



Ricerca di Sistema elettrico

Sistema di monitoraggio aereo per lo smart district: elaborazione dati, ausilio al pilota, sviluppo sensori

S. Taraglio, L. Blasi, G. Cupertino, C. Moriconi, V. Nanni, S. De Vito,
F. Formisano

Sistema di monitoraggio aereo per lo smart district: elaborazione dati, ausilio al pilota, sviluppo sensori

S. Taraglio, L. Blasi, G. Cupertino, C. Moriconi, V. Nanni, S. De Vito, F. Formisano

Settembre 2018

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dello Sviluppo Economico - ENEA

Piano Annuale di Realizzazione 2017

Area: Efficienza energetica e risparmio di energia negli usi finali elettrici e interazione con altri vettori energetici

Progetto: D.6 Sviluppo di un modello integrato di smart district urbano

Obiettivo: d – Sicurezza infrastrutture critiche e monitoraggio aereo dello Smart District – Sotto-obiettivo d.2: Monitoraggio aereo dello smart district

Responsabile del Progetto: Claudia Meloni, ENEA

Indice

SOMMARIO.....	4
1 INTRODUZIONE.....	5
2 DESCRIZIONE DELLE ATTIVITÀ SVOLTE E RISULTATI.....	6
2.1 ALGORITMI PER LA RICOSTRUZIONE TRIDIMENSIONALE DELLA SCENA.....	6
2.1.1 <i>Teoria della structure from motion</i>	7
2.1.2 <i>Confronto tra VisualSFM, MicMac e Colmap</i>	11
2.2 INTERFACCE SEMPLIFICATE PER LA VISUALIZZAZIONE DI MODELLI TRIDIMENSIONALI IN PAGINE WEB.....	13
2.3 ALGORITMI PER LA SEPARAZIONE DI VELIVOLI.....	17
2.3.1 <i>L’algoritmo di separazione – teoria</i>	19
2.4 STUDIO DELLA PORTABILITÀ DI ULTERIORI SENSORI CHIMICI A BORDO DEL DRONE.....	21
2.4.1 <i>Adattamento HW/FW all’infrastruttura finalizzato al test funzionale.</i>	21
2.4.2 <i>Test case funzionale</i>	23
2.4.3 <i>Adattamento sistema di calibrazioni sensori</i>	23
2.4.4 <i>Eventuali estensioni della dotazione sensoristica</i>	24
2.5 CAMPAGNE SPERIMENTALI E TEST DI SISTEMA.....	25
2.5.1 <i>Campagne dati video e termografie</i>	25
2.5.2 <i>Campagna sperimentale raccolta dati chimici in Casaccia</i>	28
2.5.3 <i>Campagne di test per il payload di interfacciamento con i sensori aggiuntivi</i>	29
2.5.4 <i>Collaudo algoritmo sotto simulatore ROS</i>	31
3 CONCLUSIONI.....	34
4 RIFERIMENTI BIBLIOGRAFICI.....	36
5 ABBREVIAZIONI ED ACRONIMI.....	38
6 INDICE DELLE FIGURE.....	40

Sommario

Questo report presenta le attività del Piano Annuale di Realizzazione 2017 (PAR) relative al sotto obiettivo “Monitoraggio aereo dello smart district” dell’obiettivo “Sicurezza infrastrutture critiche e monitoraggio aereo dello Smart District” del progetto “Sviluppo di un modello integrato di Smart District urbano”.

Le attività relative al precedente Piano Annuale di Realizzazione 2016 hanno portato alla realizzazione di un dimostratore per il monitoraggio energetico degli edifici energivori e per quello ambientale per ciò che attiene alla qualità dell’aria; tale dimostratore si basa sulla tecnologia degli aeromobili a pilotaggio remoto multi elica usualmente indicati con il termine drone.

Esso è equipaggiato con due diversi *payload* al variare della missione di monitoraggio. Il primo è una *suite* di sensori chimici sensibili ad alcuni inquinanti dell’aria, il secondo è un sistema composto da una termocamera e da una telecamera video, accoppiate. Tramite questi due *payload* è possibile monitorare le prestazioni energetiche degli edifici sia dal punto di vista delle dispersioni termiche, che da quello della produzione di inquinanti legati alle attività energetiche quali il riscaldamento.

Nel Piano Annuale di Realizzazione 2017 le attività si sono indirizzate verso l’elaborazione dei dati forniti dai sensori a bordo del drone, in modo da renderli più facilmente fruibili dall’utilizzatore finale, con particolare riferimento al *payload* termico/visibile. Allo stesso tempo è stata sperimentata l’applicazione di un algoritmo per il mantenimento della separazione tra aeromobili sviluppata in ENEA allo scopo di fornire supporto al pilota del drone in ambiente urbano. Ulteriori attività sono state indirizzate allo sviluppo di sensori chimici aggiuntivi per il *payload* del drone.

Le attività si sono sviluppate dunque lungo tre filoni principali: lo sviluppo di algoritmi per l’elaborazione dei dati (soprattutto iconografici), lo sviluppo di un ausilio al pilota e lo studio di portabilità di nuovi sensori chimici. Questi tre filoni sono stati affiancati da prove sperimentali in campo. Il primo ed il secondo argomento sono stati principalmente seguiti dal Laboratorio di Robotica dell’ENEA (DTE-SEN-IDRA) in collaborazione con il Dipartimento di Ingegneria Elettronica dell’Università di Roma “Tor Vergata” ed il terzo dal Laboratorio ENEA Dispositivi Innovativi (DTE-FSN-DIN).

Nel primo filone le attività si sono incentrate sugli algoritmi detti di *structure from motion* (SfM), ovvero struttura dal movimento, cioè quella classe di algoritmi di computer vision che permettono di inferire la tridimensionalità degli oggetti mettendo in relazioni diverse immagini riprese da punti di vista diversi degli oggetti stessi; altro aspetto sviluppato è stato quello di una presentazione dei modelli tridimensionali tramite una semplice interfaccia di tipo web nel Decision Support System (DSS, sistema di supporto alle decisioni) sviluppato nel sotto obiettivo “DSS protezione e sicurezza delle infrastrutture critiche” condotto principalmente dal Laboratorio ENEA Analisi e Protezione delle Infrastrutture Critiche (DTE-SEN-APIC).

Nel secondo filone è stato studiato ed implementato l’adattamento di un algoritmo per la separazione tra aeromobili al problema di suggerire una traiettoria di volo sicura al pilota del drone al fine di evitare un eccessivo avvicinamento agli edifici da monitorare o ad eventuale altro traffico aereo nella zona oggetto del monitoraggio.

Nel terzo filone sono proseguite le attività per equipaggiare il drone con ulteriori sensori chimici provenienti da un sistema portatile di misura recentemente sviluppato in ENEA con finalità simili.

Le attività sopra descritte sono state integrate con attività sperimentali per il test operativo delle soluzioni sviluppate. La ricostruzione tridimensionale è stata valutata in due missioni di volo, mentre l’algoritmo per il suggerimento di una traiettoria sicura è stato valutato nell’ambiente di simulazione sviluppato nel corso delle attività del PAR 2016. E’ stata infine condotta una campagna di monitoraggio ambientale all’interno del Centro Ricerche Casaccia, per verificare le funzionalità della soluzione di monitoraggio dell’aria.

1 Introduzione

Una delle attività relative allo Smart District è quella del monitoraggio, cioè la rilevazione periodica e sistematica di parametri chimici e fisici, mediante appositi strumenti, allo scopo di controllarne la situazione puntuale o l'andamento temporale. In questo Rapporto Tecnico si descrivono le attività svolte nell'ambito del Piano Annuale di Realizzazione 2017 intorno ad un dimostratore basato sulla tecnologia dei droni multi elica, volto al monitoraggio energetico esterno degli edifici energivori e a quello ambientale per ciò che attiene alla qualità dell'aria [1].

Il monitoraggio esterno degli edifici per le eventuali dispersioni di calore e la loro misura è un evidente strumento per migliorare le caratteristiche energetiche di un fabbricato, meno evidente è l'utilizzo di una analisi della qualità dell'aria. Il monitoraggio della qualità dell'aria può essere considerato come un modo integrativo per lo studio energetico degli immobili. Infatti la qualità dell'aria dipende fortemente dalle attività energetiche dell'uomo (riscaldamento, autotrazione, etc.) ed è quindi intrinsecamente legata alle caratteristiche energetiche degli edifici: migliorare l'isolamento termico in una costruzione porta a minori inquinanti atmosferici derivanti dal suo riscaldamento; allo stesso modo, prevedere modi più efficienti di trasporto urbano può ridurre il traffico ed il conseguente inquinamento. Quindi la misura della qualità dell'aria permette una descrizione più accurata del quadro energetico globale dello *Smart District*, aggiungendo ulteriori variabili quantitative e qualitative e consentendo una descrizione più accurata della situazione generale.

Nel corso del Piano Annuale di Realizzazione 2017 sono state quindi realizzate le attività riguardanti l'elaborazione dei dati provenienti dalla piattaforma volante teleguidata da terra, volta ad una presentazione al decisore più fruibile e completa. Altro filone ha riguardato lo studio e l'implementazione di un algoritmo di supporto al pilota del drone allo scopo di suggerire possibili traiettorie di volo che mantengano distanze di sicurezza tra il velivolo e la struttura monitorata ed eventuali altri sistemi RPAS (Remotely Piloted Aerial System, droni).

Sono altresì proseguite le attività per equipaggiare il drone con ulteriori sensori chimici provenienti da un sistema portatile di misura recentemente sviluppato in ENEA con finalità simili.

Le attività del Piano Annuale di Realizzazione 2017 si sono svolte con la seguente articolazione:

- studio e confronto di algoritmi per la ricostruzione tridimensionale della scena a partire da più immagini (Structure from Motion, SfM);
- studio e realizzazione di interfacce semplificate per la visualizzazione di modelli tridimensionali in pagine web HTML;
- studio e realizzazione di un algoritmo per il suggerimento di traiettorie sicure al pilota del drone;
- studio della portabilità di ulteriori sensori chimici a bordo del drone;
- campagne sperimentali e test del sistema.

