di matching (*mean matching time*), al variare di *n_routes_follow*. Sulla base di tale rapporto, si è stabilito che il valore di ottimale è pari a 20. Tuttavia, anche in questo caso si potrebbe considerare che l'incremento del numero di matches riusciti, del tutto marginale (5,1% massimo, per $n = 40$, rispetto al caso in cui $n = 10$), non sia tale da giustificare l'adozione di un *n_routes_follow* maggiore, che genera invece tempi di esecuzione molto più elevati (aumento del 91%, quindi tempi quasi raddoppiati, per $n = 40$, rispetto a $n = 10$).

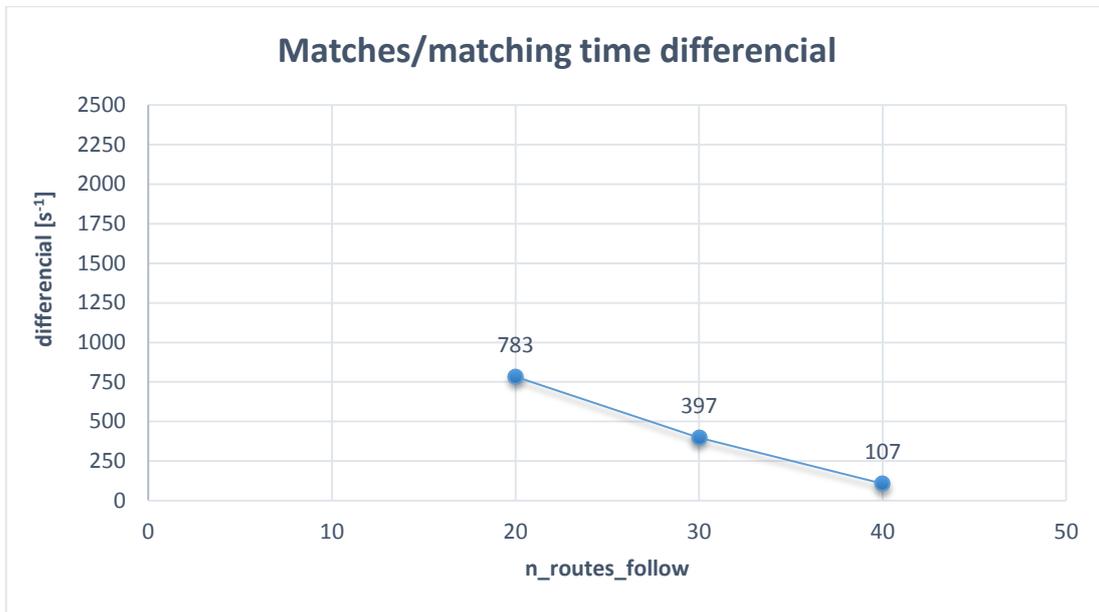


Figura 3.36: rapporto incrementale tra il numero di matches riusciti ed il tempo medio di matching, al variare del numero di percorsi mantenuto in memoria

Per lo studio del parametro α , determinante per stabilire se l'intersezione o il nodo finale dell'arco siano stati raggiunti, e di conseguenza per la ricostruzione del percorso, sono stati eseguiti quattro test sul database Endomondo, utilizzando quattro diversi valori, compresi tra 0,3 e 0,9. Di seguito è riportata una tabella riassuntiva dei risultati ottenuti.

	α	0,3	α	0,5	α	0,7	α	0,9
Successful matches	7202	65,6%	7242	66,0%	7213	65,7%	7225	65,9%
Mean matching time [seconds]	0,810 s		0,910 s		0,936 s		0,958 s	
Mean Distance Error	17,3 m/point		17,3 m/point		17,0 m/point		16,7 m/point	
Standard deviation distance error	25,8 m/point		25,3 m/point		24,9 m/point		24,3 m/point	
CV distance error	1,49		1,46		1,47		1,45	
50%percentile distance error	9,7 m/point		9,7 m/point		9,6 m/point		9,5 m/point	
75%percentile distance error	17,1 m/point		17,2 m/point		16,9 m/point		16,4 m/point	
90%percentile distance error	35,2 m/point		35,3 m/point		34,5 m/point		34,5 m/point	
95%percentile distance error	54,8 m/point		56,0 m/point		54,4 m/point		53,6 m/point	
Mean Length Index	1,298		1,245		1,201		1,167	
Standard deviation length index	0,484		0,384		0,336		0,317	
CV length index	0,373		0,308		0,280		0,272	
50%percentile length index	1,160		1,130		1,100		1,070	
75%percentile length index	1,330		1,280		1,230		1,190	
80%percentile length index	1,380		1,330		1,280		1,230	
90%percentile length index	1,600		1,500		1,440		1,380	
95%percentile length index	2,030		1,890		1,710		1,570	
Length index in range [0,80 - 1,20]	4220	38,5%	4715	43,0%	5135	46,8%	5527	50,4%
Total distance travelled [km]	13088,0 km		12805,3 km		12547,1 km		12219,2 km	
Total distance travelled on bikeways [km]	2447,8	18,7%	2369,0	18,5%	2290,7	18,3%	2220,4	18,2%
Mean distance travelled on bikeways	339,9 m/route		327,1 m/route		317,6 m/route		307,3 m/route	
Routes containing bikeways	3003	27,4%	3127	28,5%	3149	28,7%	3208	29,2%
Loading time [seconds]	410,6 s		565,5 s		483,0 s		392,2 s	
Processing data time [seconds]	10545,6 s		12206,0 s		12588,6 s		12339,1 s	
Total execution time [seconds]	10956,2 s		12771,6 s		13071,6 s		12731,3 s	

Tabella 3.16: risultati dei test sul database Endomondo, algoritmo ‘NewMarchal’, al variare di α

Dalla tabella riassuntiva è possibile notare che il numero di matches (*successful matches*) non varia significativamente, attestandosi attorno al 66%; i valori di *Length Index* e *distance error* sono comparabili, anche se come per l’algoritmo ‘Marchal’ vi è un leggero miglioramento per valori di α maggiori. Infine, anche il tempo di matching (*mean matching time*) non subisce delle variazioni significative.

In definitiva, per gli stessi motivi esposti al paragrafo 6.1 è stato ritenuto che il valore ottimale sia $\alpha = 0,7$. Infatti, valori troppo vicini all’unità forzano l’algoritmo ad attendere che venga coperta dai punti GPS l’intera lunghezza dell’arco prima di passare ai successivi, il che può causare errori notevoli, perdendo la corretta associazione tra il punto e l’arco; invece, valori vicini allo 0 suggeriscono che l’intersezione a fine arco sia stata raggiunta quasi immediatamente, e quindi conducono ad un notevole accumulo di percorsi diversi creati.

7.2 RISULTATI E CONSIDERAZIONI

Dopo aver realizzato l'analisi dei parametri, nella tabella riassuntiva che segue si riportano i risultati ottenuti eseguendo l'algoritmo di map-matching denominato 'NewMarchal' sul database di tracce Endomondo, posti i parametri d_0 pari a 10 metri, $n_routes\ follow$ pari a 20 e α pari a 0,7.

	'newMarchal'	
Traces number (cleaning)	10971	
Successful matches	7213	65,7%
Mean matching time [seconds]	0,936	s
Mean Distance Error	17,0	m/point
Standard deviation distance error	24,9	m/point
CV distance error	1,47	
50%percentile distance error	9,6	m/point
75%percentile distance error	16,9	m/point
90%percentile distance error	34,5	m/point
95%percentile distance error	54,4	m/point
Mean Length Index	1,201	
Standard deviation length index	0,336	
CV length index	0,280	
50%percentile length index	1,100	
75%percentile length index	1,230	
80%percentile length index	1,280	
90%percentile length index	1,440	
95%percentile length index	1,710	
Length index in range [0,80 - 1,20]	5135	46,8%
Total distance travelled [km]	12547,1	km
Total distance travelled on bikeways [km]	2290,7	18,3%
Mean distance travelled on bikeways	317,6	m/route
Routes containing bikeways	3149	28,7%
Loading time [seconds]	483,0	s
Processing data time [seconds]	12588,6	s
Total execution time [seconds]	13071,6	s

Tabella 3.17: risultati del map-matching dell'algoritmo 'NewMarchal' sul database Endomondo

Dalla tabella riassuntiva si può notare che i matches riusciti (*successful matches*) sono circa i $\frac{2}{3}$ del totale (65,7%), con un tempo medio di 0,936 secondi/traccia (*mean matching time*) ed un tempo totale di esecuzione di 3 h, 37 minuti e 51 secondi (*total execution time*). Anche con le modifiche apportate in questa versione perfezionata, l'aspetto del costo computazionale è indubbiamente il principale elemento a sostegno di tale metodologia, in quanto il numero di operazioni da eseguire è minore, e di conseguenza il tempo di esecuzione si attesta su valori molto più bassi rispetto alla metodologia degli algoritmi 'orig' e 'new'. I valori della media (17,0 metri/punto) e dei percentili degli errori di distanza (*distance error*) sono ottimi, dell'ordine di grandezza comparabile all'errore del GPS stesso; i valori dei *Length Index* sono distribuiti in modo molto concentrato attorno ad un valore medio di 1,2 (coefficiente di variazione di 0,280): difatti 5135 itinerari (46,8% del totale) presentano un indice compreso tra 0,8 e 1,2, in un range ritenuto accettabile.

Ciò è dovuto al punteggio (*score*) associato ad ogni percorso, il quale non dipende più esclusivamente dalla distanza punto - arco, in quanto è imposta una distanza minima (*base distance*) che limita l'associazione ad archi vicini; inoltre, in questa versione si è tenuto conto della similitudine direzionale tra la traiettoria dei punti GPS e l'arco stesso. In tal modo, gli errori esposti al paragrafo 6.2, che ricorrevano precedentemente eseguendo l'algoritmo di Marchal originale, in particolar modo nelle situazioni in cui vi era un incrocio con un arco bidirezionale, sono in buona parte evitati (Fig. 3.37). Pertanto, anche la lunghezza dell'itinerario non risente di tali errori e di conseguenza i *Length Index* si presentano molto più vicini a 1, identificando degli itinerari molto più rappresentativi.

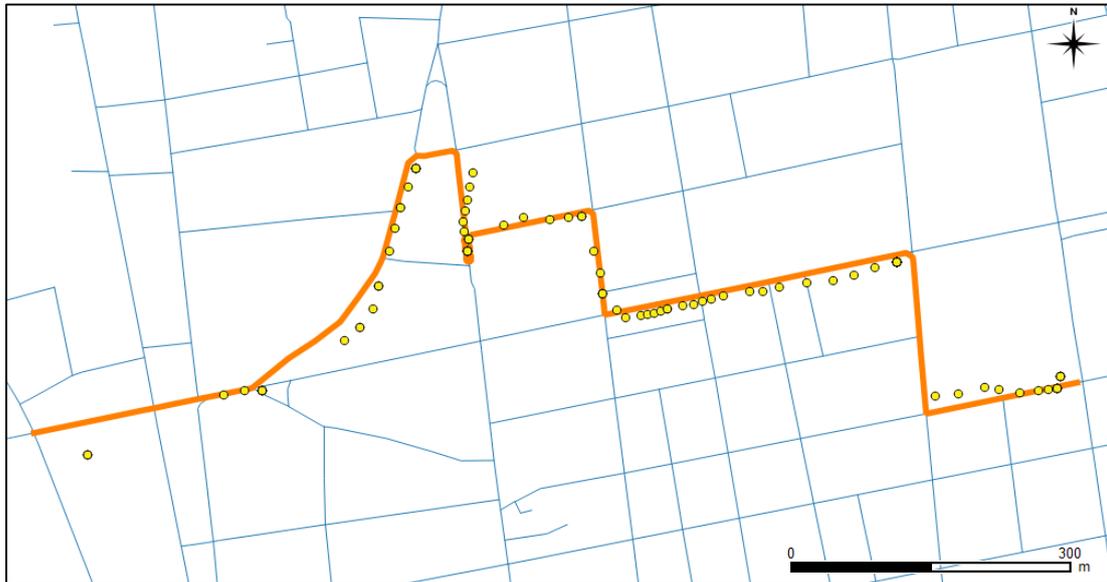


Figura 3.37: esempio di map-matching effettuato tramite l'algoritmo 'NewMarchal'; il percorso in figura è tratto dal campione raccolto ed è stato ricostruito sulla base degli stessi punti utilizzati in figura 3.24 e non presenta gli errori tipici dell'algoritmo 'Marchal' ($Length\ Index = 1,02$; $Real\ Length\ Index = 1,10$; $error\ distance = 6,4\ m/point$)

8. CONFRONTO TRA GLI ALGORITMI 'Marchal' E 'NewMarchal'

Dopo avere analizzato singolarmente gli algoritmi denominati 'Marchal' e 'NewMarchal', si procede ad una comparazione dei risultati ottenuti eseguendo il map-matching sul database di tracce Endomondo tramite i due algoritmi. Gli esiti dell'analisi si riportano nella tabella riassuntiva seguente.

