

Soluzione innovativa per la mosaicatura di ortofoto mediante tecniche di programmazione dinamica.

Luca Menci*

(*) menci software S.r.l.,
via Martiri di Civitella, 11
52100 Arezzo
luca@menci.com

Riassunto

La mosaicatura automatizzata di immagini è certamente la problematica più sentita nell'insieme dei processi di generazione automatica delle ortofoto.

La questione è tanto più complessa tanto più bassa è la scala dei fotogrammi da mosaicare.

Infatti la disparità delle prese è esaltata nelle foto realizzate ad una bassa quota di volo e diviene un problema di difficile risoluzione specialmente in contesti urbani.

L'algoritmo proposto offre la soluzione per la cucitura di fotogrammi mediante tecniche di programmazione dinamica basate sulla ottimizzazione dei percorsi.

Infatti le linee di taglio vengono scelte come percorso minimo ottimizzato rispetto al massimo coefficiente di correlazione tra le due o più immagini da unire in mosaico.

I percorsi di taglio vengono calcolati partendo da una base approssimata calcolata mediante diagramma di Voronoi.

Questa soluzione, implementata nella nuova stazione cartografica multisensore, è stata sperimentata con successo sia in termini di risultati operativi sia in termini di tempi di calcolo, anche simulando condizioni di orientamento non ottimali sia in contesti urbani che extra-urbani.

Abstract

Automatic mosaic of images is sure one of the most important problem in all the processes of automatic generation of the ortofoto. The issue is much more complex much lowest is the scale of the photograms to mosaic. In fact the disparity of the photos is exalted in the photos realized at low quota flight and becomes a problem of difficult resolution especially in city contexts. The proposed algorithm offers the solution for the fusion of photograms using techniques of dynamics programming based on the distances optimization. In fact the cut lines are chosen as the minimal distance optimized regarding the maximum coefficient of correlation between the two or more images to join in mosaic. The cut distances are calculated starting from one approximate base throught the diagram of Voronoi. This solution, implemented in the new cartographic multisensor station, has been experimented succesfully, also simulating not optimal conditions of external orientation even in city contexts or country side.

Introduzione

Con il rapido sviluppo delle tecnologie digitali di fotorestituzione e la maturazione dei sistemi informativi territoriali, l'immagine ortorettificata è divenuta il più utilizzato tra i prodotti cartografici. L'interesse nel generare immagini di crescente qualità e risoluzione determina la necessità di mettere a punto tecnologie in grado di automatizzare il più possibile i processi di produzione. Uno degli aspetti principali nella realizzazione di mosaici di ortofoto è costituito dall'assemblaggio delle diverse immagini. Ci occupiamo in questa sede della scelta ottimale della linea di ritaglio per la giunzione automatica di più fotogrammi ortorettificati mediante un algoritmo da noi messo a punto, basato sull'interazione tra metodi di image-matching e tecniche di programmazione dinamica.

Algoritmo PPD

L'algoritmo proposto, PPD come acronimo di Percorsi mediante Programmazione Dinamica, nasce con l'intento di ovviare al concreto problema di ritagliare le immagini ortoproiettate, al fine di poterle accostare l'una alle altre senza che sia evidente la linea di separazione. La gran parte delle soluzioni sino a qui proposte si basano sulla sfumatura progressiva mediante tecniche di *alpha blending*, metodo che evidenzia i propri limiti con l'uso di fotogrammi di bassa quota laddove sono presenti i fenomeni di disassamento prospettico rispetto allo zenith; oppure mediante sistemi interattivi di ritaglio manuale.

La nostra soluzione si basa anzitutto sulla determinazione della miglior linea spezzata teorica di taglio, per poi operare nell'intorno di tale linea, mediante un articolato processo di programmazione dinamica, la scelta del più conveniente percorso sulla base di specifiche funzioni di costo.

In primis per ciascun fotogramma viene individuato il luogo di punti che meglio soddisfa i principi di ortoproiezione. Si privilegia naturalmente la zona centrale del fotogramma o meglio l'area in prossimità del punto principale.