2 Descrizione delle attività svolte e risultati

2.1 Algoritmi per la ricostruzione tridimensionale della scena

Il sistema volante è in grado di utilizzare due diversi *payload*. Il primo è un insieme di sensori chimici che misurano la concentrazione di un certo numero di inquinanti, registrandone i valori in associazione con la posizione GPS del punto dove si è effettuata la misura. Il secondo è una coppia telecamera - termocamera che restituisce immagini e termografie.

La visualizzazione delle misure chimiche sul DSS è effettuata visualizzando i punti della traiettoria percorsa dal drone e permettendo l'apertura di una opportuna finestra che mostra i valori chimici per il punto selezionato.

Per ciò che riguarda invece la visualizzazione dei dati iconografici delle due camere, per rendere più fruibile l'elevato numero di immagini raccolte dal drone, si è fatto ricorso alla Structure from Motion (SfM) ovvero a quegli algoritmi fotogrammetrici che permettono la ricostruzione di un modello tridimensionale di un dato soggetto a partire da una serie di immagini dello stesso. In questo modo si è sostituita alla molteplicità delle immagini l'unicità di un modello tridimensionale, richiamabile dal DSS e visualizzabile tramite una interfaccia web.

La fotogrammetria è l'arte, scienza e tecnologia dell'estrazione di informazioni geometriche tridimensionali a partire da immagini o da sensori di immagini. La SfM è quell'approccio in cui si utilizzano più inquadrature di un dato soggetto, riprese da una camera in movimento (di qui il nome), oppure più immagini riprese da più camere da diversi punti di vista. La triangolazione di punti corrispondenti sulle varie immagini permette poi la ricostruzione della tridimensionalità del soggetto ripreso. L'uomo coglie la tridimensionalità dell'ambiente circostanze sfruttando l'uso di due camere (gli occhi) e quindi da due punti diversi di ripresa (stereoscopia). Infatti chiudendo uno dei due occhi la tridimensionalità viene persa. Se però, pur usando un solo occhio, ci si muove la tridimensionalità riappare in quanto in modo inconscio si applica proprio la SfM.

Questa classe di algoritmi è recentemente tornata di attualità grazie alle aumentate disponibilità di capacità di calcolo nei computer odierni e, più in generale, allo sviluppo della computer vision.

Sul mercato sono presenti un certo numero di software commerciali dedicati alla ricostruzione tridimensionale, tra questi si possono ricordare Agisoft PhotoScan (3500 dollari USA), Pix4D Mapper (€6500 per uso perpetuo, €2600 un anno, €260 un mese), 3DF Zephyr (fino a €4750). Questi prodotti sono, tipicamente, realizzati da spinoff della ricerca universitaria ed hanno dei costi notevolmente elevati, in quanto il loro impiego è principalmente indirizzato al rilevamento tridimensionale professionale. Allo stesso tempo esistono anche soluzioni open source (ovvero liberamente utilizzabili) sempre provenienti da ambienti di ricerca e che rappresentano lo stato dell'arte della ricerca in questo campo. Esistono diversi articoli scientifici che comparano le soluzioni commerciali con quelle open source e complessivamente la qualità dei risultati è sovrapponibile; in alcuni casi gli algoritmi commerciali si comportano meglio in altri sono peggiori. Naturalmente una soluzione commerciale implica anche un servizio di supporto, ma la soluzione open source è spesso tecnologicamente più avanzata, pur implicando una gestione più complessa. Una ultima nota: almeno in un caso è stato verificato che il SW commerciale produce file temporanei con i medesimi nomi di uno di quelli open source, testimoniando il fatto che, di base, il SW open source viene anche utilizzato dalla soluzione commerciale!

Tra le soluzioni open source si citano VisualSfM [2] [3], MicMac [4] [5] e Colmap [6] [7] [8]. Il primo è stato sviluppato in un ambiente misto accademico industriale, con la presenza di Microsoft; esso presenta una interfaccia grafica e non è di facile utilizzo, oltre ad evidenziare alcuni limiti operativi, specialmente nel caso di immagini con basso contrasto. Il secondo è il risultato di anni di ricerca presso l'ENGS (Ecole Nationale des Sciences Geographiques) e presso l'IGN (Institut National dell'information Geographique et forestiere) entrambi francesi. Il software è sicuramente il più completo ma è utilizzabile solo tramite terminale a riga di comando, rimanendo così alquanto ostico; va qui notato che è tuttora in corso di sviluppo continuando ad aggiungere nuove funzionalità, in aggiunta esso è abbastanza ben documentato anche *on line*, aspetto

usualmente poco curato quando si tratta di soluzioni *open source*. Il terzo è sulla stessa falsariga del secondo, ma utilizzabile tramite una interfaccia grafica che permette un'interazione meno ruvida con il software. Fornisce ottimi risultati anche senza affinare particolarmente i parametri di elaborazione. La scelta è dunque ricaduta sull'uso dei due ultimi software accademici, che, tra l'altro, permettono di modificare a piacere parametri e funzioni per poterli adattare a situazioni diverse. Ad esempio MicMac è in grado di affrontare, oltre alla ricostruzione 3D di soggetti ripresi da droni, anche la fotogrammetria aerea di ampi superfici (es riprese da satelliti o aerei) e la ricostruzione di oggetti planari.

2.1.1 Teoria della structure from motion

Camera a foro stenopeico, parametri intrinseci, parametri estrinseci e distorsioni

Si consideri una camera a foro stenopeico (in inglese pin-hole camera), una camera dove i raggi passano attraverso un foro piccolissimo e si proiettano su di una superficie sensibile.

Un punto $M=(X,Y,Z,1)^T$ in coordinate omogenee in un sistema di riferimento globale ed il relativo punto $m=(x,y,1)$ sul piano immagine sono messi in relazione da una matrice di trasformazione proiettiva:

$$sm=PM$$

Dove s è un fattore di scala e P è una matrice proiettiva 3 x 4 che può essere decomposta come:

$$P=A[R|t]$$

Dove A è la matrice 3x4 dei parametri intrinseci della camera

$$A = \begin{bmatrix} f\alpha & \gamma & c_x & 0 \\ 0 & f\beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Con (c_x, c_y) è indicato il centro ottico della camera, f la lunghezza focale, con α e β le dimensioni fisiche del pixel e γ rappresenta l'inverso della tangente dell'angolo tra l'asse x e quello y del CCD (angolo di skew) nel caso non fossero ortogonali (usualmente lo sono e quindi $\gamma = 0$). Alla matrice A viene usualmente dato il nome di matrice intrinseca in quanto è descritta dai parametri intrinseci della camera, quelli, cioè, che rappresentano le caratteristiche ottiche della camera.

La matrice $[R|t]$ è una matrice che mette in relazione il sistema di coordinate della camera con quello globale, ovvero la posizione della camera t e la sua matrice di rotazione R :

$$[R|t] = \begin{bmatrix} R & t \\ \mathbf{0}_3^T & 1 \end{bmatrix}$$

essendo $\mathbf{0}_3^T$ il vettore nullo di tre elementi. A questa quantità viene usualmente dato il nome di matrice estrinseca, in quanto composta dai parametri estrinseci della camera: posizione e rotazione nel sistema di coordinate mondo.

Le camere normalmente utilizzate non sono a foro stenopeico, bensì possiedono un obiettivo composto da una serie di lenti, necessarie ad aumentare la quantità di luce raccolta dall'elemento sensibile. Queste lenti, pur realizzate con elevata precisione, introducono delle distorsioni che modificano la posizione di m sul CCD. Le distorsioni vanno opportunamente modellate per permettere una descrizione precisa del fenomeno proiettivo. Le distorsioni possono essere principalmente di due tipi: radiale o tangenziale, ma la prima è quella usualmente più importante essendo le lenti usualmente dei solidi di rotazione. Essa viene così descritta: il pixel corretto (x',y') è ottenuto da quello proiettato secondo il modello stenopeico (x,y) con la correzione:

$$\begin{aligned} x' &= x + L(r)x \\ y' &= y + L(r)y \end{aligned}$$

con $r^2=x^2+y^2$ ed $L(r)$ la funzione distorsione sviluppata in serie di Taylor usualmente con $L(r)\approx k_1r^2+k_2r^4$. I coefficienti k_1 e k_2 vengono considerati come parametri intrinseci aggiuntivi e vengono calcolati con un procedimento noto come calibrazione, si veda ad esempio [9].

Tra le attività condotte nel PAR2017 una è stata proprio indirizzata alla calibrazione della termo camera presente a bordo del drone, infatti l'ambito termico dei segnali registrati dalla camera implica delle difficoltà rispetto alla calibrazione di una telecamera video standard. Nel report RdS/PAR2017/065 [10] redatto dal Dipartimento di Ingegneria Elettronica dell'Università di Tor Vergata è descritta in dettaglio questa attività.

E' chiaro che il processo di ripresa di una scena con una camera (proiezione), porta comunque ad una riduzione da tre a due coordinate, ovvero offre una rappresentazione bidimensionale di una scena tridimensionale. La ricostruzione della perduta terza dimensione a partire da una singola immagine non è possibile, ma l'utilizzo di almeno due immagini permette il calcolo della terza dimensione.

Questo processo si basa sulla triangolazione di punti corrispondenti sulle due diverse immagini. Su questo aspetto si tornerà in seguito, per il momento si prenda per assodata la capacità di poter accoppiare pixel relativi al medesimo oggetto provenienti da due o più immagini.

Il Bundle adjustment

Gli algoritmi più recenti di SfM si basano sul cosiddetto *bundle adjustment* che può essere tradotto con 'aggiustamento del malloppo', si tratta di affinare contemporaneamente le coordinate 3D dei punti della scena, i parametri della trasformazione spaziale tra una posizione di ripresa e l'altra ed i parametri intrinseci delle camere tramite un algoritmo che ottimizzi la riproiezione di tutti i punti nello spazio.