	'Marchal'		'newMarchal'	
Traces number (cleaning)	10971		10971	
Successful matches	6491	59,2%	7213	65,7%
Mean matching time [seconds]	0,924	s	0,936	s
Mean Distance Error	12,6	m/point	17,0	m/point
Standard deviation distance error	20,5	m/point	24,9	m/point
CV distance error	1,62		1,47	
50%percentile distance error	7,9	m/point	9,6	m/point
75%percentile distance error	12,8	m/point	16,9	m/point
90%percentile distance error	23,5	m/point	34,5	m/point
95%percentile distance error	35,6	m/point	54,4	m/point
Mean Length Index	1,923		1,201	
Standard deviation length index	0,444		0,336	
CV length index	0,231		0,280	
50%percentile length index	1,880		1,100	
75%percentile length index	2,090		1,230	
80%percentile length index	2,170		1,280	
90%percentile length index	2,390		1,440	
95%percentile length index	2,680		1,710	
Length index in range [0,80 - 1,20]	136	1,2%	5135	46,8%
Total distance travelled [km]	17707,4	km	12547,1	km
Total distance travelled on bikeways [km]	3304,4	18,7%	2290,7	18,3%
Mean distance travelled on bikeways	509,1	m/route	317,6	m/route
Routes containing bikeways	3751	34,2%	3149	28,7%
Loading time [seconds]	415,1	s	483,0	s
Processing data time [seconds]	12575,6	s	12588,6	s
Total execution time [seconds]	12990,7	s	13071,6	s

Tabella 3.18: confronto tra i risultati del map-matching degli algoritmi 'Marchal' e 'NewMarchal' sul database Endomondo

L'algoritmo proposto, denominato 'NewMarchal', si presentava come possibile perfezionamento dell'algoritmo 'Marchal', in quanto, sfruttando la medesima metodologia, apportava dei correttivi, al fine di risolvere alcune problematiche di quest'ultimo, evidenziate nel paragrafo 6.2. In particolare, l'obiettivo perseguito era quello di aumentare il numero di matches riusciti e di migliorare la loro accuratezza, limitando considerevolmente le situazioni di matching errato in corrispondenza di incroci con strade bidirezionali mostrate al paragrafo 6.2 - che costituiscono i

principali errori che rendono poco significativi i risultati del matching dell'algoritmo 'Marchal' - senza incrementare eccessivamente i tempi di esecuzione, che rappresentano il più importante pregio di tale metodologia.

La tabella 3.18 suesposta evidenzia i miglioramenti nei risultati, ottenuti mediante le modifiche che sono state apportate: il numero di matches riusciti (*successful matches*, Fig. 3.38a) è incrementato da 6491 a 7213 (+11%) e la loro accuratezza è senza dubbio notevolmente migliorata, come attestano i valori della media e dei percentili (Fig. 3.38b) dei *Length Index*, tutti decisamente inferiori rispetto al caso precedente. In particolare, essi si distribuiscono adesso intorno ad un valore medio di 1,201, molto più vicino a 1 rispetto al valore medio precedente di 1,923, mostrando una dispersione limitata (coefficiente di variazione pari a 0,280); difatti, i matches aventi un valore di *Length Index* nel range ritenuto accettabile di 0,8 - 1,2 (Fig. 3.39) sono ben 5135 (46,8% del totale), contro i 136 precedenti (appena 1,2% del totale).

Nel contempo, il tempo medio di matching (*mean matching time*) rimane praticamente invariato, passando da 0,924 a 0,936 secondi/traccia (+1,3%), confermandosi così uno degli aspetti più vantaggiosi di questa metodologia. I valori medi e percentili dell'errore di distanza (*error distance*), invece, sono leggermente peggiori, ma ampiamente accettabili (il 75% dei matches riusciti presenta valori inferiori a 16,9 metri/punto, comparabili con l'errore del GPS stesso).

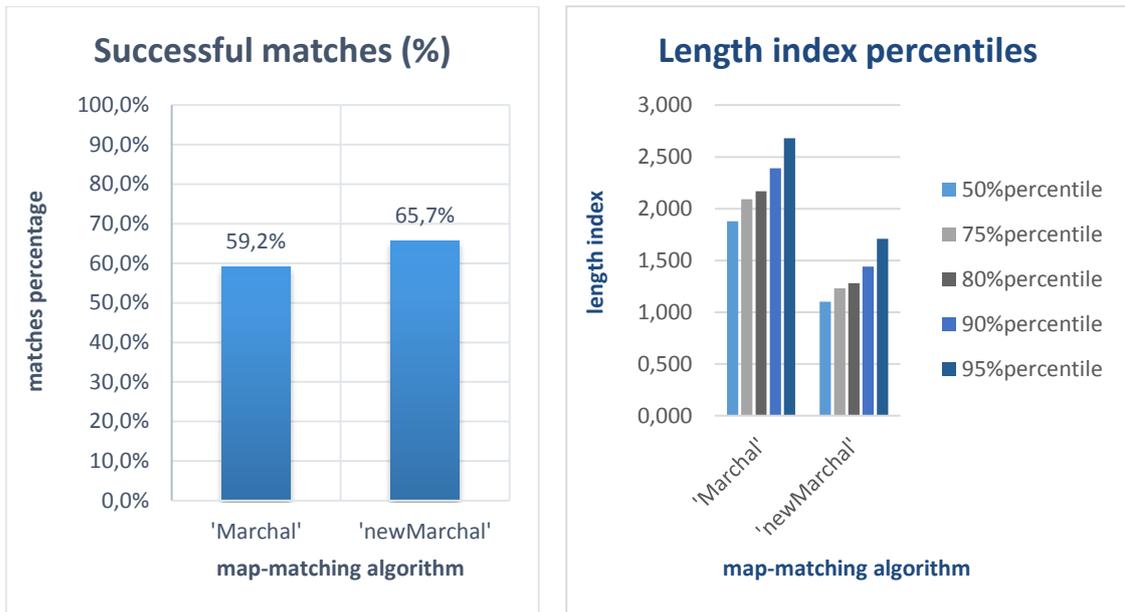


Figura 3.38: percentuale di matches riusciti (a) e percentili del *Length Index* (b)

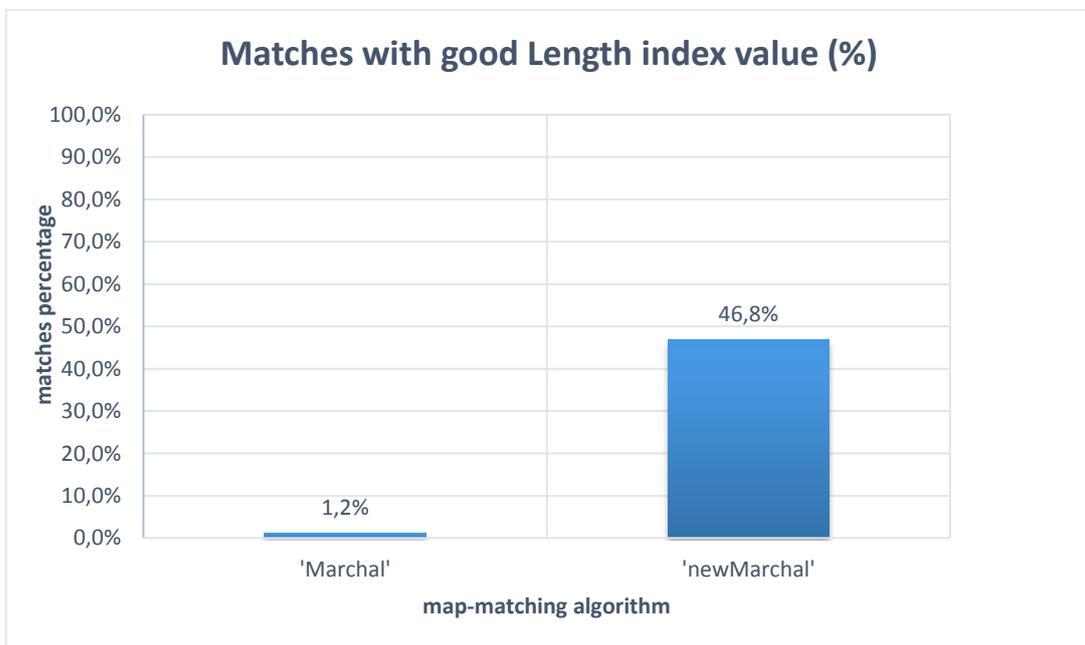


Figura 3.39: percentuale di matches aventi valori del *Length Index* che si considerano accettabili

In definitiva i risultati del map-matching eseguito mediante l'algoritmo 'NewMarchal' implementato hanno evidenziato un netto miglioramento, per cui gli obiettivi delle modifiche effettuate possono considerarsi raggiunti.

Per quanto riguarda il campione di tracce raccolte, i risultati hanno confermato le osservazioni condotte sul database Endomondo: difatti, l'accuratezza del matching è notevolmente migliorata, come evidenziato dai valori di *Length Index* e di *Real Length Index*. Nelle figure seguenti sono mostrati alcuni esempi, tratti dal campione raccolto, dei miglioramenti raggiunti e delle differenze nel matching tra i due algoritmi.

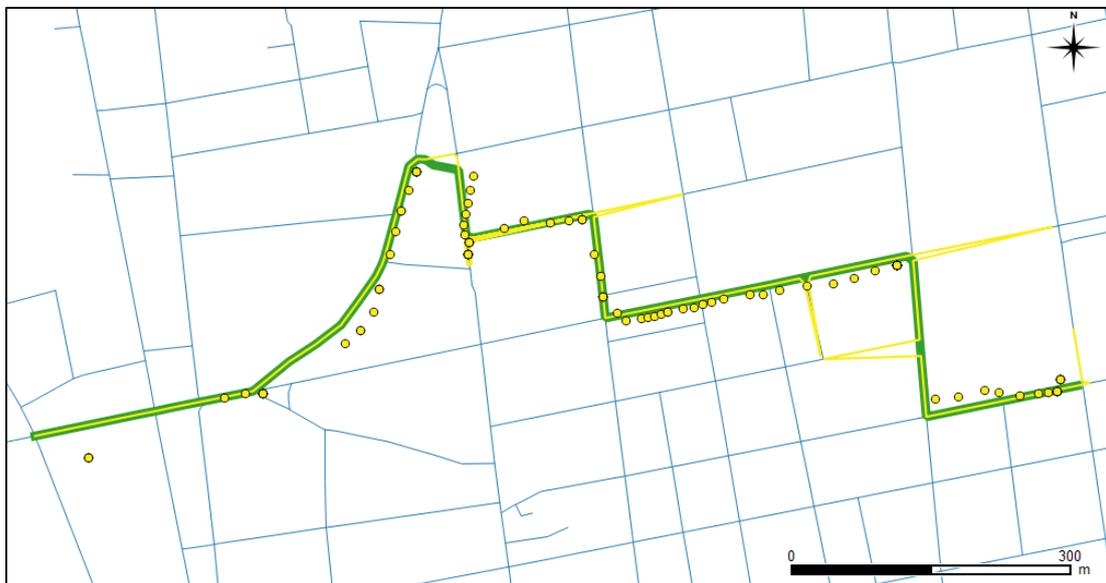


Figura 3.40 a): esempio di map-matching effettuato con l'algoritmo 'Marchal' (in giallo), sovrapposto al percorso reale (in verde), tratto dal campione raccolto; sono evidenti diversi errori (*Length Index* = 1,49; *Real Length Index* = 1,60; *error distance* = 5,7 m/point)



Figura 3.40 b): esempio di matching effettuato con l’algoritmo ‘NewMarchal’ (in arancione), sovrapposto al percorso reale (in verde), tratto dal campione raccolto; in questo caso il matching è quasi perfetto (*Length Index* = 1,02; *Real Length Index* = 1,10; *error distance* = 6,4 m/point)