Nel caso di foto aeree e comunque in tutti i casi in cui la scala media dei fotogrammi è sufficientemente omogenea, possiamo considerare come ottima la suddivisione del piano di proiezione in porzioni, ciascuna costruita attorno alla proiezione del punto principale.

Le celle che questa suddivisione genera, sono tali che ciascun punto che vi appartiene è più vicino a quel punto principale piuttosto che a ciascun altro. Una ripartizione del piano con questo criterio si effettua mediante il cosiddetto "Diagramma di Voronoi".

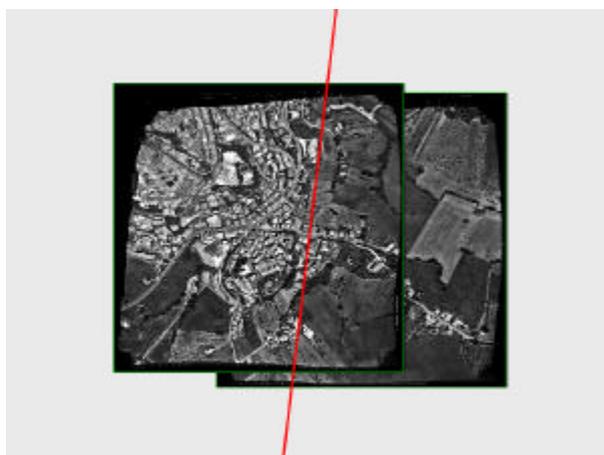


Figura 1 - Diagramma di Voronoi per due fotogrammi

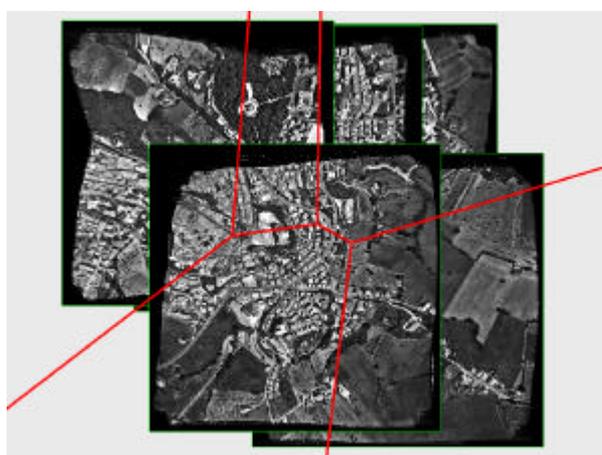


Figura 2 - Diagramma di Voronoi per 5 fotogrammi

La suddivisione in celle di Voronoi consente di ottenere ottimi risultati sia con l'uso di una coppia (Fig. 1) che di un numero maggiore di fotogrammi (Fig. 2).

Una volta individuate le linee in prossimità delle quali vogliamo tracciare il ritaglio, consideriamo gli estremi di queste linee, sia quelli interni alla zona di mosaico, sia quelli appartenenti al confine esterno. Nel caso esemplificativo di quattro fotogrammi disposti a coppie sovrapposte i punti interni 1 e 2 sono prossimi al baricentro dell'area complessiva e sono tali da essere approssimativamente allineati sulla verticale e sull'orizzontale, con i punti A, B, C e D, collocati sul contorno così come mostrato in figura 3.