Più formalmente si assume che n punti della scena 3D siano visti in m riprese e sia x_{ij} la proiezione del punto i nell'immagine j . Sia v_{ij} una variabile binaria uguale a 1 se il punto i è visibile nell'immagine j e 0 altrimenti. Sia poi la camera j parametrizzata da un vettore di parametri \mathbf{C}_j , che rappresentano i parametri intrinseci (lunghezza focale, punto principale, distorsione) ed estrinseci (posizione e orientamento), e ciascun punto 3D da un vettore \mathbf{P}_i , che contenga la posizione ignota del punto. Il bundle adjustment minimizza l'errore totale di riproiezione rispetto a tutti i punti 3D e rispetto ai parametri delle camere, ovvero:

$$\min_{\mathbf{C}_j, \mathbf{P}_i} \sum_{i=1}^n \sum_{j=1}^m v_{ij} d(\mathbf{Q}(\mathbf{C}_j, \mathbf{P}_i), \mathbf{x}_{ij})^2$$

dove $\mathbf{Q}(\mathbf{C}_j, \mathbf{P}_i)$ è il punto i come riproiettato nell'immagine j attraverso la camera di parametri \mathbf{C}_j , mentre \mathbf{x}_{ij} è il punto immagine vero, $d(x,y)$ è la distanza euclidea tra x e y .

Il *bundle adjustment* dunque è un'ottimizzazione che vuole estremizzare una funzione di costo operando contemporaneamente su moltissimi parametri, questa ottimizzazione però dipende fortemente da una opportuna inizializzazione, altrimenti l'algoritmo può facilmente essere vittima di un minimo locale nella funzione di costo, fallendo così l'ottimizzazione globale. Infatti la funzione risulta non essere convessa e la sua convergenza è, in condizioni realistiche, tipicamente verso un minimo locale.

Per evitare ciò si deve fornire al *bundle adjustment* una configurazione iniziale di punti 3D e di parametri delle camere il più possibile vicina a quella vera.

Configurazione iniziale di punti 3D

Consideriamo due immagini della stessa scena tridimensionale, ottenute da due punti di vista diversi (Figura 1), la geometria epipolare definisce la geometria proiettiva di due camere organizzate in un sistema stereoscopico. Dato un punto M nello spazio e le sue proiezioni sui piani focali delle camere m e m' , i tre punti definiscono un piano che interseca i due piani focali nelle cosiddette linee epipolari lm ed lm' mentre e ed e' sono gli epipoli, cioè quel punto immagine dove il centro ottico dell'altra camera si riproietta.

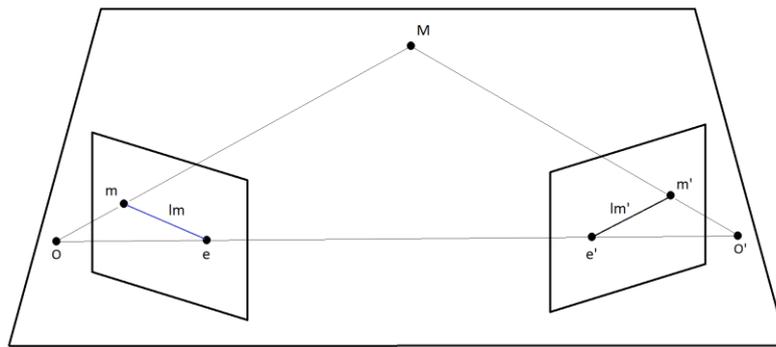


Figura 1. La geometria epipolare

Il cosiddetto vincolo epipolare afferma che se un punto si riproietta nell'immagine in m allora lo stesso punto avrà un punto corrispondente nell'altra immagine che giacerà su lm' . In termini matematici questo vincolo si esprime con una matrice 3 x 3 detta matrice fondamentale con la:

$$m'Fm=0$$

La matrice fondamentale F contiene i parametri intrinseci di entrambe le camere e la trasformazione rigida di una camera rispetto all'altra descrivendo la relazione tra le corrispondenze in termini di pixel dell'immagine. L'approccio standard per il calcolo della matrice fondamentale si basa sulla soluzione di un sistema omogeneo di equazioni nelle nove incognite della matrice:

$$Zf=0$$

dove $f = (f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9)^T$ e

$$Z = \begin{pmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{pmatrix}$$

Qualora fossero noti più di nove punti il sistema diviene sovra determinato e la soluzione va calcolata tramite un approccio ai minimi quadrati. Dalla matrice fondamentale è poi possibile derivare facilmente l'informazione tridimensionale, nota la matrice dei parametri intrinseci. A seconda che si consideri o meno un vincolo aggiuntivo dato dal rango pari a 2 della matrice F gli algoritmi utilizzati sono chiamati a 7 o a 8 punti. Naturalmente il vincolo sul rango della matrice viene considerato in seguito dai secondi. Va citata anche la possibilità di calcolare la matrice F con un approccio che fa uso di algoritmi genetici [11].

Una volta calcolata la matrice fondamentale è possibile derivare la rototraslazione compiuta dalla camera nel passaggio tra le due immagini. Nel caso in cui le immagini siano poi più di due è necessario passare all'allineamento delle successive immagini. Ciò può essere compiuto con diversi approcci: si può iterare il calcolo della matrice fondamentale mettendo in relazione ogni immagine con la precedente, si possono utilizzare i punti 3D già ricostruiti per il calcolo del punto di vista dell'immagine corrente, si può assemblare ricostruzioni parziali sulla base dei punti 3D corrispondenti. Un approccio completamente diverso è rappresentato dai metodi di fattorizzazione dove la matrice delle osservazioni, ovvero dei pixel, viene fattorizzata producendo una matrice di rotazione ed una di traslazione tramite l'utilizzo di algoritmi di SVD (Single Value Decomposition) per trattare i dati affetti da rumore.

Qualunque sia l'approccio utilizzato, al termine di questa registrazione incrementale delle posizioni spaziali (o fattorizzazione) si ottiene un modello di punti distribuiti nello spazio e relative posizioni delle camere. Questo rappresenta finalmente l'input per l'algoritmo di bundle adjustment che può operare così su una configurazione iniziale sufficientemente vicina alla soluzione per raffinarla ulteriormente.

Calcolo delle corrispondenze tra le immagini (SIFT)

Tutto quanto precedentemente esposto si basa sulla individuazione di punti corrispondenti tra coppie o più immagini. L'individuazione di una corrispondenza indica, in qualche misura, la capacità di riconoscimento di due punti su due immagini diverse come relativi alla stesso punto 3D, pur essendo le due immagini prese da punti di vista anche distanti e con parametri intrinseci che possono anche essere differenti, ad esempio per un cambio di focale (zoom).

La letteratura sull'estrazione di caratteristiche da immagini è vastissima, presentando i risultati di circa 50 anni di lavori sulla rilevazione di bordi, angoli, blob e regioni, su metodi basati sulla segmentazione, sul machine learning. La descrizione di tale corpus esula dalla presente relazione, per una survey su questi temi si rimanda a [12].

Il metodo di gran lunga oggi più utilizzato per il calcolo delle corrispondenze è il SIFT (Scale Invariant Feature Transform, trasformazione di caratteristiche invariante per trasformazioni di scala) [13] [14]. In buona sostanza il SIFT trasforma l'immagine in ingresso in una collezione di caratteristiche locali (rappresentate come vettori in un dato spazio vettoriale) che sono invarianti sotto cambiamento di scala, rotazione e traslazione e parzialmente invarianti per cambiamento di punto di vista o di illuminazione della scena.

Il processo si svolge in quattro passi successivi. Nel primo si estraggono delle caratteristiche (*feature*) candidate utilizzando l'estremizzazione locale delle DoG (Difference of Gaussian, differenze di convoluzioni con kernel gaussiani), in questo modo si osservano potenziali *feature* a diverse scale. In un secondo passo si analizzano le curvature principali, posizione e scala in modo da scremare le *feature* che non abbiano sufficiente contrasto nell'immagine e che non siano parte di un qualche contorno nell'immagine. Nel terzo passo si analizza un intorno di ogni *feature* in termini di gradiente locale di luminosità, giungendo a realizzare un istogramma delle principali direzioni del gradiente. Si arriva così ad un vettore di descrizione della *feature* di 128 elementi che alla fine del processo viene anche normalizzato per assicurare l'invarianza sotto cambi di illuminazione dell'intera immagine. L'ultimo passo è quello dell'accoppiamento tra immagini utilizzando come criterio la distanza euclidea, al di sotto di una data soglia due *feature* provenienti da due immagini diverse vengono indicate come corrispondenti allo stesso punto 3D nella scena.

Interpolazione del modello con superfici

Una volta ottenuta la nuvola di punti 3D tramite il *bundle adjustment*, il passo successivo e finale è quello di interpolare questa nuvola con una superficie e di colorare opportunamente quest'ultima. Quella usualmente utilizzata è la superficie di Poisson [15] che fornisce una superficie continua, ovvero priva di buchi che poi può essere drappeggiata con le immagini fornendo così un modello con una tessitura (*texture*). Questo approccio prevede la ricostruzione della superficie tramite un insieme di triangoli. Questi triangoli (*mesh*) vengono poi drappeggiati con le informazioni iconografiche provenienti dalle immagini delle quali si conoscono i punti di ripresa ed è quindi possibile scegliere la migliore immagine per qualunque triangolo facente parte della superficie triangolata.