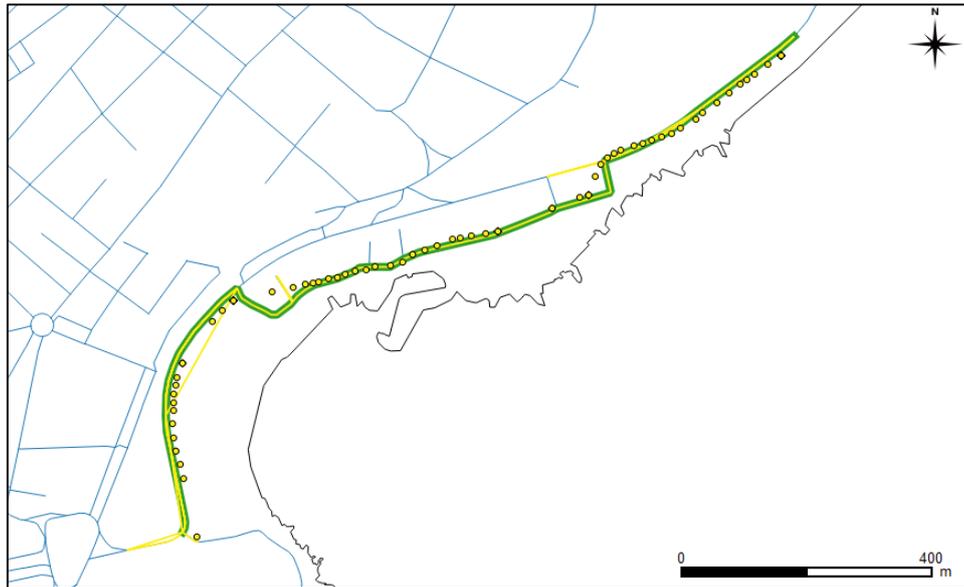


Figura 3.41 a): esempio di map-matching effettuato con l'algoritmo 'Marchal' (in giallo), sovrapposto al percorso reale (in verde), tratto dal campione raccolto; si possono notare alcuni errori ($Length\ Index = 1,71$; $Real\ Length\ Index = 1,73$; $error\ distance = 5,6\ m/point$)

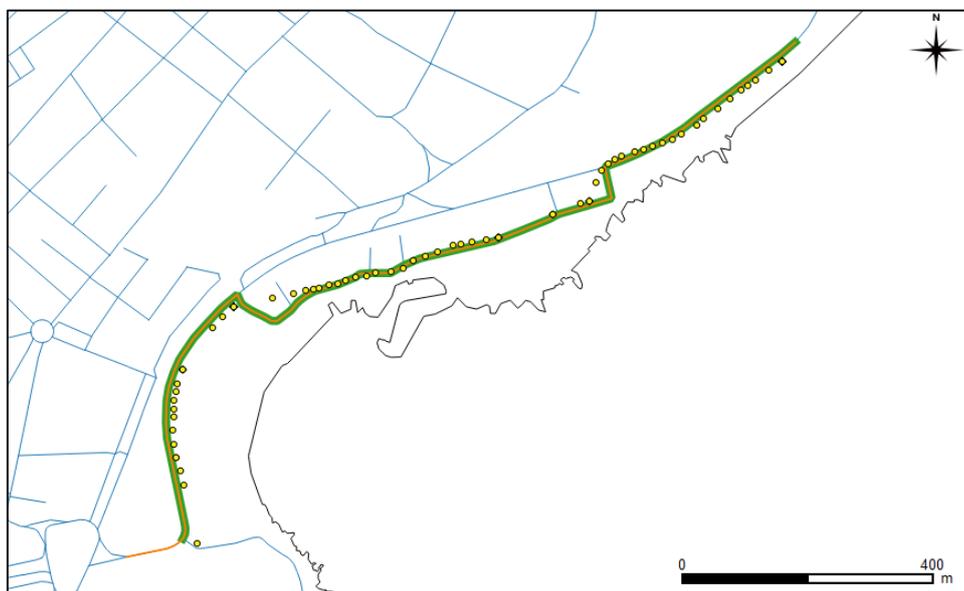


Figura 3.41 b): esempio di matching effettuato con l'algoritmo 'NewMarchal' (in arancione), sovrapposto al percorso reale (in verde), tratto dal campione raccolto; in

questo caso il matching è quasi perfetto (*Length Index* = 1,05; *Real Length Index* = 1,06; *error distance* = 6,8 m/point)

9. CONFRONTO GENERALE TRA GLI ALGORITMI

Dopo avere analizzato le metodologie e gli algoritmi denominati ‘orig’, ‘new’, ‘Marchal e ‘NewMarchal’, si procede ad una comparazione dei risultati ottenuti eseguendo il map-matching sul database di tracce Endomondo tramite gli algoritmi descritti in precedenza. Gli esiti dell’analisi si riportano nella tabella riassuntiva seguente.

	'orig'		'new'		'Marchal'		'newMarchal'	
Traces number (cleaning)	5969		5969		10971		10971	
Successful matches	5284	88,5%	5266	88,2%	6491	59,2%	7213	65,7%
Mean matching time [seconds]	2,612 s		2,130 s		0,924 s		0,936 s	
Mean Distance Error	63,4 m/point		73,5 m/point		12,6 m/point		17,0 m/point	
Standard deviation distance error	196,8 m/point		213,1 m/point		20,5 m/point		24,9 m/point	
CV distance error	3,10		2,90		1,62		1,47	
50%percentile distance error	14,1 m/point		16,2 m/point		7,9 m/point		9,6 m/point	
75%percentile distance error	28,5 m/point		36,1 m/point		12,8 m/point		16,9 m/point	
90%percentile distance error	111,5 m/point		149,3 m/point		23,5 m/point		34,5 m/point	
95%percentile distance error	286,5 m/point		353,5 m/point		35,6 m/point		54,4 m/point	
Mean Length Index	0,988		0,963		1,923		1,201	
Standard deviation length index	0,281		0,301		0,444		0,336	
CV length index	0,285		0,312		0,231		0,280	
50%percentile length index	0,960		0,950		1,880		1,100	
75%percentile length index	1,070		1,060		2,090		1,230	
80%percentile length index	1,110		1,100		2,170		1,280	
90%percentile length index	1,230		1,230		2,390		1,440	
95%percentile length index	1,400		1,380		2,680		1,710	
Mean Match Index	0,816		0,787					
Standard deviation match index	0,225		0,241					
CV match index	0,276		0,307					
5%percentile match index	0,290		0,220					
10%percentile match index	0,470		0,390					
20%percentile match index	0,710		0,640					
25%percentile match index	0,760		0,710					
50%percentile match index	0,890		0,870					
Length index in range [0,80 - 1,20]	4040	67,7%	3883	65,1%	136	1,2%	5135	46,8%
Match index in range [0,80 - 1,00]	3765	63,1%	3403	57,0%				
Total distance travelled [km]	15660,5 km		15232,5 km		17707,4 km		12547,1 km	
Total distance travelled on bikeways [km]	3665,0	23,4%	3163,2	20,8%	3304,4	18,7%	2290,7	18,3%
Mean distance travelled on bikeways	693,6 m/route		600,7 m/route		509,1 m/route		317,6 m/route	
Routes containing bikeways	4243	71,1%	3774	63,2%	3751	34,2%	3149	28,7%
Loading time [seconds]	506,8 s		534,6 s		415,1 s		483,0 s	
Processing data time [seconds]	16741,2 s		14139,6 s		12575,6 s		12588,6 s	
Total execution time [seconds]	17248,0 s		14674,2 s		12990,7 s		13071,6 s	

Tabella 3.19: confronto tra i risultati del matching degli algoritmi ‘orig’, ‘new’, ‘Marchal e ‘NewMarchal’ sul database Endomondo

Come già descritto nel paragrafo 2, le due metodologie testate necessitano di una fase di filtraggio dei dati (*clean*) leggermente diversa; difatti, l’approccio ideato da Marchal, operando sequenzialmente sui punti, richiede altresì che essi non siano troppo distanti tra loro nel tempo e nello spazio, procedendosi in tal caso all’ulteriore suddivisione delle tracce. Ciò determina un numero di tracce nel database Endomondo

decisamente maggiore (10971 contro le 5969 dell'altro approccio) e, di conseguenza, un aumento del tempo impiegato nella fase di *cleaning*.

Il numero di matches riusciti (*successful matches*) è decisamente superiore nei primi due algoritmi (88%, contro circa il 60% degli altri due, Fig. 3.42), ma è da valutare quanti di questi siano considerabili esatti.

Difatti, i valori dei *Length Index* sono generalmente più bassi nei risultati degli algoritmi 'orig' e 'new', i cui itinerari sono in genere più brevi, giacché essi utilizzano l'algoritmo di Dijkstra per la ricerca del percorso più breve anche nel caso in cui non vi siano punti GPS tracciati. In entrambi i casi, circa il 66% dei matches ha un *Length Index* accettabile. Gli algoritmi che invece utilizzano la seconda metodologia ('Marchal' e 'NewMarchal') presentano invece dei *Length Index* maggiori. Tali valori sono inaccettabili nella versione originale 'Marchal' (solo l'1,2% dei matches ha un valore di *Length Index* accettabile), mentre sono molto più realistici nella versione aggiornata 'NewMarchal', che è stata proposta nel presente studio, i cui risultati evidenziano che quasi il 47% dei matches ha un *Length Index* in un range compreso tra 0,8 e 1,2, considerato accettabile. Un confronto grafico complessivo è mostrato in figura 3.43.

Per quanto riguarda l'errore di distanza (*distance error*), esso è mediamente più alto per i primi due algoritmi, anche se nel 75% dei casi è minore rispettivamente di 28,5 e 36,1 metri/punto; gli errori di distanza sono invece minori per gli altri due algoritmi, nel 90% dei casi comparabili all'errore di misurazione del GPS stesso.

I valori dei *Match Index* sono comparabili per gli algoritmi 'orig' e 'new', sia nei valori medi che nei percentili, e difatti il numero di matches aventi un *Match Index* accettabile, compreso tra 0,8 e 1, è sovrapponibile (circa il 60%); tale indice è al contrario privo di significato per gli algoritmi 'Marchal' e 'NewMarchal', in quanto essi non operano tramite buffers creati attorno agli archi.

Infine, il dato relativo al tempo impiegato per il matching (*mean matching time*) è quello che presenta maggiori discrepanze tra i vari algoritmi (Fig. 3.44). Difatti, il primo approccio mostra dei tempi notevolmente superiori, con 2,612 secondi/traccia per l'algoritmo 'orig' e 2,130 secondi/traccia per l'algoritmo 'new' - la maggior parte

dei quali spesi nella fase di attribuzione dei costi ad ogni arco della rete. Al contrario, il secondo approccio si dimostra essere molto più veloce, con circa 0,9 secondi/traccia impiegati per entrambi gli algoritmi 'Marchal' e 'NewMarchal'.

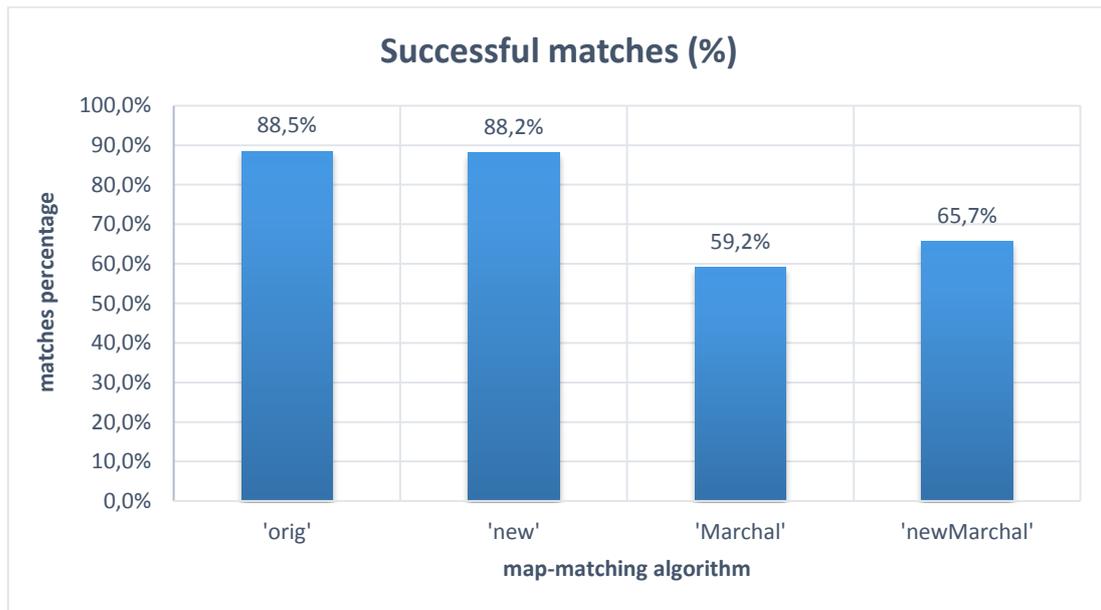


Figura 3.42: percentuale di matches riusciti

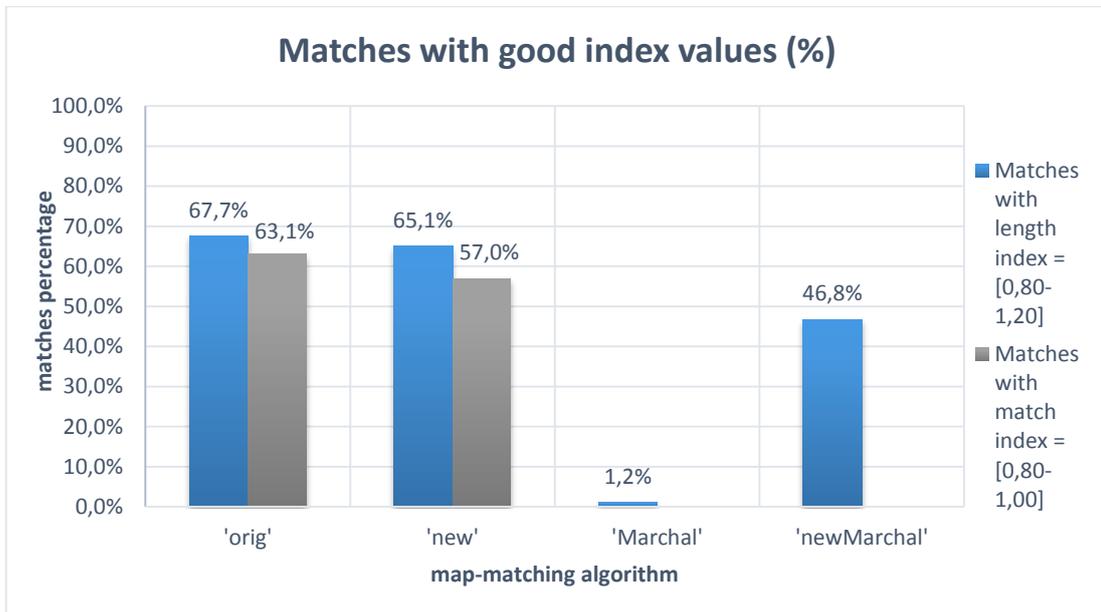


Figura 3.43: percentuale di matches i cui indici presentano dei valori ritenuti accettabili