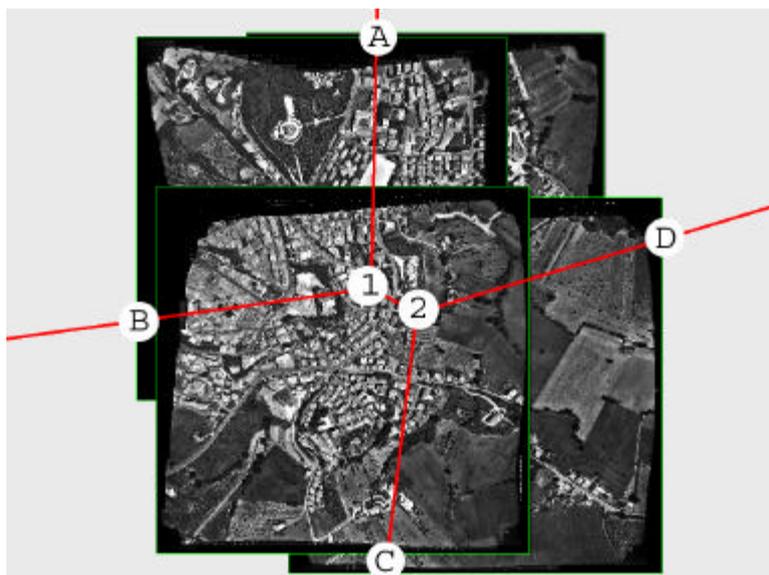


Figura 3 - Distribuzione dei nodi con quattro fotogrammi sovrapposti a due a due

Sono così definiti i punti di partenza e di arrivo di ciascuna spezzata.

Dobbiamo adesso ottimizzare il percorso così da minimizzare la visibilità del taglio tra un fotogramma e l'altro.

L'idea è di privilegiare i percorsi che attraversino le zone di massima similitudine tra i due fotogrammi confinanti. A questo scopo viene calcolata una mappa di similitudine sulla base dei coefficienti di correlazione della sovrapposizione delle immagini che si condividono ciascun segmento. La mappa, normalizzata su valori da 0 a 255, contiene coefficienti più elevati nelle zone in cui le due ortofoto sono maggiormente simili. Tipicamente si ha un alto coefficiente di correlazione nelle zone pianeggianti e senza vegetazione in ambito extra-urbano, al centro delle strade e nelle piazze, in ambito urbano; la mappa si presenta come un'immagine in toni di grigio con evidenziate in chiaro le aree più omogenee.

Si tratta adesso di scegliere il percorso più efficiente che privilegi il passaggio al di sopra di elementi caratterizzati da un alto coefficiente di correlazione: questo viene fatto mediante metodi di programmazione dinamica.

La definizione del contorno mediante programmazione dinamica può essere formulata come un problema di ricerca di grafo, dove l'obiettivo è quello della ricerca del percorso più "economico" tra un nodo di partenza e un nodo di arrivo. Nel nostro caso, il nodo di partenza e di arrivo, così come ogni nodo del percorso sono costituiti dai pixel della mappa. Tra un pixel e i suoi otto adiacenti è creato idealmente un lato di congiunzione (*edge*). Viene definito *costo cumulativo* il costo accumulato nel percorso che conduce dal pixel di partenza a quello di arrivo nel passaggio attraverso gli edge che collegano i vari pixel con gli adiacenti.

Vengono poi definite funzioni di costo che caratterizzano l'attraversamento di un edge; per ogni nodo vengono espresse tutte le possibili combinazioni ottenute muovendosi in tutte le direzioni. Il metodo applicato è di tipo *euristico*. Necessitiamo di un'euristica per decidere se interrompere o meno la complessa operazione della ricerca di un percorso. In generale, se non conosciamo

esattamente quanto siamo lontani dalla soluzione, il problema di stabilire se è il momento di cambiare strada o meno è piuttosto complesso. Quando ci riferiamo ad ogni nodo del grafo e quindi ad ogni pixel della mappa, siamo in grado di stabilire quanto siamo distanti dal risultato e quanto abbiamo speso per giungere al punto in cui siamo. Abbiamo già detto che al passaggio attraverso ogni edge è possibile associare un costo. Vengono così assegnati costi diversi nel passaggio attraverso edge orizzontali, verticali od obliqui, si definiscono inoltre le condizioni d'inalicabilità. Quando vogliamo valutare l'agibilità di un certo percorso, non dobbiamo far altro che sommare i costi accumulati nelle transizioni attraverso gli edge e negli stazionamenti sui nodi, determinati, a loro volta, dal coefficiente di correlazione.