Riassumendo

Dopo aver esaminato i punti salienti della procedura, con riferimento alla Figura 2 si riassumono in questo schema per una migliore comprensione:

1. acquisizione delle immagini;
2. estrazione delle caratteristiche in ciascuna immagine (SIFT);
3. comparazione ed accoppiamento delle caratteristiche su immagini diverse (SIFT);
4. calcolo di un modello 3D a nuvola di punti tramite approcci stereoscopici (matrice fondamentale);
5. ottimizzazione non lineare del modello di punti (*bundle adjustment*), completo della traiettoria della telecamera;

6. interpolazione della superficie (Poisson) e drappeggio delle immagini sulla superficie.

Al termine di questa procedura si avrà un modello di superficie usualmente descritto da un file di tipo PLY (PoLYgon file format), testuale o binario, nel quale vengono elencati i vertici ed i triangoli che formano la superficie insieme ad un file immagine che contiene il drappeggio.

La visualizzazione di questo file può essere fatta usando opportuni software di visualizzazione 3D, ad esempio: Meshlab, Blender, CloudCompare, etc.

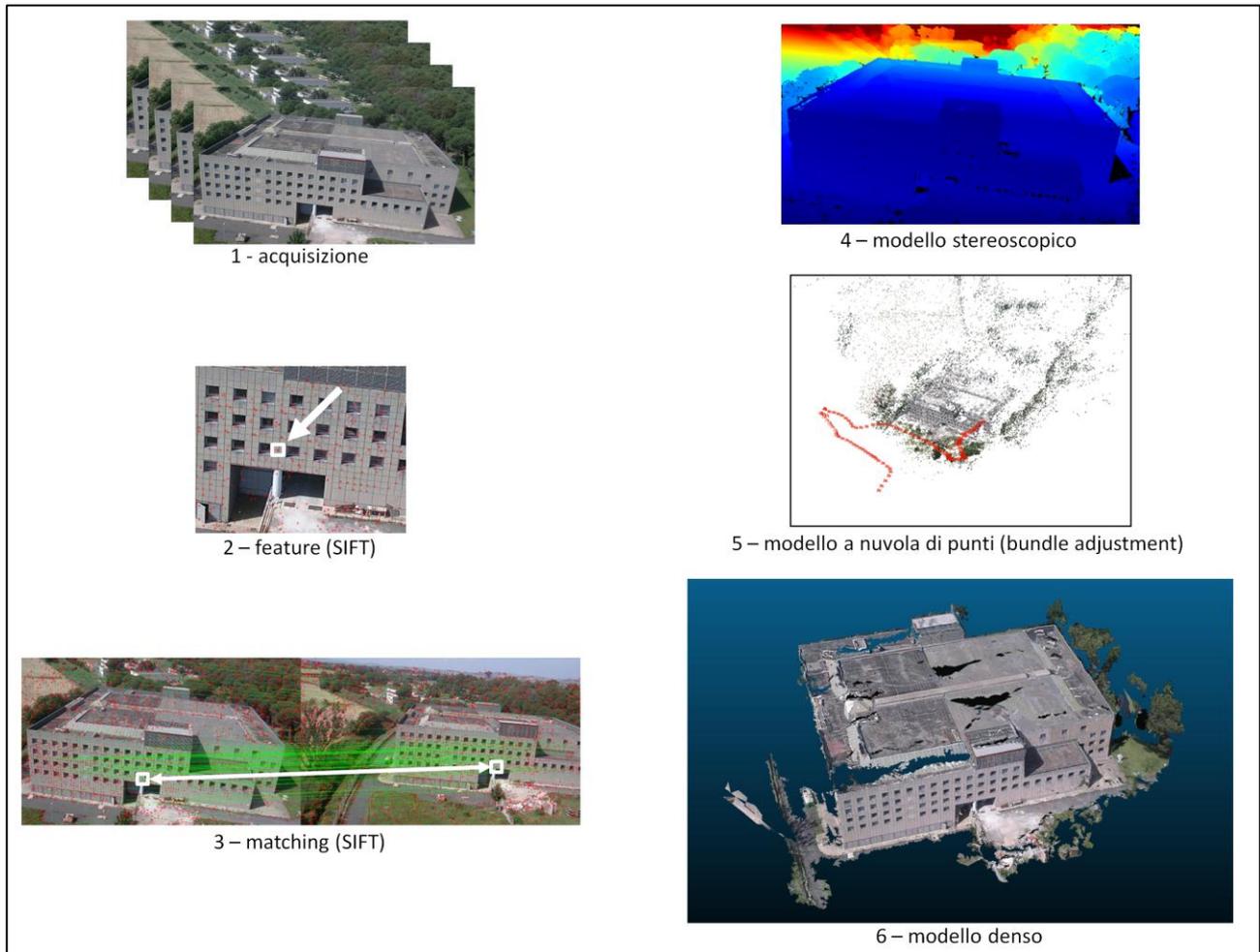


Figura 2. I sei passi necessari alla costruzione di un modello 3D

2.1.2 Confronto tra VisualSFM, MicMac e Colmap

Come detto nel paragrafo precedente pur esistendo molti prodotti commerciali per la realizzazione di modelli 3D a partire da immagini, ci si è concentrati sulle soluzioni open source. Le motivazioni di questa scelta sono molteplici: le soluzioni *open source*, per loro stessa natura, sono ‘esplorabili’ e sono più facilmente comprensibili eventuali comportamenti di errore; esse rappresentano usualmente lo stato dell’arte più aggiornato, essendo il prodotto di istituzioni di ricerca; hanno un vasto seguito e ciò implica l’esistenza di gruppi in rete che ne aiutano l’utilizzo (blog, mailing list, et.); la sostanziale uguaglianza con le soluzioni commerciali in termini di prestazioni; il rapporto qualità prezzo.

Esistono diversi articoli scientifici che comparano le soluzioni commerciali con quelle open source e complessivamente la qualità dei risultati è sovrapponibile; in alcuni casi gli algoritmi commerciali si comportano meglio in altri sono peggiori [16] [17] [18].

La soluzione commerciale è, per questo genere di applicazioni, costosa e, allo stesso tempo, implica un approccio a *black-box*, dove i risultati sono molto buoni ma, appunto, provengono da una scatola nera sulla quale non si può avere alcun controllo. La soluzione open source è spesso tecnologicamente più avanzata, implicando una gestione più complessa del software, ma totalmente aperta e verificabile in ogni passo della procedura.

I software open source che sono stati confrontati sono VisualSFM, MicMac e Colmap.

VisualSFM è un'applicazione con interfaccia grafica che implementa tutta la sequenza procedurale necessaria alla ricostruzione tridimensionale. E' temporalmente il primo dei software per la ricostruzione 3D e ciò è abbastanza evidente nell'uso e nei risultati ottenibili. E' abbastanza macchinoso e spesso non riesce a convergere su di un solo modello, ma al contrario ne realizza molti dello stesso soggetto, costringendo poi l'utente a molte manipolazioni per riportare i vari modelli in uno solo.

```
mm3d Tapioca MulScale *.jpg 350 -1
mm3d Tapas RadialStd ".*.jpg" Out=Cal1
mm3d Tapas AutoCal ".*.jpg" InCal=Cal1 Out=VisMM
mm3d Apericloud *.jpg VisMM Out=poscam.ply
mm3d SaisieMasqQT
mm3d C3DC BigMac *.jpg VisMM Masq3D=poscam_selectionInfo.xml
mm3d TiPunch C3DC_BigMac.ply Pattern=".*.jpg" Mode="BigMac"
mm3d Tequila ".*.jpg" VisMM "/C3DC_BigMac_mesh.ply" Out="F65VMMD8.ply"
```

Figura 3. Una tipica sequenza di comandi per una ricostruzione 3D in MicMac

Risultati molto soddisfacenti invece si raggiungono con uno sforzo nettamente minore con gli altri due software. MicMac è sicuramente quello che può affrontare il maggior spettro di applicazioni, spaziando senza problemi dalla ricostruzione 3D di edifici alla orto fotogrammetria e ad un'accuratezza planimetrica molto elevata. Esso è tutt'ora in fase di forte evoluzione ed è continuamente aggiornato ed 'allargato' ad ulteriori campi di applicazione da un numero importante di ricercatori soprattutto francesi. Il suo principale difetto, o la sua forza, è nell'essere una suite di differenti programmi richiamabili a riga di comando con vari parametri, ciò rende molto flessibile la gestione del flusso di lavoro, ma al contempo difficile, si veda a titolo di esempio la Figura 3. Da essa appare sicuramente evidente una certa predilezione per gli aperitivi da parte dei ricercatori francesi; entrando più nel dettaglio il programma Tapioca è quello responsabile di trovare le feature e farne il match attraverso le varie immagini, Tapas compie una calibrazione della telecamera, Apericloud crea una prima nuvola di punti 3D, SaisiemasqQT permette la selezione di parte della nuvola di punti per limitare a quanto di interesse, infine C3DC, Tipunch e Tequila si occupano del bundle adjustment, della interpolazione con una superficie e il drappeggio delle immagini su di essa.

Colmap è sempre una serie di software con i quali, però, si può interagire tramite un'interfaccia grafica, dove è possibile usare i valori di default oppure variare gli innumerevoli parametri in funzione dei risultati ottenuti. Ciò rende l'utilizzo di colmap molto più intuitivo anche se, di base, è sempre necessario sapere cosa si sta facendo.

I risultati che i due programmi offrono sono abbastanza sovrapponibili, tendenzialmente MicMac è metricamente più preciso mentre Colmap fornisce superfici più continue, ovvero con meno parti mancanti, si veda la Figura 4.