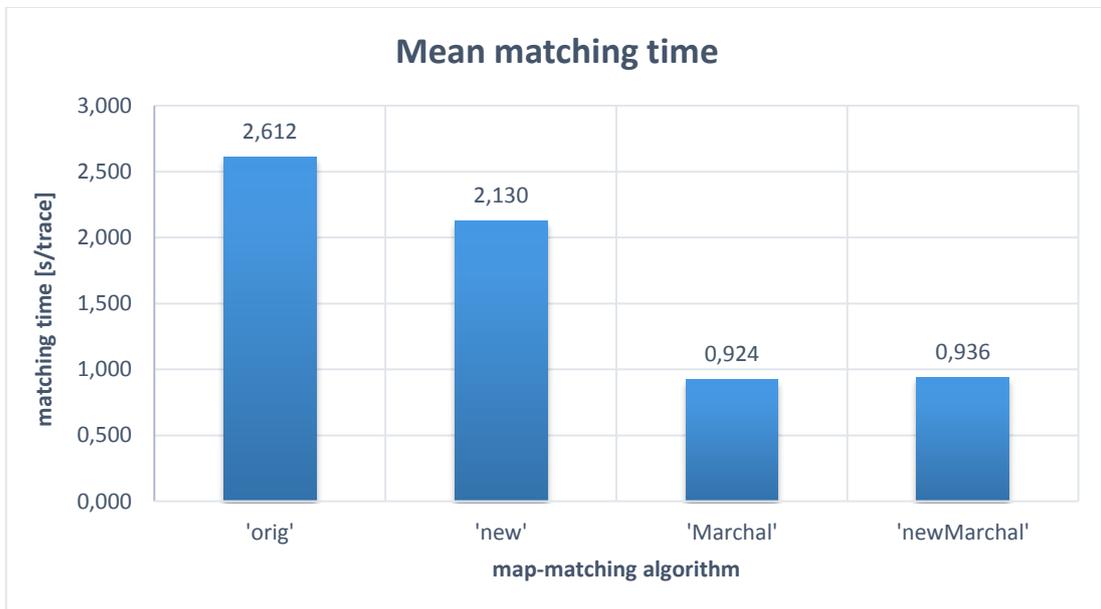


Figura 3.44: tempo medio di matching per traccia

Per quanto riguarda il campione di tracce raccolto, i risultati hanno mostrato un buon matching, senza evidenti differenze, tra gli algoritmi ‘orig’ e ‘new’; l’algoritmo ‘Marchal’ ha messo in luce i difetti di cui si è già discusso (paragrafo 6.2), mentre il ‘NewMarchal’ ha evidenziato un migliore comportamento. Di seguito è riportata una tabella riassuntiva dei risultati ottenuti.

		'orig'					
ID_ROUTE	REAL LEN [m]	LEN_ROUTE [m]	REAL LENGTH INDEX	MATCHIND	LENGTHIND	TIME_MATCH [s]	ERR_DIST [m]
11	992,420	789,800	0,80	0,88	0,77	0,574	20,7
10	2709,370	3108,120	1,15	0,94	1,09	1,246	9,9
12	2084,170	2168,880	1,04	0,98	0,95	1,460	15,1
7#	1128,730	1235,340	1,09	1,00	1,02	0,555	6,8
1	525,350	574,550	1,09	1,00	1,06	1,038	4,9
0	956,870	967,860	1,01	1,00	1,11	1,160	4,9
3	1297,040	1252,910	0,97	0,98	0,99	2,827	7,4
2	1596,590	1824,510	1,14	0,85	1,13	0,759	12,6
5	1266,520	1336,520	1,06	1,00	0,94	1,032	5,9
4	1514,570	1785,150	1,18	0,98	1,07	0,773	6,9
6	735,970	792,730	1,08	1,00	1,13	1,351	7,3
9	1101,030	1180,830	1,07	1,00	1,06	3,811	6,8
8	1064,700	1133,230	1,06	1,00	0,97	3,953	8,0
mean		1396,19	1,06	0,97	1,02	1,580	9,0

		'new'					
ID_ROUTE	REAL LEN [m]	LEN_ROUTE [m]	REAL LENGTH INDEX	MATCHIND (%)	LENGTHIND (%)	TIME_MATCH [s]	ERR_DIST [m]
11	992,420	735,120	0,74	0,88	0,72	1,262	21,8
10	2709,370	2693,320	0,99	0,88	0,95	1,288	12,2
12	2084,170	2084,170	1,00	0,98	0,91	2,305	15,1
7#	1128,730	1083,010	0,96	0,83	0,90	0,434	14,9
1	525,350	495,700	0,94	1,00	0,91	1,585	5,3
0	956,870	929,790	0,97	1,00	1,07	1,443	4,9
3	1297,040	1297,040	1,00	1,00	1,02	1,804	6,1
2	1596,590	1266,960	0,79	0,66	0,79	1,017	33,6
5	1266,520	1278,830	1,01	1,00	0,90	1,034	5,8
4	1514,570	1514,570	1,00	1,00	0,91	1,278	6,4
6	735,970	no match					
9	1101,030	1101,030	1,00	1,00	0,99	2,403	6,8
8	1064,700	1074,100	1,01	1,00	0,92	3,172	8,0
mean		1296,14	0,95	0,94	0,92	1,585	11,7

		'Marchal'				
ID_ROUTE	REAL LEN [m]	LEN_ROUTE [m]	REAL LENGTH INDEX	LENGTHIND	TIME_MATCH [s]	ERR_DIST [m]
11	992,420	1189,330	1,20	1,17	0,593	7,6
10	2709,370	3692,810	1,36	1,30	1,496	21,1
12	2084,170	2806,590	1,35	1,14	1,590	17,1
7#	1128,730	1801,590	1,60	1,49	0,608	5,7
1	525,350	739,450	1,41	1,36	0,326	2,3
0	956,870	1000,680	1,05	1,15	0,531	4,9
3*	1297,040	1247,360	0,96	1,30	0,763	4,9
2	1596,590	2624,950	1,64	1,67	0,998	5,4
5	1266,520	1933,170	1,53	1,36	2,387	5,7
4	1514,570	2169,430	1,43	1,29	0,919	6,5
6	735,970	870,970	1,18	1,24	0,328	10,6
9	1101,030	1906,110	1,73	1,71	0,702	5,6
8	1064,700	1848,040	1,74	1,59	0,514	4,0
	mean	1833,11	1,40	1,37	0,904	7,8

		'newMarchal'				
ID_ROUTE	REAL LEN [m]	LEN_ROUTE [m]	REAL LENGTH INDEX	LENGTHIND	TIME_MATCH [s]	ERR_DIST [m]
11	992,420	1069,680	1,08	1,05	0,546	8,3
10	2709,370	2849,010	1,05	1,00	3,151	11,1
12	2084,170	2473,580	1,19	1,01	1,404	20,3
7#	1128,730	1237,070	1,10	1,02	1,871	6,4
1	525,350	631,530	1,20	1,16	1,092	2,6
0	956,870	929,790	0,97	1,07	1,122	4,9
3*	1297,040	1053,420	0,81	1,09	1,139	5,0
2	1596,590	1598,040	1,00	1,02	1,013	5,6
5	1266,520	1526,310	1,21	1,07	0,811	5,8
4	1514,570	1698,410	1,12	1,01	0,826	6,6
6	735,970	791,350	1,08	1,13	0,296	7,3
9	1101,030	1170,920	1,06	1,05	0,686	6,8
8	1064,700	1213,990	1,14	1,04	0,967	5,5
	mean	1403,32	1,08	1,06	1,148	7,4

Tabella 3.20: confronto tra i risultati del map-matching degli algoritmi 'orig', 'new', 'Marchal' e 'NewMarchal' sul campione di tracce raccolto

Nelle figure seguenti è esposto un esempio di confronto (Fig. 3.44).

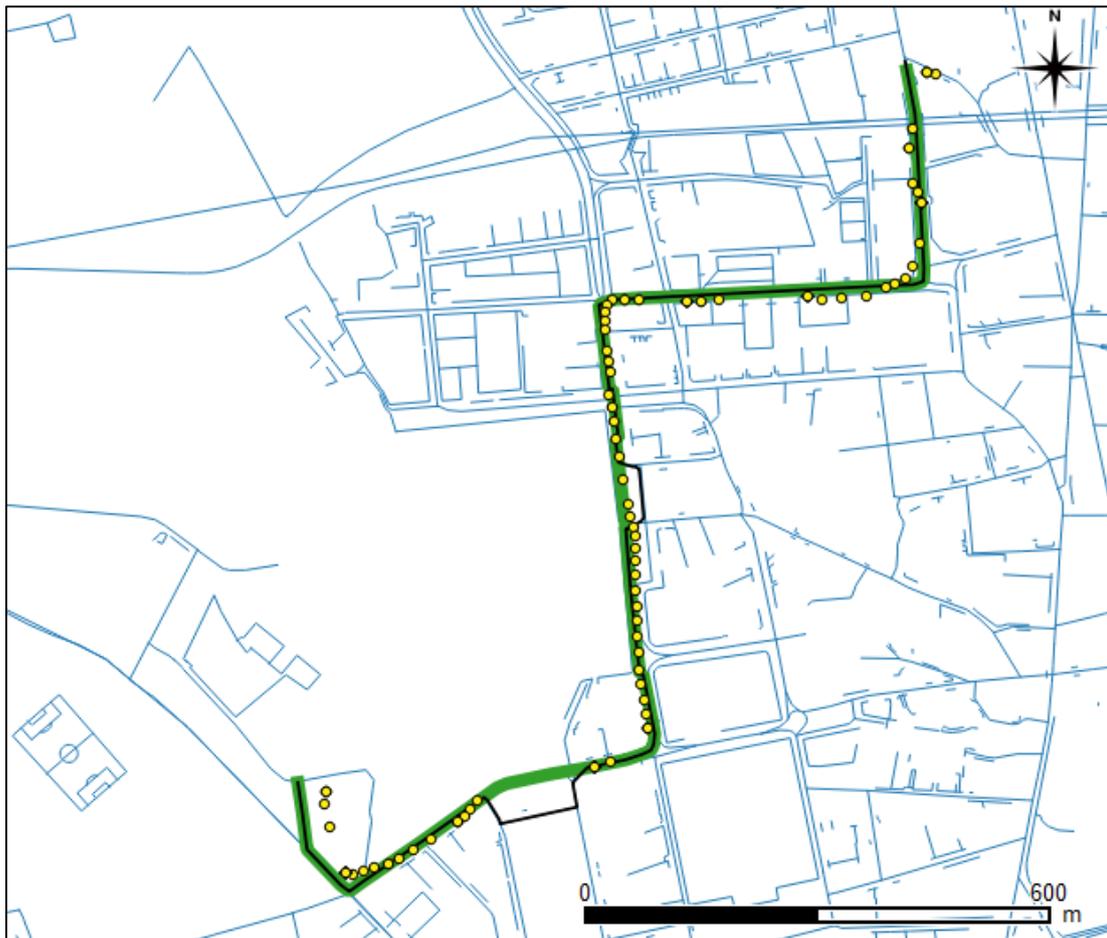


Figura 3.44 a): esempio di matching effettuato tramite l'algoritmo 'orig' (in nero), sovrapposto al percorso reale (in verde), tratto dal campione raccolto; in questo caso il matching presenta alcune imperfezioni, dovute al fatto che i punti ricadono anche in buffers di archi errati di lunghezza minore a quella degli archi corretti (*Length Index* = 0,94; *Real Length Index* = 1,06; *Match Index* = 1,00; *error distance* = 5,9 m/point)