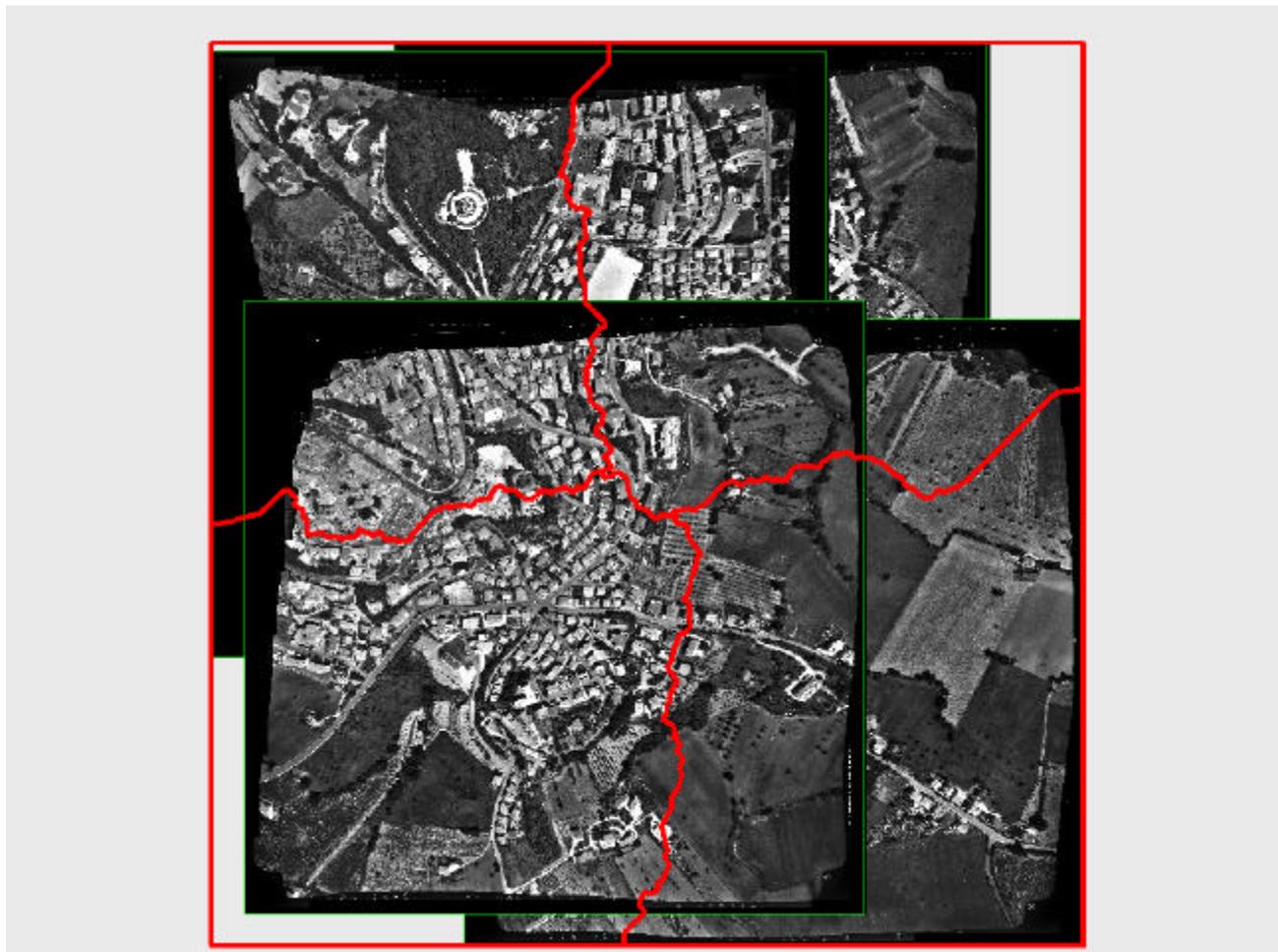


Figura 4 - Linee di taglio calcolate mediante l'algoritmo PPD proposto

La ricerca del percorso ottimale (*pathfinding*), è un problema importante in molte applicazioni, incluse la pianificazione del traffico, la movimentazione di robot, simulazioni militari e videogiochi. Il *pathfinding* consiste nella ricerca del “migliore” percorso per passare da un punto ad un altro. La funzione che determina la scelta ottimale è a variabili multiple e può rispondere a criteri quali: percorso più breve, più sicuro, percorso di costo minimo, eccetera.