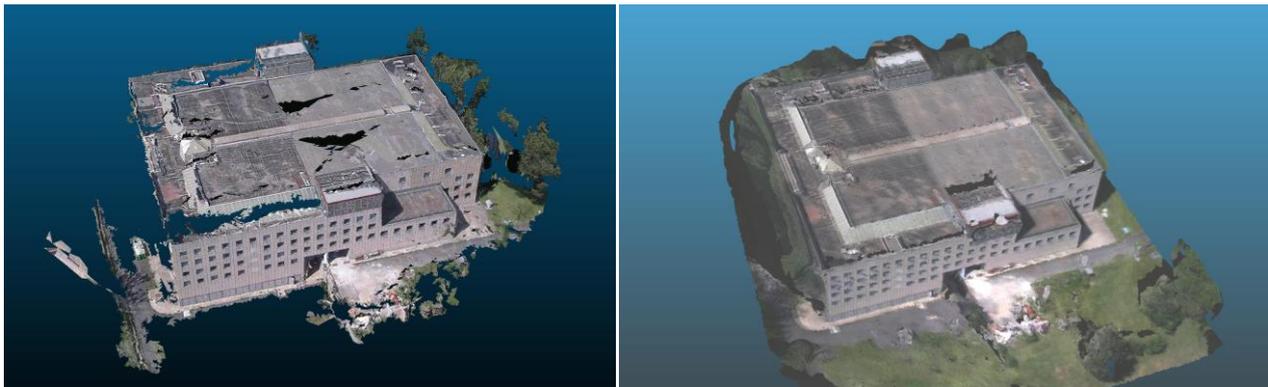


Figura 4. Il modello 3D di un edificio della Casaccia come ricostruito da MicMac (sinistra) e Colmap (destra)

Per ciò che riguarda i tempi di calcolo, questi tendono ad aumentare notevolmente all'aumentare del dettaglio dei modelli ricostruiti. Una elaborazione tipica per un input di 120 immagini circa può arrivare ad impiegare alcune ore. Per fortuna tutte e tre le suite software hanno un codice che può utilizzare le capacità delle moderne schede grafiche utilizzando la potenza di calcolo parallelo offerta. Ciò permette di ridurre i tempi a poco meno di un'ora (utilizzando una NVIDIA Quadro P2000, dotata di 1024 core di calcolo).

2.2 Interfacce semplificate per la visualizzazione di modelli tridimensionali in pagine web

Nelle attività del PAR2017 è stato realizzato un visualizzatore per modelli tridimensionali tramite pagina web in HTML. In questo modo la visualizzazione risulta estremamente semplificata nell'ambito del DSS (Decision Support System) sviluppato nell'obiettivo d.1.

Il dato di input per la ricostruzione 3D è, come detto, rappresentato da una collezione di molte immagini (dell'ordine almeno delle centinaia) o da filmati. Tale quantità di informazioni sono difficili da rendere disponibili all'interno di un GIS quale è il DSS, dove è di precipua importanza la sintesi, per poter offrire all'utente una visione d'insieme sincretica, si è quindi scelto di presentare all'utente direttamente il modello 3D. Naturalmente le immagini o i filmati di input rimangono sempre a disposizione per un eventuale controllo, ma non vengono ad appesantire la presentazione.

La soluzione scelta è quella nella quale il DSS possa richiamare una pagina web all'interno della quale viene visualizzato il modello 3D con la possibilità di interagire con esso in termini di movimentazione e display.

Lo strumento utilizzato è la suite software *three.js*, una libreria JavaScript che può operare all'interno di molteplici browser (Chrome, Mozilla, etc.). Essa è in grado di creare ed animare grafica tridimensionale all'interno di una pagina web senza utilizzare plugin del browser, ma utilizzando solo WebGL una API JavaScript per il rendering grafico una tecnologia web che fornisce grafica 3D accelerata direttamente in hardware ad un browser (anche su GPU, Graphical Processing Unit, ovvero i processori presenti su scheda grafica), senza dover installare plugin aggiuntivi o software aggiuntivo. In questo modo il peso del calcolo necessario all'animazione rimane sul client evitando un eccessivo traffico dati sulla rete e velocizzando la visualizzazione. Three.js è un software open source con licenza MIT, ovvero licenza Expat, cioè una licenza di software libero.

Three.js è composto da diversi pacchetti che permettono l'implementazione di una serie di caratteristiche quali: l'inserimento di telecamere, di controlli (*pan*, *tilt*, *zoom*, etc.), l'inserimento di oggetti descritti da primitive semplici geometriche (cubi, sfere, coni, piramidi, etc.), il caricamento di modelli complessi descritti da *mesh* in diversi standard, utilità per la manipolazione dei dati tridimensionali (matrici, quaternioni, campi di vista delle camere, etc.). Per ulteriori particolari si rimanda al sito [19].

La realizzazione del visualizzatore passa dunque attraverso la realizzazione di un file di tipo HTML che richiama al suo interno gli script necessari ad implementare le sue specifiche. In questo caso oltre allo script *three.js*, obbligatorio, vengono utilizzati altri tre script, uno per leggere i modelli tridimensionali descritti

con file OBJ, uno per il controllo della camera ed uno per la pubblicazione delle statistiche, vedi Figura 5. Come esposto nel paragrafo precedente, in realtà il modello ottenuto a valle della ricostruzione 3D è di tipo PLY, three.js è in grado di leggere tali modelli, ma si è preferito passare attraverso il tipo OBJ per una maggiore flessibilità ed affidabilità del codice.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    ... ..
  </head>
  <body>
    //=====>>>> 0. Load scripts & init
    <script src="/home/sergio/sw/three.js-master/build/three.js"></script>
    <script src="/home/sergio/sw/three.js-master/examples/js/loaders/OBJLoader.js"></script>
    <script src="/home/sergio/sw/three.js-master/examples/js/controls/OrbitControls.js"></script>
    <script src="/home/sergio/sw/three.js-master/examples/js/Detector.js"></script>
    <script src="/home/sergio/sw/three.js-master/examples/js/libs/stats.min.js"></script>