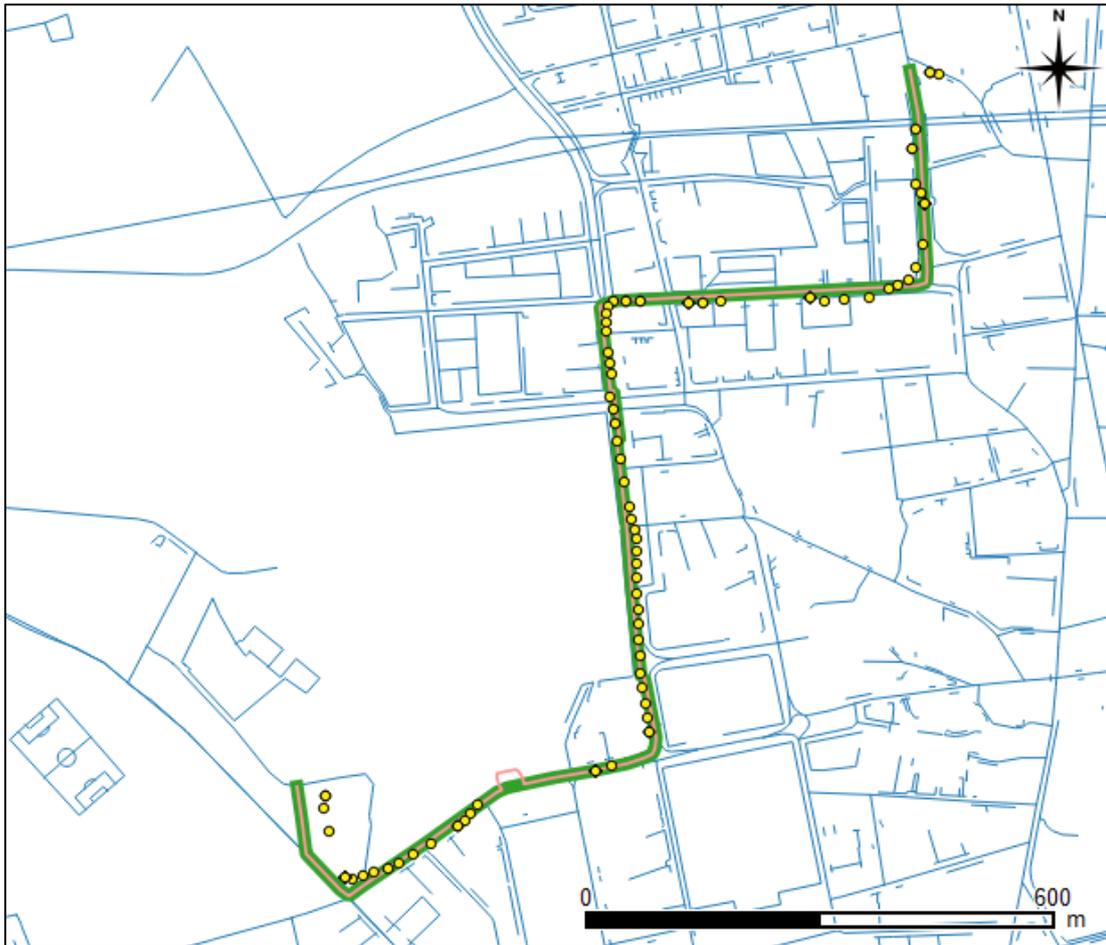


Figura 3.44 b): esempio di matching effettuato tramite l'algoritmo 'new' (in rosa), sovrapposto al percorso reale (in verde), tratto dal campione raccolto; in questo caso il matching è quasi perfetto, grazie alla maggiore influenza dei punti rispetto alla lunghezza nel calcolo dei costi degli archi ($Length Index = 0,90$; $Real Length Index = 1,01$; $Match Index = 1,00$; $error distance = 5,8$ m/point)

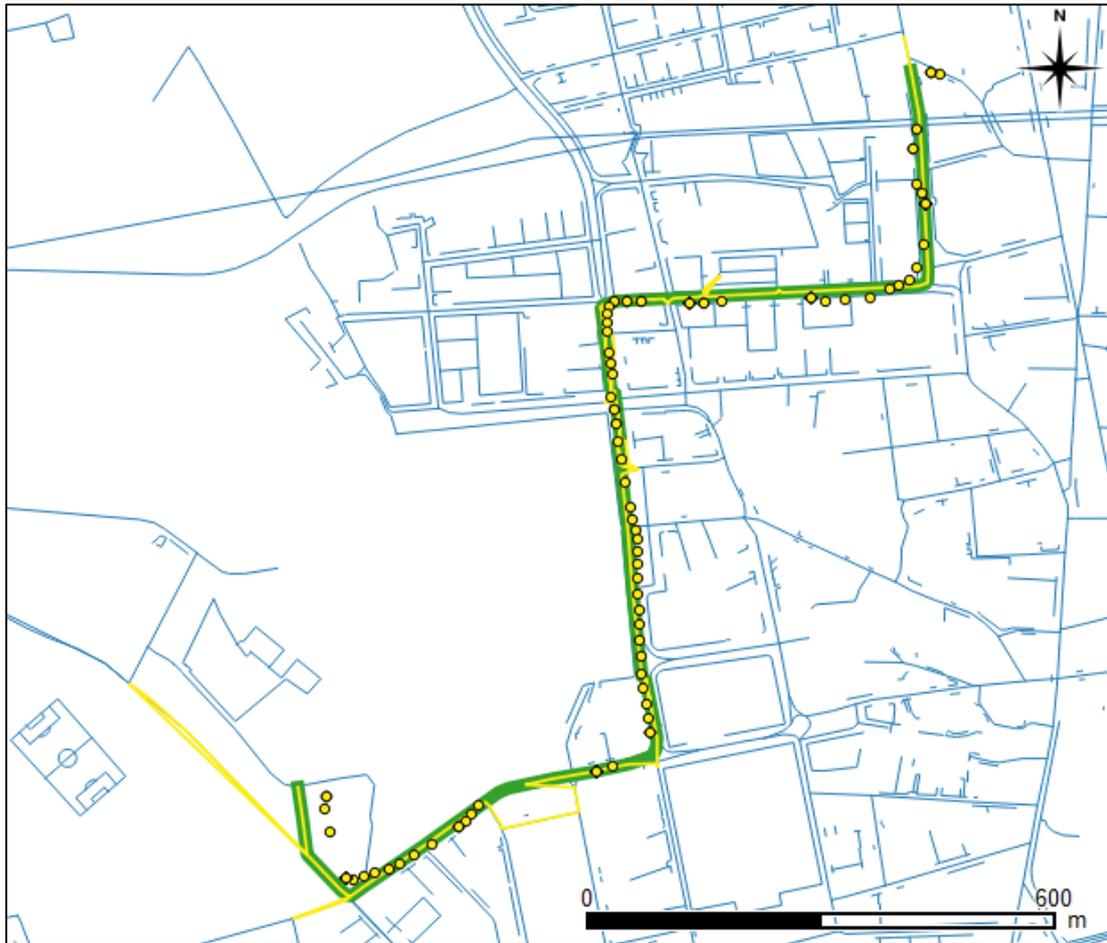


Figura 3.44 c): esempio di matching effettuato tramite l’algoritmo ‘Marchal’ (in giallo), sovrapposto al percorso reale (in verde), tratto dal campione raccolto; in questo caso il matching presenta molteplici inesattezze, dovute al calcolo dello score dell’itinerario attraverso la misura della distanza punto - arco (*Length Index* = 1,36; *Real Length Index* = 1,53; *error distance* = 5,7 m/point)

benché nella versione aggiornata che è stata proposta ('new') il tempo medio di matching è inferiore di circa il 20% e i parametri hanno minore influenza sui risultati.

La seconda metodologia, invece, ha fornito un minor numero di risultati, assolutamente inaccurati nel caso dell'algoritmo originale fornito da Marchal. L'esattezza dei matching è invece notevolmente migliorata eseguendo la versione aggiornata proposta nel presente lavoro ('NewMarchal'), che peraltro non comporta alcun peggioramento significativo nel tempo medio di matching. Questo è senza alcun dubbio l'aspetto più vantaggioso di tale metodologia, in quanto i valori medi del tempo di matching sono più che dimezzati rispetto a quelli della prima metodologia testata.

10. RISULTATI FINALI

Nel presente paragrafo sono riportati i risultati ritenuti corretti del map-matching eseguito tramite tutte le metodologie e gli algoritmi descritti in precedenza: sono mantenuti gli itinerari i cui indici di valutazione presentano un valore ritenuto accettabile (*Length Index* compreso tra 0,8 e 1,2, e *Match Index* compreso tra 0,8 e 1) e sono scartati i risultati restanti, considerati errati. Di seguito è riportata una tabella riassuntiva dei risultati che sono stati ottenuti eseguendo gli algoritmi di map-matching denominati 'orig', 'new', 'Marchal' e 'NewMarchal' sul database di tracce Endomondo.

	'orig'		'new'		'Marchal'		'newMarchal'	
Traces number (cleaning)	5969		5969		10971		10971	
Successful matches	3249	54,4%	2937	49,2%	121	1,1%	5032	45,9%
Mean matching time [seconds]	2,703 s		2,066 s		0,604 s		1,055 s	
Mean Distance Error	12,1 m/point		12,3 m/point		11,0 m/point		16,1 m/point	
Standard deviation distance error	6,7 m/point		7,0 m/point		12,7 m/point		22,2 m/point	
CV distance error	0,55		0,57		1,15		1,38	
50%percentile distance error	10,6 m/point		10,5 m/point		7,4 m/point		9,6 m/point	
75%percentile distance error	15,2 m/point		15,6 m/point		11,2 m/point		16,4 m/point	
90%percentile distance error	20,2 m/point		20,8 m/point		17,8 m/point		31,0 m/point	
95%percentile distance error	24,3 m/point		24,5 m/point		41,0 m/point		48,1 m/point	
Mean Length Index	0,972		0,959		1,117		1,070	
Standard deviation length index	0,087		0,089		0,055		0,057	
CV length index	0,090		0,092		0,049		0,053	
50%percentile length index	0,960		0,940		1,120		1,060	
75%percentile length index	1,030		1,020		1,170		1,110	
80%percentile length index	1,050		1,040		1,170		1,120	
90%percentile length index	1,100		1,090		1,180		1,160	
95%percentile length index	1,150		1,130		1,190		1,170	
Mean Match Index	0,931		0,929					
Standard deviation match index	0,060		0,064					
CV match index	0,067		0,068					
5%percentile match index	0,820		0,810					
10%percentile match index	0,840		0,830					
20%percentile match index	0,870		0,860					
25%percentile match index	0,880		0,880					
50%percentile match index	0,940		0,940					
Length index in range [0,80 - 1,20]	3249	54,4%	2937	49,2%	121	1,1%	5032	45,9%
Match index in range [0,80 - 1,00]	3249	54,4%	2937	49,2%				
Total distance travelled [km]	9989,4 km		8996,0 km		161,8 km		9343,7 km	
Total distance travelled on bikeways [km]	2316,2	23,2%	1938,9	21,6%	78,8	48,7%	1090,8	11,7%
Mean distance travelled on bikeways	712,9 m/route		660,2 m/route		651,5 m/route		216,8 m/route	
Routes containing bikeways	2672	44,8%	2135	35,8%	58	0,5%	2063	18,8%
Loading time [seconds]	412,5 s		369,4 s		368,7 s		377,2 s	
Cleaning data time [seconds]	134,8 s		132,9 s		1990,6 s		1937,0 s	
Processing data time [seconds]	16741,2 s		13770,2 s		9823,2 s		10090,8 s	
Total execution time [seconds]	17248,0 s		14272,5 s		12182,5 s		12405,1 s	

Tabella 3.21: confronto tra i risultati del map-matching degli algoritmi 'orig', 'new', 'Marchal e 'NewMarchal' sul database Endomondo

Il numero di matches ritenuti corretti (*successful matches*) risulta sovrapponibile per gli algoritmi 'orig', 'new' e 'NewMarchal' (54,4%, 49,2% e 45,9% rispettivamente, mentre è praticamente trascurabile per l'algoritmo 'Marchal' (appena 1,1%) a causa dei problemi già descritti al paragrafo 6.2 che allungano di molto l'itinerario, aumentandone così il valore del *Length Index* oltre il valore limite imposto di 1,2; per tale motivo la maggior parte dei percorsi ricostruiti sono scartati (Fig. 3.45).