Per la ricerca del miglior percorso di solito viene usato l'algoritmo denominato A^* (*AStar*).

Nei casi consueti d'uso questo algoritmo funziona egregiamente senza ulteriori modifiche. L'efficienza di A^* è subordinata alla scelta di un'opportuna euristica in grado di determinare l'abbandono di una strada a vantaggio di un'altra. Se l'euristica è correttamente determinata e non sovrastimata, l'algoritmo conduce sempre ad una soluzione. Si può dimostrare [7] che in una griglia la complessità di determinazione dell'euristica varia con il quadrato della distanza del punto in cui ci troviamo dal punto di destinazione. Questo fa sì che nella nostra applicazione, essendo le distanze in gioco (in termini di pixel) molto grandi, la complessità della determinazione della soluzione sia

notevole. Il numero di possibili soluzioni diviene talmente grande da rendere A* praticamente ingestibile in termini di occupazione di memoria. Per questo motivo abbiamo messo a punto una tecnica di allocazione dinamica della memoria in modo da liberare progressivamente quanto allocato per le aree che sono già state calcolate. Un metodo simile, denominato IDA* viene già utilizzato con successo in alcuni videogiochi[5].

La complessità di calcolo cresce con il quadrato della distanza tra il punto di partenza e quello di arrivo ed è pertanto sensibile alla risoluzione delle ortofoto da mosaicare. All'aumentare del numero di fotogrammi aumenta il numero di segmenti da calcolare, ma diminuisce la loro lunghezza.

In generale i tempi di calcolo sono comunque contenuti in dieci/quindici secondi per ogni fotogramma aereo.

Risultati

L'algoritmo PPD, oggetto di deposito presso l'Ufficio Brevetti, è stato inserito come funzionalità all'interno della stazione fotogrammetrica digitale Z-Map della quale sfrutta appieno la capacità di roaming delle immagini di grandi dimensioni. La rapidità di calcolo è considerevole anche per l'istantaneo accesso ai pixel delle immagini; sono state effettuate prove in diverse condizioni operative in termini di risoluzione, di disposizione e numero di fotogrammi ortorettificati, ottenendo sempre buoni risultati. Allo stato attuale gli algoritmi di correlazione operano sui toni di grigio, ottenendo pertanto risultati equivalenti sia sulle immagini in toni di grigio che su quelle a colori. In futuro pensiamo di avvalerci anche delle informazioni cromatiche contenute nelle immagini a colori.

Bibliografia

- [1] K. Forbus, J. Mahoney, and K. Dill. *How qualitative spatial reasoning can improve strategy game AIs*. AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment, pages 35-40, 2001.
- [2] P. Hart, N. Nilsson, and B. Raphael. *A formal basis for the heuristic determination of minimum cost paths*. IEEE Trans. Syst. Sci. Cybernet., 4(2):100-107, 1968.
- [3] S. Johnson. *Wild Things*. Wired, pages 78-83, 2002. March issue.
- [4] R. Korf. *Depth-first iterative-deepening: An optimal admissible tree search*. Artificial Intelligence, 27(1):97-109, 1985.
- [5] R. Korf, M. Reid, and S. Edelkamp. *Time complexity of iterative-deepening A**. Artificial Intelligence, 129(2):199-218, 2001.
- [6] J. Laird and M. van Lent. *Human-level AI's killer application: Interactive computer games*. In AAAI National Conference, pages 1171-1178, 2000.
- [7] J. Pearl. *Heuristics: Intelligent search strategies*. In Addison-Wesley, 1984.
- [8] S. Rabin. *A* Aesthetic Optimizations*. Game Programming Gems. Charles River Media, pages 264-271, 2000.
- [9] B. Stout. *Smart moves: Intelligent path-finding*. Game Developer Magazine, (October):28-35, 1996.