    <script>
      if ( ! Detector.webgl ) Detector.addGetWebGLMessage();
      init();
      function init() {
        ... ..
        scene = new THREE.Scene();
    //=====>>>> 1. Camera
        camera = new THREE.PerspectiveCamera( 75, window.innerWidth / window.innerHeight, 0.1, 2000 );
        scene.add( camera );
    //=====>>>> 2. Renderer [WebGL]
        renderer = new THREE.WebGLRenderer( { antialias: true } );
        scene.background = new THREE.Color( 0x0a6697 );
    //=====>>>> 3. Load OBJ file
        //----> texture
        var manager = new THREE.LoadingManager();
        var textureLoader = new THREE.TextureLoader( manager );
        var texture = textureLoader.load( './texture4096.png' );
        //----> model
        var loader = new THREE.OBJLoader( manager );
        loader.load( './visible.obj', function ( object ) {
          object.traverse( function ( child ) {
            if ( child instanceof THREE.Mesh ) {
              child.material.map = texture;
              THREE.GeometryUtils.center( child.geometry );
            }
          } );
          scene.add( object );
        }, onProgress, onError );
    //=====>>>> 4. Lights
        // Create an AmbientLight
        lightsource = new THREE.AmbientLight( 0xaaaaaa );
        // Add the light to the scene
        scene.add( lightsource );
    //=====>>>> 5. Statistics
        stats = new Stats();
        container.appendChild( stats.dom );
    //=====>>>> 6. Controls
        controls = new THREE.OrbitControls( camera, renderer.domElement );
        controls.addEventListener( 'change', render );
        // resize
        window.addEventListener( 'resize', onWindowResize, false );
      }
      function render() {
        renderer.render( scene, camera );
        stats.update();
      }
    </script>
  </body>
</html>

```

Figura 5. Un esempio di scheletro di file in three.js per il display di un modello 3D. Nei riquadri le chiamate alla libreria

Nel file del visualizzatore prima si definisce una scena (vuota), poi si aggiunge una camera per poter osservare la scena. In seguito si carica il file OBJ, si aggiungono le luci e il pannello delle statistiche che informano l'utente sulla velocità del rendering e sui consumi di memoria. Si aggiunge poi una sezione dedicata ai controlli, che permettono di muovere la camera intorno al modello tramite il mouse e come ultima sezione di codice si definisce il *loop* di rendering, ovvero che tipo di animazione dovrà essere compiuta nel browser. Nel presente caso l'animazione è quella relativa all'eventuale riposizionamento della telecamera, ma sono anche possibili altri tipo di animazioni, ad esempio far ruotare l'oggetto osservato.

In Figura 6 e Figura 7 sono mostrati due esempi di modello tridimensionale mostrati con il visualizzatore realizzato sotto three.js all'interno di Firefox. I modelli sono quelli ottenuti rispettivamente a seguito di un volo su di un'aviosuperficie vicino Terni e nelle prossimità dell'edificio F65 in Casaccia, entrambi nella banda dell'infrarosso termico. In Figura 6 è possibile vedere un capannone alla cui destra sono alcune persone ed ancora più a destra tre auto parcheggiate evidentemente più calde del prato. Il volo è stato compiuto in inverno ed è interessante notare come il lato frontale del capannone abbia delle dispersioni di calore che seguono le cornici dei pannelli frontali, sulle quali si potrebbe intervenire per ridurre gli sprechi energetici.

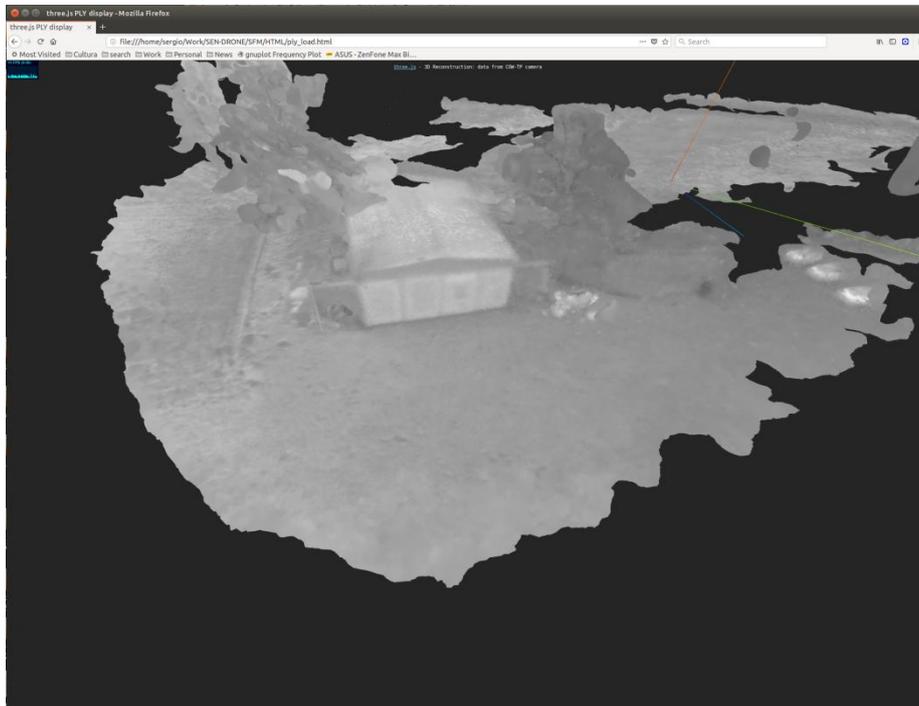


Figura 6. Un esempio di modello 3D visualizzato in una pagina web

Nella Figura 7 è mostrato il modello 3D nell'infrarosso dell'edificio F65; in questo caso il volo è stato compiuto durante l'estate. E' qui visibile come la struttura del tetto e della strada in asfalto intorno all'edificio siano notevolmente più calde (più chiare) della struttura laterale dell'edificio. Le due torri (una al centro davanti ed una dietro) sono più fredde in quanto ospitano gli scambiatori di calore dell'impianto di condizionamento. Sopra al tetto sulla sinistra è chiaramente visibile una striscia più fredda, è il lucernario composto da una serie di lastre di vetro. Questo modello termografico 3D ci suggerisce un intervento al lucernario in modo da limitare il consumo elettrico per il condizionamento dell'edificio.

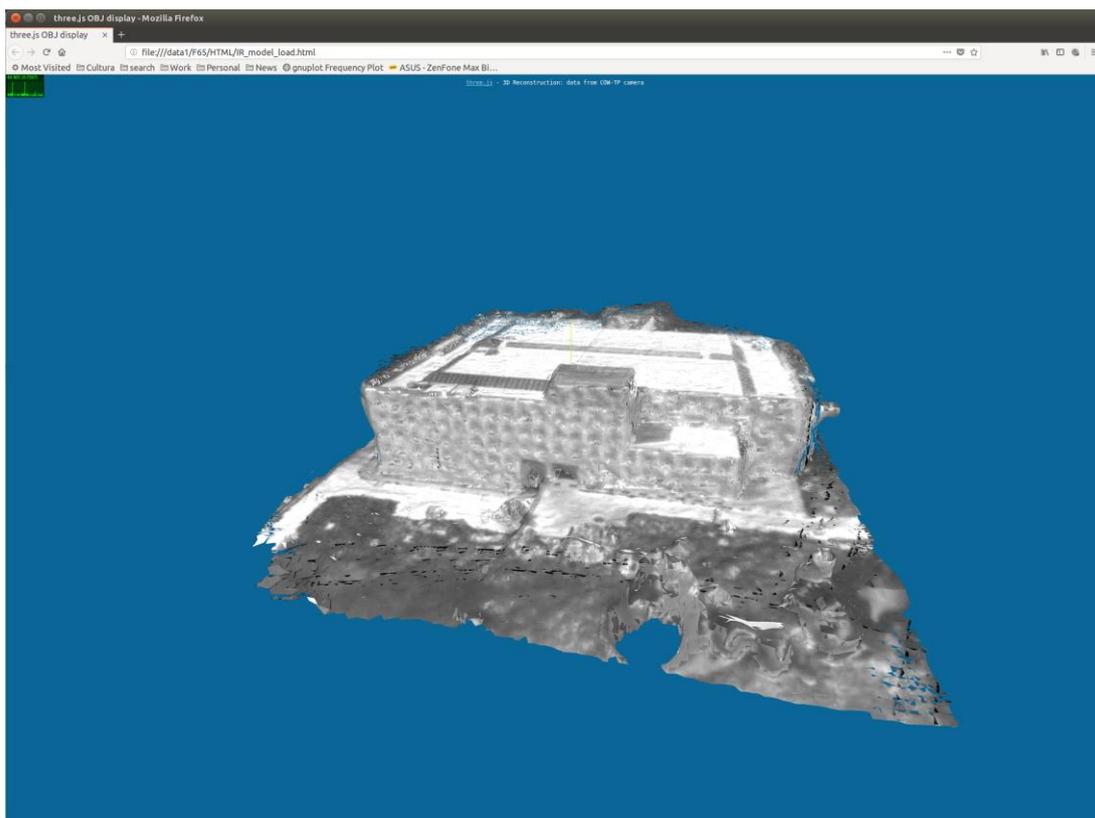


Figura 7. Il modello 3D in infrarosso dell'edificio F65 visualizzato nella pagina web

Si consiglia di utilizzare il visualizzatore sotto Firefox, in quanto sotto Chrome presenta alcuni problemi che dipendono dai parametri di sicurezza interni al browser di Google.

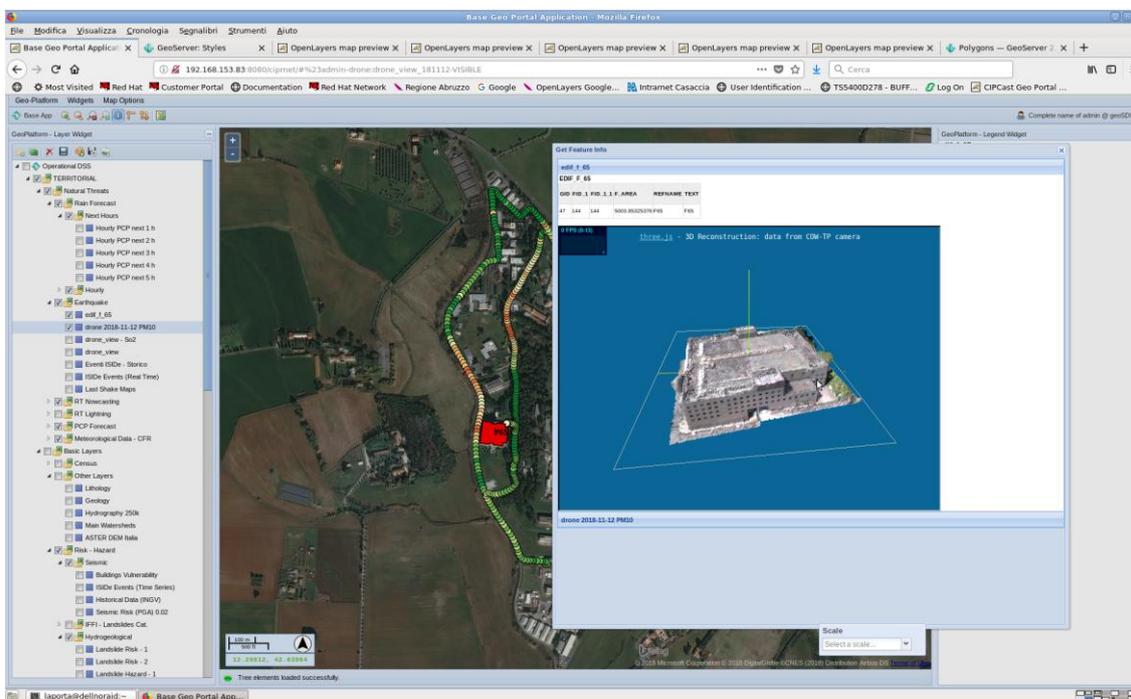


Figura 8. Il modello 3D richiamato nel DSS

Una volta realizzata, la pagina web di visualizzazione viene a risiedere su di un server che è accessibile al DSS. In Figura 8 è mostrato il risultato dell'interrogazione del DSS: l'edificio F65 è evidenziato (in rosso) ed è mostrato il suo modello 3D tramite il visualizzatore in HTML.

2.3 Algoritmi per la separazione di velivoli

Un aspetto rilevante dell'utilizzo di velivoli pilotati remotamente è senz'altro quello della sicurezza. I RPAS (Remotely Piloted Aerial System) possono rappresentare un fattore di rischio sia nella gestione del traffico aereo che nella sicurezza di persone e cose a terra.

Il Laboratorio di Robotica dell'ENEA è stato partner di una serie di progetti europei e regionali (MACRO, EUROSTARS ARCA, SARA, OLGA) volti all'ideazione, sperimentazione e perfezionamento di un algoritmo che permetta ad un velivolo unmanned autonomo un volo sicuro per sé e per il traffico limitrofo [20]. Questa serie di progetti è stata sviluppata in collaborazione con la PMI italiana Deep Blue s.r.l.. Tema comune di questi progetti è la realizzazione di un sistema sia hardware che software che possa essere alloggiato a bordo di un UAV (Unmanned Aerial Vehicle) e che sia in grado di permettere al pilota del velivolo, sia esso automatico o umano, di evitare eventi di mancata separazione tra aeromobili. Ovvero di segnalare al pilota la manovra evasiva che permetta al veicolo di non avvicinarsi mai ad altro aeromobile entro un dato raggio di sicurezza (per i velivoli di linea cinque miglia nautiche, circa 8 km) prescritto dalle regole del controllo del traffico aereo. Nel caso di un velivolo completamente autonomo, la manovra evasiva viene direttamente suggerita all'autopilota. In Figura 9 è mostrato il risultato della guida da parte dell'algoritmo di quattro velivoli autonomi posti inizialmente ai quattro punti cardinali con rotta convergente al punto centrale. E' evidente come l'algoritmo sia in grado di far manovrare gli aeromobili in modo da evitare i conflitti (il cerchio verde rappresenta l'area di rispetto di 5 miglia nautiche) cercando al contempo di minimizzare l'aumento del percorso.

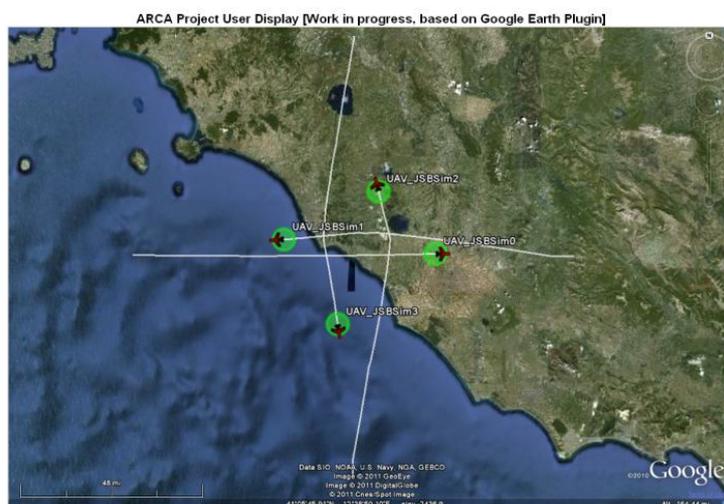


Figura 9. Simulazione di quattro UAV in rotte di collisione controllate e risolte dall'algoritmo

Il progetto OLGA (Optimal routing for General Aviation), condotto nell'ambito di un accordo bilaterale Italia Israele, ha portato alla realizzazione di uno strumento, basato su *tablet*, che suggerisse al pilota di aviazione civile la rotta più sicura, tenendo in considerazione sia il traffico limitrofo che le condizioni meteo. L'idea è quella di avere l'algoritmo precedentemente descritto utilizzato anche nel campo dell'aviazione civile (diporto), che naviga tipicamente a vista, per suggerire al pilota manovre che permettano il volo in sicurezza in presenza di traffico e/o maltempo.

Il cuore del sistema è un algoritmo basato sulla Teoria dei Giochi [21]. Essa, in estrema sintesi, analizza matematicamente situazioni di conflitto tra agenti e ne ricerca soluzioni competitive e/o cooperative