I valori medi e percentili dei *Length Index*, dei *Match Index* e dell'errore di distanza (*distance error*) sostanzialmente confermano la bontà dei risultati ottenuti: difatti, i primi sono in tutti i casi distribuiti in modo poco disperso attorno ad un valore medio molto prossimo all'unità (0,972 per l'algoritmo 'orig', 0,959 per l'algoritmo 'new', 1,070 per l'algoritmo 'NewMarchal'), con bassi coefficienti di variazione. Anche i *Match Index*, validi solo per la prima metodologia, sono distribuiti in modo molto concentrato attorno a valori medi vicini all'unità (0,931 per l'algoritmo 'orig' e 0,929 per l'algoritmo 'new').

Infine, anche l'errore di distanza (*distance error*) risulta, per quasi tutti gli itinerari ricostruiti con le diverse metodologie di map-matching, di ordine di grandezza comparabile all'errore di misurazione del GPS stesso: anch'esso, infatti, presenta in tutti i casi una distribuzione poco dispersa attorno ad un valore medio decisamente basso (12,1 m/punto per l'algoritmo 'orig', 12,3 m/punto per l'algoritmo 'new' e 16,1 m/punto per l'algoritmo 'NewMarchal').

Per quanto riguarda l'aspetto dei tempi di esecuzione (Fig. 3.46), le osservazioni compiute al paragrafo 9 possono essere confermate. La prima metodologia si riafferma come la più gravosa, con tempi medi di matching (*mean matching time*) decisamente maggiori: 2,703 secondi/traccia per l'algoritmo 'orig' e 2,066 secondi/traccia per l'algoritmo 'new', che quindi mostra un cospicuo miglioramento rispetto al precedente. La seconda metodologia vanta invece tempi di esecuzione molto più rapidi, con un tempo medio di matching pari a 1,055 secondi/traccia per l'algoritmo 'NewMarchal', che comprova ancora una volta la sua maggiore efficienza computazionale.



Figura 3.45: percentuale di matches riusciti

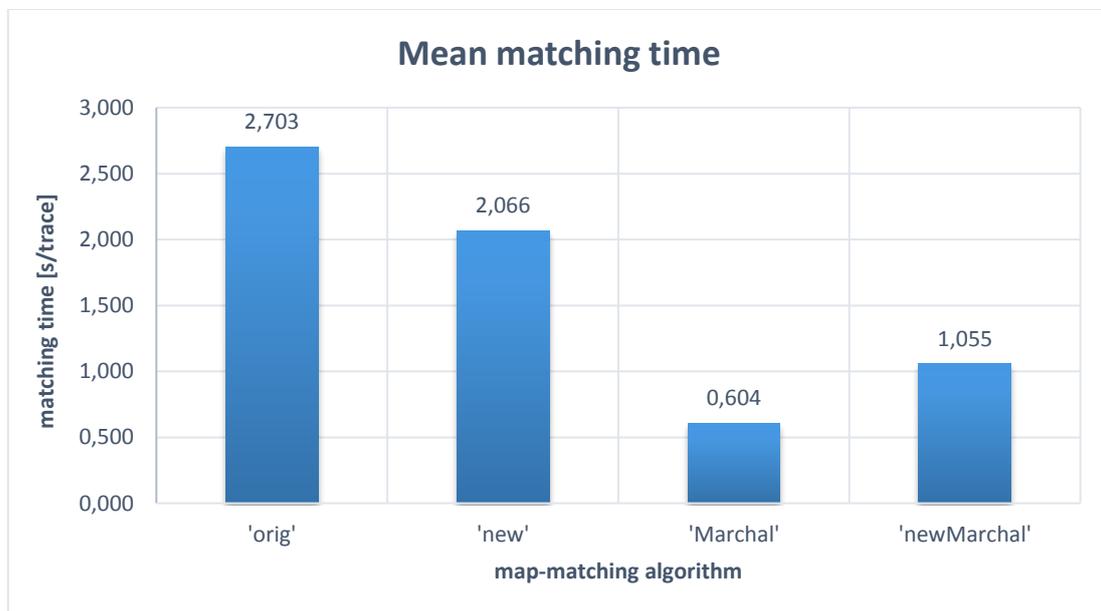


Figura 3.46: tempo medio di matching

CONCLUSIONI

Il presente lavoro conduce un'analisi di alcune metodologie di map-matching per l'identificazione degli itinerari degli utenti, in particolare della mobilità ciclabile. Si tratta di un procedimento che presenta una molteplicità di applicazioni, le quali spaziano dal controllo e gestione in tempo reale del sistema di trasporto, delle situazioni di emergenza e di altri *Location Based Services*, alla pianificazione dei trasporti, mediante l'analisi delle scelte modali e del percorso degli utenti.

Obiettivo del presente studio è dunque l'elaborazione di algoritmi per il map-matching, al fine di identificare gli itinerari scelti dagli utenti, che presentino una buona accuratezza dei risultati ed un costo computazionale accettabile.

La presenza di errori nel posizionamento GPS e nelle mappe disponibili rende infatti necessaria l'applicazione di algoritmi per il map-matching, a partire da semplici dati di localizzazione GPS, costituiti da tracce - un insieme ordinato di punti identificati tramite le coordinate geografiche e l'istante di rilevamento - e dalla rete, ovvero un grafo costituito da un insieme di archi e nodi.

Nel presente studio è stato quindi descritto il procedimento generale del map-matching; sono state successivamente riportate le problematiche più diffuse riscontrate e descritte le classificazioni degli algoritmi elaborati in letteratura, effettuate in base al periodo di tempo durante il quale essi operano (*online*, *offline*) o in base all'approccio metodologico al problema (*geometrico*, *topologico*, *probabilistico*, *avanzato*). Infine, è stato eseguito uno studio su alcuni metodi di map-matching, utilizzando il database di spostamenti di ciclisti "Endomondo", le cui tracce sono state registrate in forma anonima e volontaria nel mese di Maggio 2013, attraverso gli smartphone di utenti della mobilità ciclabile di Bologna; è stato raccolto inoltre, in modo analogo, un campione di tracce di prova di spostamenti in bicicletta, effettuati nel mese di Novembre 2015 nell'area urbana di Catania, adibita ad ambiente di testing per gli algoritmi. In entrambi i casi ci si è avvalsi della rete cittadina, scaricabile in forma libera e gratuita dal server di *OpenStreetMap*; per l'esecuzione del map-matching, si è lavorato in ambiente SUMO e si è adoperato il linguaggio di programmazione Python.

Le metodologie applicate per realizzare il map-matching utilizzando i dati a disposizione sono state due. La prima, ideata da J.Schweizer e F.Rupi, sfrutta un approccio di tipo probabilistico; la seconda, pubblicata da F.Marchal, J.Hackney e K.Axhausen, adopera invece un approccio misto topologico - probabilistico, mediante la tecnica delle ipotesi multiple. Poiché le metodologie studiate hanno evidenziato alcuni difetti, nel presente lavoro si è proposta una versione aggiornata dei due algoritmi che potesse risolvere tali problematiche, operando delle modifiche ove ritenuto necessario.

Dopo una prima fase di filtraggio dei dati (*clean*), per ogni algoritmo è stata effettuata un'analisi dei parametri, al fine di ricavarne i valori ottimali; le comparazioni tra i risultati del map-matching, eseguito mediante i vari algoritmi, sono state quindi condotte graficamente e mediante degli indici di valutazione, in termini di accuratezza e di costo computazionale e tempo impiegato per il matching.

La prima metodologia, nella versione originale - algoritmo denominato 'orig' - esegue per ogni traccia un processo di assegnazione di un costo o peso (*weight*) ad ogni arco della rete, dipendente da due parametri - c_{length} e c_{bike} , rispettivamente per archi stradali e ciclabili - e dal numero di punti ricadenti all'interno di un buffer di larghezza pari a 30 metri, creato attorno ad ogni arco; la quantità di tali punti ne diminuisce progressivamente il costo. L'algoritmo di Dijkstra è quindi applicato per la ricerca del percorso di minimo costo (*shortest path*), dati i costi calcolati in tal modo; tale percorso è infine identificato come itinerario effettivamente seguito.

I risultati hanno evidenziato una buona accuratezza degli itinerari ricostruiti, una volta individuato il valore ottimale del parametro c_{length} , decisivo nel calcolo dei costi degli archi, ma anche una forte dipendenza dallo stesso, e soprattutto un alto tempo medio di matching, dovuto perlopiù alla fase di attribuzione del costo a tutti gli archi della rete.

L'algoritmo 'new', proposto come possibile aggiornamento del precedente, utilizza la medesima metodologia, con l'apporto di alcune modifiche, al fine di eliminare la dipendenza dal parametro c_{length} e ridurre il tempo di matching. In particolare, è stata ridefinita la fase di assegnazione dei costi agli archi, in quanto si utilizza una funzione di costo differente in base alla presenza o meno di punti GPS all'interno del buffer

dell'arco, con l'intento di aumentare l'influenza dei punti nella ricostruzione del percorso. L'algoritmo di Dijkstra è quindi applicato per la ricerca del percorso di minimo costo (*shortest path*), dati i nuovi costi calcolati; tale percorso è infine individuato come itinerario effettivo.

I risultati hanno mostrato anche in questo caso una buona accuratezza degli itinerari ricostruiti, con una migliore precisione in particolare nei tratti in assenza di punti GPS; inoltre, la dipendenza dal parametro principale c_{length} è fortemente limitata, in quanto i risultati sono del tutto analoghi al variare di tale parametro. Infine, anche il tempo di matching per traccia ha palesato dei miglioramenti, con un valore inferiore di circa il 20% rispetto a quello della versione originale, sebbene esso rimanga comunque abbastanza oneroso a causa della fase di attribuzione del costo ad ogni arco della rete.

La seconda metodologia studiata processa invece sequenzialmente i punti GPS in base al loro ordine cronologico adoperando la tecnica delle ipotesi multiple, ovvero la formulazione di una serie di ipotesi sul percorso sulla base della topologia della rete - connessioni tra gli archi ai nodi e ricerca topologica degli archi seguenti alle intersezioni - e la conservazione in memoria di un set di possibili alternative per il percorso effettivo. Ad esse è assegnato un punteggio (*score*), che nella versione originale - algoritmo denominato 'Marchal' - dipende esclusivamente dalla distanza tra il punto processato e l'ultimo arco del percorso ricostruito progressivamente. Ad ogni passo viene dapprima valutato se l'intersezione seguente o il nodo finale dell'ultimo arco sono stati superati; infine, vengono eliminate le alternative aventi il punteggio peggiore. Tra le rimanenti, dopo l'elaborazione dell'ultimo punto della traccia, è scelta quella con il migliore punteggio.

I risultati hanno evidenziato una pessima accuratezza degli itinerari, i quali si sono dimostrati decisamente lunghi e poco realistici; gli errori si verificano in particolare nelle situazioni di incrocio con strade bidirezionali, ove i percorsi ricostruiti presentano spesso delle deviazioni irrealistiche. Tuttavia, il tempo medio di matching ha mostrato dei valori decisamente bassi, più che dimezzati rispetto a quelli della prima metodologia studiata.

Dato il tempo di matching per traccia decisamente basso, che costituisce il principale vantaggio di tale metodologia, nel presente lavoro è stato quindi proposto l'algoritmo

‘NewMarchal’ come possibile miglioramento del precedente, con l’obiettivo di migliorare l’accuratezza degli itinerari senza aumentare eccessivamente il tempo medio di matching. In particolare, ci si è proposto di limitare le situazioni erronee di cui sopra, le quali costituiscono le maggiori imprecisioni, ed in tal modo di ricostruire degli itinerari più precisi e significativi.

Le modifiche apportate riguardano sia le condizioni per verificare se l’intersezione seguente o il nodo finale dell’ultimo arco sono stati superati, sia il calcolo del punteggio di ogni percorso conservato in memoria. Nel calcolo dello score, infatti, è stato imposto un valore minimo per la distanza punto - arco da considerare; inoltre, è stata tenuta in considerazione non soltanto la distanza tra il punto e l’arco, ma anche la similitudine tra la direzione dell’arco e la traiettoria dei punti GPS consecutivi, individuata mediante il segmento che unisce il punto precedente e quello processato, e che si presume essere indicativa della reale direzione di spostamento dell’utente. Tale similitudine è stata valutata attraverso l’angolo formato dai due elementi; l’utilizzo della funzione seno, presa in valore assoluto, permette di premiarne il parallelismo, minimizzando lo score, e di penalizzarne la perpendicolarità, massimizzando il punteggio.

