

Un nuovo approccio allo strip alignment per i dati Lidar da sistemi di rilievo aerei e terrestri

Giovanni Drera, Mauro Contrafatto e Paolo Girardi

3D Target srl, Brescia (Italy), info@3dtarget.it

Abstract. Gli algoritmi di strip alignment (SA) sono delle procedure che consentono di allineare delle nuvole di punti misurate in momenti e pose diversi da strumenti Lidar ad alte prestazioni. Lo SA permette di aumentare la compattezza delle nuvole finali correggendo i possibili errori introdotti dal sistema di navigazione inerziale (INS). In questo lavoro è mostrato un nuovo algoritmo di SA adottato da 3D Target, che consente di allineare dati provenienti da diverse sorgenti (droni, rilievi da auto o altro), senza vincoli geometrici (i.e., strip rettilinee) e con alte prestazioni.

1 Introduzione

In generale, i rilievi Lidar mobili su aree di grande dimensione vengono eseguiti tramite scansioni multiple, eseguite muovendo il sistema di misura su una traiettoria costituita da molteplici tratti sovrapposti (si veda la Fig. 1). Questo metodo permette di ottenere una maggiore densità di punti ed una maggiore varietà nella posa dei soggetti. Tuttavia, nel corso delle scansioni possono subentrare delle variazioni nella precisione della misura della traiettoria e dell'orientamento dello scanner; in questo caso, le diverse scansioni, che comunemente vengono chiamate *strip*, non sono sovrapposte, con una diminuzione della risoluzione finale del modello ottenuto ed una incertezza sulla loro effettiva posizione.

La generazione di una nuvola di punti avviene attraverso un processo di georeferenziazione: le osservazioni del Lidar sono sincronizzate con i dati del relativo sistema di navigazione inerziale (INS), che consiste in un ricevitore GNSS e in giroscopi ed accelerometri integrati in una IMU (Inertial Measurement Unit). La georeferenziazione, che mette in relazione i dati Lidar (x_{Lidar}) alle coordinate finali, è descrivibile dalla seguente formula:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_{INS} \\ y_{INS} \\ z_{INS} \end{pmatrix} + M_{Config} M_{IMU} \left(\begin{pmatrix} x_{BS} \\ y_{BS} \\ z_{BS} \end{pmatrix} + M_{BS} \right) \begin{pmatrix} x_{Lidar} \\ y_{Lidar} \\ z_{Lidar} \end{pmatrix}$$

dove M_{Config} , M_{IMU} e M_{BS} sono le matrici di rotazione definite dai sistemi di riferimento della INS, dalla configurazione specifica (montaggio della INS rispetto al dispositivo) e dal Lidar (parametri di *boresight*, BS). Lo SA agisce ottimizzando i parametri dipendenti dall'INS, ossia la posizione ($x_{INS}, y_{INS}, z_{INS}$) e l'orientamento (M_{IMU}) che descrivono la traiettoria della piattaforma di acquisizione durante il rilievo. Questi parametri hanno accuratezze variabili nel tempo; in particolare modo, i rilievi terrestri eseguiti da un veicolo in movimento presentano delle accuratezze estremamente variabili, per via delle ostruzioni alla visibilità dei satelliti GNSS costituite da edifici e alberi, o nel caso della percorrenza di un tunnel. Lo scopo dello SA è quindi la modifica della traiettoria, eseguita in modo da minimizzare le distanze relative fra le nuvole di punti in diversi istanti temporali. In assenza di riferimenti geografici assoluti, viene in genere indicato come relative SA. L'approccio allo SA proposto è affine al metodo degli *Iterative Closest Points* [1](ICP), basato sull'individuazione di coppie di punti fra le nuvole da allineare (*tie point*, TP). Una volta individuati i TP, la relativa distanza viene ridotta modificando la traiettoria all'intervallo temporale corrispondente. Questo metodo, sebbene approssimativo,

permette di ottenere un ottimo allineamento a fronte di un costo computazionale contenuto, necessario per via del grande numero di punti solitamente considerato (> 100 M). Per questo lavoro si è scelto di adottare la metrica della distanza punto-piano (*P2P norm* [1]) invece della norma euclidea, consentendo una convergenza più rapida ed una più ampia mobilità dei parametri di fit; l'algoritmo di SA agisce quindi su punti appartenenti a piani paralleli. Il processo si divide in due sezioni; una prima parte di analisi, che filtra e prepara i punti delle strip, ed una seconda parte di ottimizzazione, che modifica la traiettoria identificando i TP e minimizzando la distanza fra i piani corrispondenti.

2 Analisi e filtraggio dei dati

Le strip sono filtrate al fine di ottimizzare il processo di SA per le diverse situazioni (drone o ricognizione da veicolo). Sono rimossi i punti spuri, viene ridotto il numero totale di dati tramite subsampling e filtri Voxel selettivi, sono calcolati i vettori normali ed eliminati i punti con eccessiva curvatura. Tutti questi processi sono gestiti in multithreading e con ampio utilizzo di rappresentazioni ad albero (*KD-tree*). La gestione della geometria è particolarmente complessa; pur considerando le strip come semplici proiezioni 2D, rilievi diversi possono dare luogo a configurazioni complesse e molto generiche. La descrizione delle aree di interesse è stata fatta tramite l'unione[2] geometrica delle *Convex Hull* determinate dai punti associati ad ogni intervallo di traiettoria. Il risultato è una *Concave Hull* complessa, che permette di descrivere accuratamente sia situazioni di volo che rilievi da veicolo in presenza di edifici.

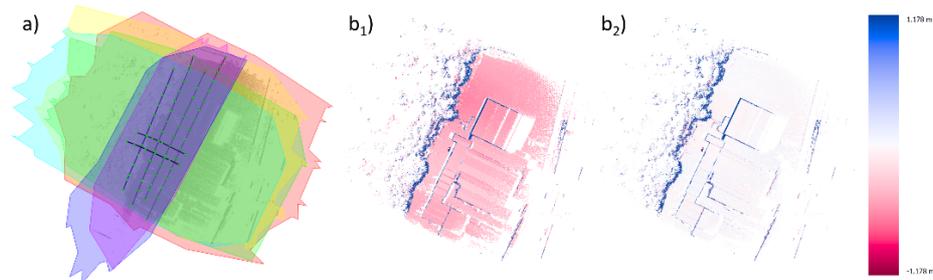


Fig. 1: a), report grafico dello SA eseguito su sei strips rilevate da drone; le linee rappresentano la traiettoria del drone con i punti pivot modificati (verde); in grigio, i TP. b), immagini proiettate delle differenze relative di una strip dal riferimento, prima (b₁) e dopo lo SA (b₂).

3 Risultati

I risultati dell'analisi sono usati dallo SA per individuare aree di sovrapposizione relativa, dove sottogruppi di punti e relativi piani sono allineati tramite discesa del gradiente (BFGS). L'errore calcolato della traiettoria ad ogni istante temporale viene considerato durante il processo. Alcuni risultati sono mostrati in Fig. 1, dove è mostrata l'ottimizzazione di sei strip da rilievo aereo. Grazie alla gestione ottimizzata di memoria, geometrie e grazie all'uso del multithreading, l'intero processo di SA in questo caso richiede circa 180 s su un notebook standard, operando su un dataset di 40M punti.

Riferimenti bibliografici

1. Glira P. et al, *PFG 4* (2015), 0275-0289
2. Boost C++ library, <https://www.boost.org/>