tramite modelli. Studia cioè le decisioni individuali in situazioni dove ci siano interazioni tra più soggetti interdipendenti, ovvero dove le decisioni di uno possano influire sui risultati conseguibili da un altro, finalizzate al massimo guadagno del soggetto considerato. Nata per analizzare le strategie per vincere nei giochi, si è poi estesa ai campi più disparati, in modo particolare in economia.

L’algoritmo in realtà sfrutta un’estensione di questa teoria denominata *Satisficing Game Theory* (SGT) [22]. Il neologismo *satisficing* è l’unione di *satisfy* (soddisfare) e *suffice* (essere sufficiente). Tale estensione prende in considerazione, mediante l’uso di opportuni strumenti matematici, i pro e i contro di una decisione di un singolo agente (nel caso in esame il drone), tenendo però anche in conto le possibili decisioni che possono essere prese dagli altri agenti che operano nel contesto considerato, secondo una scala condivisa di priorità. La decisione finale di ogni agente sarà dunque il frutto di un bilanciamento tra i propri desiderata e le esigenze degli altri agenti con cui si confronta, dando così luogo all’instaurazione di un comportamento collettivo di tipo anche altruistico. Questa strategia quindi fornisce una soluzione adeguata alle esigenze di tutti gli agenti in gioco, ma, conseguentemente, non necessariamente quella ottimale per tutti gli agenti.

Lo strumento del progetto OLGA è rappresentato da un tablet sul cui display è visualizzata l’area geografica dove avviene il volo con tutti gli altri velivoli e con anche le eventuali zone di maltempo, l’algoritmo considera se stesso come l’agente a più bassa priorità rispetto agli altri aeromobili e al tempo meteorologico e quindi calcola la miglior traiettoria per poter passare a distanza di sicurezza sia dagli aerei che dalle zone di maltempo. Questa traiettoria è poi presentata al pilota, al quale è lasciata la decisione finale di seguire il suggerimento o meno, Figura 10.



Figura 10. Il sistema OLGA: in alto la simulazione del volo di un Cessna, in basso lo strumento OLGA dove è visibile in alto un aereo (in verde) in rotta di collisione ed il suggerimento di rotta in rosso

Nel caso qui in esame l’algoritmo è stato ripensato per l’applicazione di monitoraggio degli edifici: l’agente che utilizza l’algoritmo (il drone) continua a considerare tutti gli altri eventuali droni presenti come a maggiore priorità e, soprattutto, considera gli edifici come ‘zone di maltempo’ entro cui non si deve volare.

Le attività condotte dal Laboratorio di Robotica sono state rivolte agli aspetti teorici della customizzazione dell’algoritmo alle mutate condizioni di utilizzo. Ovvero all’inserimento di edifici nella rappresentazione

dell'ambiente, al calcolo delle rotte utilizzando una strategia di *fast forward* in cui ad ogni passo temporale si calcolano le posizioni di tutti gli attori presenti in un tempo accelerato per prevederne le posizioni successive e poter così costruire una traiettoria sicura con un buon grado di sicurezza ed affidabilità.

Le attività implementative sono state invece affrontate dal Dipartimento di Ingegneria Elettronica dell'Università Tor Vergata di Roma.

2.3.1 L'algoritmo di separazione – teoria

L'algoritmo di separazione è basato sulla SGT che fondamentalemente calcola la migliore tra le varie scelte utilizzando due funzioni: la *rejectability* e la *selectability*, che rappresentano rispettivamente i costi ed i benefici per un agente quando compie una scelta. Le scelte possibili sono fondamentalemente di due tipi: la virata (o la prosecuzione diritto) e il cambio di livello di volo (salita o discesa)

Nel contesto del volo, il drone viene considerato un agente e la SGT viene utilizzata come segue. La *selectability*, ovvero i benefici relativi ad una data manovra, sono essenzialmente proporzionali alla ottimalità della rotta risultante data la destinazione finale. La *rejectability*, ovvero i costi, sono proporzionali ai rischi di collisione con altri velivoli, se presenti, o con il maltempo. L'agente calcola costi e benefici di ogni manovra e quindi sceglie la direzione che massimizza la differenza tra *selectability* e *rejectability*, suggerendola al pilota. In parole povere l'agente suggerisce la rotta migliore con riguardo al minimo rischio.

Per poter operare l'agente deve essere a conoscenza della configurazione del maltempo e delle caratteristiche degli altri eventuali droni presenti nell'area. Questi dati sono tipicamente posizione, velocità e direzione di tutti gli agenti presenti, il maltempo è descritto come un velivolo ma con velocità necessariamente nulla.

Nella versione 2D dell'algoritmo, dove si ragiona per livelli di volo, l'agente può scegliere cinque diverse direzioni: una in avanti e due per parte a destra e sinistra. Nella versione originale dell'algoritmo esiste una classifica di priorità per gli agenti presenti: chi ha maggiore priorità ha il diritto di passare per primo (ad esempio perché in volo da più tempo), il maltempo ha la massima priorità in quanto tutti i velivoli devono evitarlo. Nel caso presente l'agente che suggerisce al pilota si mette nella condizione di essere l'agente a più bassa priorità, ovvero sarà l'unico a manovrare per evitare tutti gli altri, velivoli o maltempo che siano. Questo per mettersi nelle condizioni di massima sicurezza. In aggiunta nella versione dell'algoritmo qui sviluppato il posto del maltempo viene preso dagli edifici dai quali è ovviamente necessario stare lontani.

La *rejectability* è incondizionata: l'agente la calcolerà con 'egoismo' comparando la estensione lineare di ciascuna scelta di virata con le proiezioni delle traiettorie di tutti gli agenti con priorità maggiore. Quindi con tutti gli altri elementi presenti nello scenario. Ogni eventuale conflitto aggiunge un peso alla funzione di *rejectability* relativa a quella determinata manovra sulla base della distanza nel tempo della minaccia e della sua severità. Queste regole accrescono la *rejectability* per quelle manovre che portano a conflitti (separazioni troppo piccole) con un peso maggiore per incidenti più vicini nel tempo.

La *selectability* rispecchia il raggiungimento della meta. Essa è influenzata anche dalle preferenze degli altri agenti. La *selectability* di un agente è la combinazione lineare tra quella propria e quella di tutti quegli agenti con priorità più alta.

Una volta calcolate queste funzioni la direzione verrà scelta in modo da massimizzare la differenza tra *selectability* e *rejectability*. Quindi ogni agente cerca di massimizzare i benefici e minimizzare i costi, ma tenendo conto dei desiderata degli altri agenti.

L'algoritmo tiene anche conto della tridimensionalità, permettendo il cambio del livello di volo dell'agente, andando a calcolare *rejectability* e *selectability* anche nei livelli di volo adiacenti (superiore ed inferiore) per verificare se una eventuale soluzione di cambio di livello non sia migliore.

L'algoritmo nella presente applicazione è utilizzato per poter suggerire al pilota del drone una traiettoria che sia il più possibile sicura, ovvero che si mantenga distante dagli edifici presenti e che eviti di andare in situazione di conflitto con eventuali altri droni o velivoli che operino nella stessa area.

Il calcolo della traiettoria viene eseguito applicando l'algoritmo nel seguente modo.

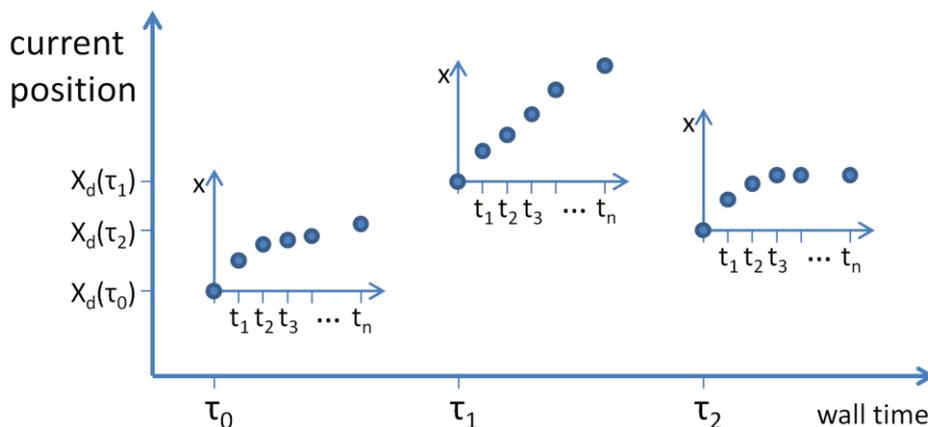


Figura 11. I due diversi riferimenti temporali: tempo reale e tempo di fast forward