I risultati hanno mostrato un notevole miglioramento dell’accuratezza degli itinerari, eliminando quasi del tutto gli errori che si verificavano in precedenza nelle situazioni di incrocio con strade bidirezionali, grazie alle modifiche apportate in questa versione proposta. In aggiunta, il tempo medio di matching non ha presentato degli aumenti significativi rispetto all’algoritmo originale ‘Marchal’, mantenendosi su valori decisamente inferiori a quelli della prima metodologia studiata.

I risultati finali dei map-matching eseguiti mediante i vari algoritmi, ottenuti scartando gli itinerari ritenuti errati sulla base dei valori dei loro indici, hanno quindi confermato le indicazioni sul tempo medio di matching, decisamente minore per la seconda metodologia (algoritmi ‘Marchal’ e ‘NewMarchal’). Inoltre, essi hanno mostrato una notevole sovrapponibilità per gli algoritmi ‘orig’, ‘new’ e ‘NewMarchal’, in termini di percentuale di matches riusciti, di valori degli indici e degli errori di distanza, i quali si sono rivelati di ordine di grandezza comparabile all’errore di misurazione del GPS

stesso: difatti, gli itinerari identificati hanno evidenziato un ottimo livello di accuratezza.

Tra gli obiettivi delle possibili future ricerche inerenti al map-matching, oltre al miglioramento dell'accuratezza degli itinerari, specialmente nelle situazioni più complesse, riveste un ruolo fondamentale il tempo impiegato per l'esecuzione degli algoritmi, considerando il notevole aumento della disponibilità di dati GPS e la diffusione sempre maggiore di smartphone e di altri dispositivi mobili. Difatti, una quantità di dati superiore necessita di un'attenzione crescente ai tempi di esecuzione, che vanno ridotti adottando metodologie che richiedono un numero di operazioni minore ed a minore complessità. È altresì di fondamentale importanza disporre di dati di migliore qualità: le reti disponibili dovrebbero essere perfezionate, in quanto talvolta esse appaiono incomplete, mancando di archi o di informazioni su di essi, o di connessioni tra archi realmente congiunti ai nodi. Tali carenze possono risultare oltremodo penalizzanti nel map-matching, in particolare nel caso di spostamenti di ciclisti, i quali percorrono spesso sentieri di parchi o aree pedonali non riportati sulle mappe; per tale motivo la rete dovrebbe essere il più dettagliata possibile. Infine, nuove tecnologie nell'ambito del sistema GPS o degli smartphone, così come l'uso di sistemi di posizionamento alternativi quali GLONASS o Galileo, possono certamente incrementare la precisione della geolocalizzazione, ed in tal modo facilitare l'operazione di map-matching ed aumentarne l'accuratezza.

BIBLIOGRAFIA

- Behrisch M., Bieker L., Erdmann J., Krajzewicz D., *SUMO - Simulation of Urban MObility: an overview*, in SIMUL 2011, The Third International Conference on Advances in System Simulation, 2011, <http://sumo-sim.org/>
- Bernstein D., Kornhauser A., *An introduction to map-matching for personal navigation assistants*, in Transportation Research Board, Technical Report, 1998.
- Bierlaire M., Chen J., Newman J., *A probabilistic map matching method for smartphone GPS data*, in Transportation Research Part C, 2012, 26, pp. 78-98.
- Bohte W., Maat K., *Deriving and validating trip purposes and travel modes for multi-day GPS-based data travel surveys: a large-scale application in the Netherlands*, Transportation Research Part C, 2009, 17, pp. 285-297.
- Câmara Pereira F., Costa H., Martinho Pereira N., *An off-line map-matching algorithm for incomplete map databases*, European Conference of Transport Research Institutes (ECTRI), 2009, 1, pp. 107-124.
- Cina A., *GPS: Principi, modalità e tecniche di posizionamento*, Celid, 4° ristampa, 2004.
- Fu M., Li J., Wang M., *A hybrid map-matching algorithm based on fuzzy comprehensive Judgment*, in IEEE Proceedings on Intelligent Transportation Systems, 2004, pp. 613-617.
- Joo S., Oh C., Jeong E., Lee G., *Categorizing bicycling environments using GPS-based public bicycle speed data*, in Transportation Research Part C, 2015, 56, pp. 239-250.
- Kim W., Jee G., Lee J., *Efficient use of digital road map in various positioning for ITS*, IEEE Symposium on position, location and navigation, San Diego, CA, USA, 2000.
- Marchal F., Hackney J., Axhausen K., *Efficient map-matching of large GPS data sets - Tests on a speed monitoring experiment in Zurich*, Arbeitsbericht Verkehrs- und Raumplanung, Institut für Verkehrsplanung und Transportsysteme, ETH Zürich, Zürich, 2004.
- Menghini G., Carrasco N., Schüssler N., Axhausen K., *Route choice of cyclists: discrete choice modelling based on GPS-data*, Arbeitsberichte Verkehrs- und Raumplanung, 544, IVT, ETH Zurich, Zurich, 2009.

- Ochieng W., Quddus M., Noland R., *Map-matching in complex urban road networks*, in Brazilian Journal of Cartography, 2004, 55(2), pp. 1-18.
- Ochieng W., Sauer K., Walsh D., Brodin G., Griffin S., Denny M., *GPS integrity and potential impact on aviation safety*, in Journal of Navigation, 2003, 56 (1), pp. 51–65.
- Phuyal B., *Method and use of aggregated dead reckoning sensor and GPS data for map-matching*, in proceedings of the Institute of Navigation (ION) annual conference, Portland, OR, USA, 2002, pp. 430-437.
- Pyo J., Shin D., Sung T., *Development of a map matching method using the multiple hypothesis technique*, in IEEE Conference proceedings on Intelligent Transportation Systems, 2001, pp. 23–27.
- Quddus M., Ochieng W., Noland R., *Integrity of map-matching algorithms*, in Transportation Research Part C, 2006, 14, pp. 283-302.
- Quddus M., Ochieng W., Noland R., *Current map-matching algorithms for transport applications: State-of-the art and future research directions*, in Transportation Research Part C, 2007, 15, pp. 312-338.
- Quddus M., Ochieng W., Zhao L., Noland R., *A general map matching algorithm for transport telematics applications*, in GPS Solutions, 2003, 7(3), pp. 157-167.
- Quddus M., Washington S., *Shortest path and vehicle trajectory aided map-matching for low frequency GPS data*, in Transport Research Part C, 2015, 55, pp. 328-339.
- Rao G.S., *Global Navigation Satellite Systems*, Tata McGraw-Hill, 2010.
- Schweizer J., Rupi F., *Map matching and cycling infrastructure analyses with SUMO and python*, in proceedings of the SUMO2014 Modeling Mobility with Open Data Conference, Berlin – Adlershof, Germany, 2014, pp. 195-205.
- Schüssler N., Axhausen K., *Processing GPS raw data without additional information*, in Transportation Research Board, 2009, 12, pp. 28-36.
- USA Government – Department of Defense, *Global Positioning System standard positioning service performance standard*, 4th Edition, 2008.
- White C., Bernstein D., Kornhauser A., *Some map-matching algorithms for personal navigation assistants*, in Transportation Research Part C, 2000, 8, pp. 91-108.
- Yang D., Cai B., Yuan Y., *An improved map-matching algorithm used in vehicle navigation system*, in IEEE Proceedings on Intelligent Transportation Systems, 2003, 2, pp. 1246-1250.

DLR and contributors, SUMO, <http://sumo.sourceforge.net>
European cycling challenge, <http://www.europeancyclingchallenge.eu>
Endomondo, <http://www.endomondo.com>
Latex Equation Editor, Codecogs, <http://www.codecogs.com/latex/eqneditor.php>
Numpy, <http://www.numpy.org/>
Oil and Gas Journal, <http://www.ogj.com>
OpenStreetMap, <http://www.openstreetmap.org>
Python, <http://www.python.org/>
Quantum GIS (QGIS), <http://www.qgis.org/>
Runtastic Road Bike, <http://www.runtastic.com>
Shapely, <http://toblerity.org/shapely/>
USA Government, <http://www.gps.gov/>

APPENDICE

È riportato in appendice lo script redatto per l'esecuzione degli algoritmi di map-matching e per una prima elaborazione dei risultati.

```

1 import time
2
3 t0 = time.time()
4
5 import os, sys
6
7 sys.path.append(os.path.join('.', '..', 'sumopy'))
8
9 from sumopy import SumoScenario
10 from plugins.traces import traces
11 from lib_meta import logger
12 import numpy as np
13 import pylab as P
14 import math
15
16 scenario = SumoScenario('bo', name='Bologna', dirpath = '.', is_connections = False)
17
18 outputfilename = 'out_'+os.path.basename(__file__).split('.')[0] #sys.argv[0].split('.')[0]# or __file__ gets entire path
19 print 'outputfilename', outputfilename
20 #sys.exit()
21 log = logger.Logger()
22

```

Figura A.1: import dei moduli e dello scenario, setting degli output

```

23 mytraces = traces.Traces(scenario,
24     method = 'newmarchal',
25     width_buffer = 50.0, #m
26     resolution_buffer = 4,
27     #c_length = 0.0001,
28     #c_bike = 0.9,
29     n_routes_follow = 20, # marchal
30     alpha = 0.7, # marchal
31     d0 = 10.0, #m newmarchal
32     id_mode = 'bicycle',
33     v_av = 3.33, # 3.33m/s = 12 km/h
34     #matchindex_min = 0.8,
35     lengthindex_min = 0.8,
36     lengthindex_max = 1.2,
37     duration_min = 30, #s
38     n_points_min = 10,
39     dist_min = 300, #m
40     dist_point_max = 300.0, #m
41     t_point_max = 30.0, #s
42     n_shortestroute = 1,
43 )
44
45 traces.import_traces(scenario, 'ecc-team-9172232-points.csv', mytraces)
46 #traces.import_gpx(scenario, 'Track1_Endomondo.gpx', mytraces, log = log)
47
48 t1 = time.time()
49
50 mytraces.clean(log=log)
51
52 tc = time.time()
53
54 mytraces.match(log=log)
55 #mytraces.filter_traces_between_zones()
56

```

Figura A.2: import delle tracce, clean ed esecuzione del map-matching. L'algoritmo è selezionato in 'method', e di seguito sono settati i valori dei parametri e degli indici

```

57 inds = np.flatnonzero(mytraces.cols.routes)
58 n_matches = len(inds)
59 times_match_mean = np.mean(mytraces.cols.times_match[inds])
60
61 errors_dist_mean = np.mean(mytraces.cols.errors_dist[inds])
62 errors_dist_stand_dev = math.sqrt(np.var(mytraces.cols.errors_dist[inds]))
63 errors_dist_CV = errors_dist_stand_dev/errors_dist_mean
64 err_percentiles = np.percentile(mytraces.cols.errors_dist[inds]/1000.0,[50,75,90,95])
65
66 ##Only for Marchal algorithm: 'Route distance error resulting from Marchal algorithm,\
67 # divided by the total number of GPS points of the trace.'
68 #errors_dist_meanM = np.mean(mytraces.cols.errors_dist_march[inds])
69 #errors_dist_stand_devM = math.sqrt(np.var(mytraces.cols.errors_dist_march[inds]))
70 #errors_dist_CVM = errors_dist_stand_devM/errors_dist_meanM
71 #err_percentilesM = np.percentile(mytraces.cols.errors_dist_march[inds]/1000.0,[50,75,90,95])
72
73 lengthindexes_mean = (np.mean(mytraces.cols.lengthindexes[inds]/100.00))
74 LImin,LImax = (0.8, 1.2)
75 n1 = len((mytraces.cols.lengthindexes[inds]/100.00)[(mytraces.cols.lengthindexes[inds]/100.00 >= LImin) &\
76 (mytraces.cols.lengthindexes[inds]/100.00 <= LImax)])
77 LI_stand_dev = math.sqrt(np.var(mytraces.cols.lengthindexes[inds]/100.00))
78 LI_CV = LI_stand_dev/lengthindexes_mean
79 LI_percentiles = np.percentile(mytraces.cols.lengthindexes[inds]/100.00,[50,75,80,90,95])
80
81 #In Marchal approach Match Index MI is nonsense
82 MImin, MImax = (0.8, 1.0)
83 matchindexes_mean = (np.mean(mytraces.cols.matchindexes[inds]/100.00))
84 n2 = len((mytraces.cols.matchindexes[inds]/100.00)[(mytraces.cols.matchindexes[inds]/100.00 >= MImin) &\
85 (mytraces.cols.matchindexes[inds]/100.00 <= MImax)])
86 MI_stand_dev = math.sqrt(np.var(mytraces.cols.matchindexes[inds]/100.00))
87 MI_CV = MI_stand_dev/matchindexes_mean
88 MI_percentiles = np.percentile(mytraces.cols.matchindexes[inds]/100.00,[5,10,20,25,50])
89
90 tot_length = sum(mytraces.cols.length[inds])/1000.0
91 nbike = len(np.flatnonzero(mytraces.cols.length_bikeway[inds]))
92 meanbike = np.mean(mytraces.cols.length_bikeway[inds])
93 sumbike = np.sum(mytraces.cols.length_bikeway[inds])/1000.0 #km
94

```

Figura A.3: calcolo del numero di matches riusciti, del tempo medio di matching, dei valori medi e percentili degli errori di distanza e degli indici *Length Index* e *Match Index*, e della distanza percorsa su rete ciclabile

```

95 binsLI = np.arange(0.0,3.1,0.1)
96 P.figure(1)
97 P.hist(mytraces.cols.lengthindexes[inds]/100.00,binsLI,align='mid',rwidth=0.8)
98 P.title('Length index: frequency')
99 P.xlabel('Length index')
100 P.ylabel('Frequency')
101 P.xlim(xmax=2)
102 P.ylim(ymax=1800)
103 P.savefig('out_lengthindex_freq')
104 P.show()
105 P.figure(2)
106 P.hist(mytraces.cols.lengthindexes[inds],binsLI*100,align='mid',rwidth=0.8, normed=True)
107 P.title('Length index: probability density function')
108 P.xlabel('Length index (%)')
109 P.ylabel('Probability density')
110 P.xlim(xmax=200)
111 P.ylim(ymax=0.05)
112 P.savefig('out_lengthindex_PDF')
113 P.show()
114 P.figure(3)
115 P.hist(mytraces.cols.lengthindexes[inds]/100.00,binsLI,align='mid',rwidth=0.8,normed=True,cumulative=True)
116 P.title('Length index: cumulative frequency')
117 P.xlabel('Length index')
118 P.ylabel('Cumulative frequency')
119 P.xlim(xmax=2)
120 P.ylim(ymax=1)
121 P.savefig('out_lengthindex_CF')
122 P.show()
123

```

Figura A.4: analisi statistica grafica dei *Length Index*

Appendice

```
124 #In Marchal approach Match Index MI is nonsense
125 binsMI = np.arange(0.0,1.05,0.05)
126 P.figure(1)
127 P.hist(mytraces.cols.matchindexes[inds]/100.00,binsMI,align='mid',rwidth=0.8)
128 P.title('Match index: frequency')
129 P.xlabel('Match index')
130 P.ylabel('Frequency')
131 P.xlim(xmax=1)
132 P.ylim(ymax=1800)
133 P.savefig('out_matchindex_freq')
134 P.show()
135 P.figure(2)
136 P.hist(mytraces.cols.matchindexes[inds],binsMI*100,align='mid',rwidth=0.8, normed=True)
137 P.title('Match index: probability density function')
138 P.xlabel('Match index (%)')
139 P.ylabel('Probability density')
140 P.xlim(xmax=100)
141 P.ylim(ymax=0.1)
142 P.savefig('out_matchindex_PDF')
143 P.show()
144 P.figure(3)
145 P.hist(mytraces.cols.matchindexes[inds]/100.00,binsMI,align='mid',rwidth=0.8,normed=True,cumulative=True)
146 P.title('Match index: cumulative Frequency')
147 P.xlabel('Match index')
148 P.ylabel('Cumulative frequency')
149 P.xlim(xmax=1)
150 P.ylim(ymax=1)
151 P.savefig('out_matchindex_CF')
152 P.show()
153
```

Figura A.5: analisi statistica grafica dei *Match Index*

```
154 binsERR = np.arange(0,4570,5)
155 P.figure(1)
156 P.hist(mytraces.cols.errors_dist[inds]/1000.0,binsERR,align='mid',rwidth=0.8)
157 P.title('Distance error: frequency')
158 P.xlabel('Distance error [m/point]')
159 P.ylabel('Frequency')
160 P.xlim(xmax=100)
161 P.ylim(ymax=1500)
162 P.savefig('out_err_freq')
163 P.show()
164 P.figure(2)
165 P.hist(mytraces.cols.errors_dist[inds]/1000.0,binsERR,align='mid',rwidth=0.8, normed=True)
166 P.title('Distance error: probability density function')
167 P.xlabel('Distance error [m/point]')
168 P.ylabel('Probability density')
169 P.xlim(xmax=100)
170 P.ylim(ymax=0.1)
171 P.savefig('out_err_PDF')
172 P.show()
173 P.figure(3)
174 P.hist(mytraces.cols.errors_dist[inds]/1000.0,binsERR,align='mid',rwidth=0.8,normed=True,cumulative=True)
175 P.title('Distance error: cumulative frequency')
176 P.xlabel('Distance error [m/point]')
177 P.ylabel('Cumulative frequency')
178 P.xlim(xmax=100)
179 P.ylim(ymax=1)
180 P.savefig('out_err_CF')
181 P.show()
182
```

Figura A.6: analisi statistica grafica dei *distance error*

```

183 binsBIKE = np.arange(0,4800,250)
184 P.figure(1)
185 P.hist(mytraces.cols.length_bikeway[inds],binsBIKE,align='mid',rwidth=0.8)
186 P.title('Distance travelled on bikeways: frequency')
187 P.xlabel('Distance travelled on bikeways [m]')
188 P.ylabel('Frequency')
189 P.xlim(xmax=4750)
190 P.ylim(ymax=1500)
191 P.savefig('out_bike_freq')
192 P.show()
193 P.figure(2)
194 P.hist(mytraces.cols.length_bikeway[inds],binsBIKE,align='mid',rwidth=0.8, normed=True)
195 P.title('Distance travelled on bikeways: probability density function')
196 P.xlabel('Distance travelled on bikeways [m]')
197 P.ylabel('Probability density')
198 P.xlim(xmax=4750)
199 P.ylim(ymax=0.002)
200 P.savefig('out_bike_PDF')
201 P.show()
202 P.figure(3)
203 P.hist(mytraces.cols.length_bikeway[inds],binsBIKE,align='mid',rwidth=0.8,normed=True,cumulative=True)
204 P.title('Distance travelled on bikeways: cumulative frequency')
205 P.xlabel('Distance travelled on bikeways [m]')
206 P.ylabel('Cumulative frequency')
207 P.xlim(xmax=4750)
208 P.ylim(ymax=1)
209 P.savefig('out_bike_CF')
210 P.show()
211

```

Figura A.7: analisi statistica grafica della distanza percorsa su rete ciclabile

```

212 t2 = time.time()
213
214 #Riepilogo:
215 log_file = open('log.txt','w')
216 print >> log_file, '-----'
217 print >> log_file, '\n          REPORT'
218 print >> log_file, '\nSuccessful matches: \t\t\t %d of %d (%.1f%%)' %(n_matches, len(mytraces.cols.routes),\
219 float(n_matches)/float(len(mytraces.cols.routes))*100)
220 print >> log_file, '\nMean matching time: \t\t\t %3f seconds' %(times_match_mean/1000)
221 print >> log_file, '\nMean distance error: \t\t\t %1f m/point' %(errors_dist_mean/1000)
222 print >> log_file, '\nStandard deviation distance error: \t %1f m/point' %(errors_dist_stand_dev/1000)
223 print >> log_file, '\nVariation coefficient CV distance error: %2f' %errors_dist_CV
224 print >> log_file, '\n50%%percentile distance error: \t\t %1f m/point\n75%%percentile distance error: \t\t %1f m/point\
225 \n90%%percentile distance error: \t\t %1f m/point\n95%%percentile distance error: \
226 \t\t %1f m/point' %(err_percentiles[0],err_percentiles[1],err_percentiles[2],err_percentiles[3])
227 #print >> log_file, '\nMean Marchal distance error: \t\t\t %1f m/point' %(errors_dist_meanM/1000)
228 #print >> log_file, '\nStandard deviation Marchal distance error: \t %1f m/point' %(errors_dist_stand_devM/1000)
229 #print >> log_file, '\nVariation coefficient CV Marchal distance error: %2f' %errors_dist_CVM
230 #print >> log_file, '\n50%%percentile Marchal distance error: \t\t %1f m/point\
231 #\n75%%percentile Marchal distance error: \t\t %1f m/point\n90%%percentile Marchal distance error: \t\t %1f m/point\
232 #\n95%%percentile Marchal distance error: \t\t %1f m/point' %(err_percentilesM[0],err_percentilesM[1],\
233 #err_percentilesM[2],err_percentilesM[3])
234 print >> log_file, '\nMean length index: \t\t\t %3f ' %lengthindexes_mean
235 print >> log_file, '\nStandard deviation length index: \t %3f ' %LI_stand_dev
236 print >> log_file, '\nVariation coefficient length index: \t %3f ' %LI_CV
237 print >> log_file, '\n50%%percentile length index: \t\t %3f\n75%%percentile length index: \t\t %3f\
238 \n80%%percentile length index: \t\t %3f\n90%%percentile length index: \t\t %3f\n95%%percentile length index: \t\t %3f\
239 % (LI_percentiles[0],LI_percentiles[1],LI_percentiles[2],LI_percentiles[3],LI_percentiles[4])
240 print >> log_file, '\nMean match index: \t\t\t %3f' %matchindexes_mean
241 print >> log_file, '\nStandard deviation match index: \t %3f ' %MI_stand_dev
242 print >> log_file, '\nVariation coefficient match index: \t %3f ' %MI_CV
243 print >> log_file, '\n5%%percentile match index: \t\t %3f\n10%%percentile match index: \t\t %3f\
244 \n20%%percentile match index: \t\t %3f\n25%%percentile match index: \t\t %3f\n50%%percentile match index: \t\t %3f\
245 % (MI_percentiles[0],MI_percentiles[1],MI_percentiles[2],MI_percentiles[3],MI_percentiles[4])
246 print >> log_file, '\nLength index in range [%2f - %2f]: \t %d matches of %d (%.1f%%)\
247 % (LImin,LImax,n1, len(mytraces.cols.routes), float(n1)/float(len(mytraces.cols.routes))*100)
248 print >> log_file, '\nMatch index in range [%2f - %2f]: \t %d matches of %d (%.1f%%)\
249 % (MImin,MImax,n2, len(mytraces.cols.routes), float(n2)/float(len(mytraces.cols.routes))*100)

```

Figura A.8: creazione del file di report e riepilogo (1)

Appendice

```
250 print >> log_file, '\nTotal distance travelled: \t\t %.1f km' %tot_length
251 print >> log_file, 'Total distance travelled on bikeways: \t %.1f km (%.1f%%)' %(sumbike,sumbike/tot_length*100.0)
252 print >> log_file, 'Mean distance travelled on bikeways: \t %.1f m/route' %meanbike
253 print >> log_file, 'Routes containing bikeways: \t\t %d of %d (%.1f%%)\
254 %(nbike,len(mytraces.cols.routes), float(nbike)/float(len(mytraces.cols.routes))*100)
255 print >> log_file, '\nLoading time: \t\t\t %.1f seconds %s' %(t1-t0,time.strftime("[%H h %M m %S s]",\
256 time.gmtime(t1-t0)))
257 print >> log_file, 'Cleaning data time: \t\t\t %.1f seconds %s' %(tc-t1,time.strftime("[%H h %M m %S s]",\
258 time.gmtime(tc-t1)))
259 print >> log_file, 'Processing data time: \t\t\t %.1f seconds %s' %(t2-tc,time.strftime("[%H h %M m %S s]",\
260 time.gmtime(t2-tc)))
261 print >> log_file, '\nTotal execution time: \t\t\t %.1f seconds %s' %(t2-t0,time.strftime("[%H h %M m %S s]",\
262 time.gmtime(t2-t0)))
263 print >> log_file, '_____'\
264 log_file.close()
265
```

Figura A.9: riepilogo (2) e salvataggio del file di report

```
266 mytraces.export_csv(outputfilename+'_matching.csv',sep=',',
267 attrconfigs = [mytraces.length,
268 mytraces.length_bikeway,
269 mytraces.length_mindist01,
270 mytraces.durations_gps,
271 mytraces.durations_est,
272 mytraces.matchindexes,
273 mytraces.lengthindexes,
274 mytraces.times_match,
275 mytraces.errors_dist
276 ])
277
278 mytraces.export_traces_xml(outputfilename+'_matching.xml')
279
280 mytraces.export_points_to_shape(outputfilename+'_points')
281 mytraces.export_routes_to_shape(outputfilename+'_routes')
282 mytraces.export_routes_mindist_to_shape(outputfilename+'_mindistroutes')
283
284 print open('log.txt').read()
```

Figura A.10: export degli output, creazione degli shapefiles dei GPS points e dei routes identificati, e apertura del file di report