Con riferimento alla Figura 11, si considerino due tempi: il tempo reale della simulazione ed un tempo di *fast forward* per il calcolo della traiettoria. Sia τ_0 il tempo corrente, si congeli l'evoluzione temporale della simulazione a questo punto. Il drone per il quale si calcola la traiettoria avrà una data posizione, velocità e direzione. Sulla base dei dati degli edifici e degli altri droni, l'algoritmo sarà in grado di calcolare puntualmente la virata che comporta i maggior benefici (avvicinamento al prossimo waypoint, ovvero al punto di arrivo) e i minori costi (allontanamento da situazioni di conflitto con altri droni o edifici, ovvero passaggi troppo ravvicinati).

Con questa nuova direzione sarà possibile calcolare la posizione del drone considerato ad un tempo $t_1 > \tau_0$ nel futuro (simulazione in *fast forward*). Allo stesso modo, mantenendo inalterati i dati degli altri attori (droni ed edifici) è possibile calcolare dove essi saranno al medesimo tempo t_1 . A questo punto si avrà una nuova configurazione di tutti gli attori al tempo t_1 , dove ad aver manovrato è stato solo il drone in considerazione, mentre gli altri attori hanno agito mantenendo i propri parametri di volo. Con questi dati sarà ora possibile calcolare una nuova virata per il drone considerato che renda la rotta più conveniente. Sarà poi possibile fare di nuovo avanzare il tempo di *fast forward* alla successiva configurazione, ripetendo il ragionamento per il drone portato in questa posizione e così via, fino a calcolare una traiettoria composta da un dato numero di passi in *fast forward*.

Le attività del PAR 2017 relative all'ausilio al pilota oltre a quelle relative agli aspetti teorici e customizzazione dell'algoritmo qui esposte, sono state articolate nei seguenti punti:

- modellazione degli edifici all'interno dell'algoritmo;
- adattamento dell'algoritmo alla presenza degli edifici;
- implementazione dell'algoritmo in ambiente ROS (Robot Operating System);
- integrazione dell'algoritmo nel contesto del simulatore di volo per l'addestramento dei piloti realizzato nel PAR 2016;
- test operativi del sistema nel simulatore.

Queste attività sono state condotte dal Dipartimento di Ingegneria Elettronica dell'Università Tor Vergata di Roma e sono descritte nel Rapporto Tecnico "Studio e sviluppo di algoritmi per l'ausilio al pilota e l'elaborazione dei dati sensoriali di bordo" RdS/2017/065 [10], al quale si rimanda. I test operativi sono invece stati condotti presso il Laboratorio di Robotica dell'ENEA e sono descritti nel paragrafo '2.5 Campagne sperimentali e test di sistema'.

2.4 Studio della portabilità di ulteriori sensori chimici a bordo del drone

Lo studio di portabilità di ulteriori sensori chimici a bordo del drone nel precedente anno di attività (PAR2016) è passato attraverso un'analisi preliminare sulle possibili approcci e soluzioni tecniche che andassero a soddisfare i vari vincoli imposti [1]. In particolare nel corso della valutazione è stata individuata e valutata quale buon candidato l'infrastruttura Monica, un'infrastruttura sperimentale per la rilevazione pervasiva della qualità dell'aria.

Per alcune delle soluzioni tecniche sono stati individuati più approcci principali necessari all'adattamento dell'infrastruttura e del dispositivo sensoriale ai nuovi scenari.

La valutazione e l'analisi delle possibili soluzioni è stata eseguita per le varie sezioni individuandole come necessarie (o al limite opzionali) al sistema aviotrasportato. In particolare sono state valutate possibili soluzioni per le seguenti sezioni:

- sistema di comunicazione dati (opzionale);
- sistema di storage interno (datalogging);
- modifiche Hardware (HW) eventualmente necessarie per i nuovi scenari operativi;
- modifiche Firmware (FW) agli algoritmi di gestione interna;
- eventuali estensioni della dotazione sensoristica;
- upgrade software al server/backend per adattarli ai nuovi scenari operativi.

Tra le soluzioni individuate come compatibili si è voluto individuarne una da seguire per realizzare un primo *test case* funzionante anche se a funzionalità limitate ma che potesse comunque dimostrarne la fattibilità tecnica operativa.

In particolare per quanto riguarda le modifiche HW all'infrastruttura i possibili approcci prevedevano due alternative:

- integrare tutte le funzionalità del dispositivo base Monica e del terminale *android* in un'unica *single board* o *multi shield board* avente una dotazione minimale HW sufficiente ad assolvere tutte le funzioni previste;
- limitare al minimo le modifiche al dispositivo Monica e sostituire il terminale *android* con una *board* HW che ne svolgesse solo le funzionalità minime richieste.

Entrambi gli approcci dal punto di vista hardware costituiscono una possibile valida soluzione al problema tecnico. Diverso risulta l'impatto sulle modifiche al firmware. Dal punto di vista dell'adattamento il primo approccio comporta la riscrittura dell'intero FW mentre nel secondo approccio le modifiche sono limitate all'hardware aggiuntivo.

Essendo un test funzionale si è scelto di seguire una soluzione realizzativa [2] con minor impatto sull'infrastruttura e sul dispositivo attuale come previsto dalla seconda delle alternative elencate [23].

2.4.1 Adattamento HW/FW all'infrastruttura finalizzato al test funzionale.

La soluzione scelta per il *payload* del test funzionale è composta dalle seguenti piattaforme di sviluppo IoT:

- Raspberry Pi 3 Model B [24];
- Pycom Wipy [25] + PyTrack [26];

dove il Raspberry Pi 3 model B, fornisce il motore di calcolo principale e su di esso vengono implementate anche le citate funzionalità di storage interno (datalogging) e connettività. Il Wipy con la sua estensione Pytrack fornisce le funzionalità di geolocalizzazione.

Nel Raspberry 3 dal lato firmware sono stati creati *script* in Python per l'implementazione degli algoritmi necessari al *porting* delle funzioni strettamente necessarie dall'App preesistente di seguito elencate:

- gestione della comunicazione con il dispositivo Monica e *parsing* dei dati;
- gestione della comunicazione con il modulo Wipy + PyTrack e geolocalizzazione del dato;
- gestione del sistema di storage interno (datalogging);
- gestione della comunicazione con il server di *backend* (Mena);

Nel modulo Wipy sono stati creati *script* in MicroPython (mPython) per la gestione delle richieste dall'unità di controllo principale (Raspberry) e per la gestione del sottomodulo GNSS (Global Navigation Satellite System) (per il PyTrack).

L'impiego dei linguaggi basati su Python è stata dettata da un'esigenza di portabilità del codice anche nell'ipotesi di un eventuale ottimizzazione dell'hardware nella direzione di inglobare entrambe le funzionalità del Raspberry e del Wipy in quest'ultima categoria di dispositivi molto compatta che si basa su tale linguaggio.

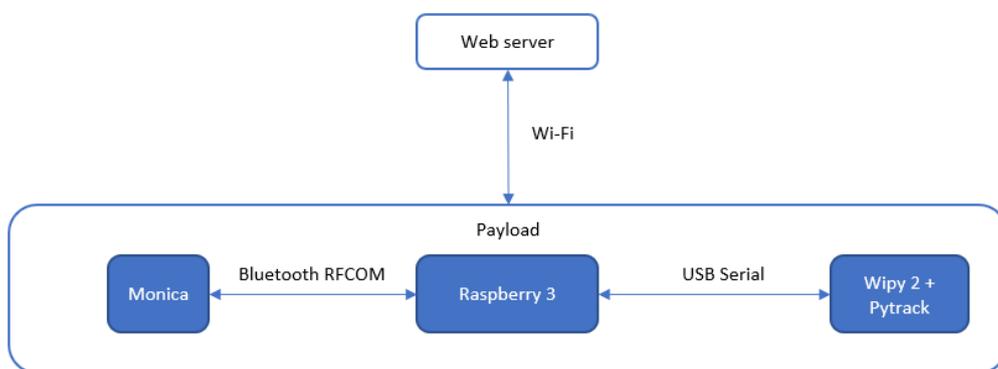


Figura 12. Schema architetturale del payload

L'architettura Hardware implementata è descritta in Figura 12.

Le due piattaforme costituiscono l'hardware aggiuntivo che sostituirà il sottoinsieme delle necessarie funzionalità precedentemente svolta dall'App sullo *smartphone*. Esse insieme al dispositivo Monica costituiscono un sistema completamente indipendente dal vettore su cui è installato (payload).

Come mostrato in Figura 12, il Raspberry 3 effettua una connessione Bluetooth (protocollo RFCOM) verso il dispositivo Monica, ne acquisisce i dati sensoriali e, attraverso un canale seriale cablato (USB), interroga il Pycom Wipy + PyTrack per ottenere i dati di georeferenziazione. I dati così composti vengono poi salvati dal Raspberry in locale e possono essere caricati sul server *backend* di Monica (MENA) sia durante la registrazione della sessione, se presente connettività, che in differita. Con la memoria di massa estraibile (SD) presente sul Raspberry sono implementate le previste funzionalità per il sistema di storage:

- il logging di dati grezzi sensoriali e di controllo (per analisi approfondite post missione);
- il backup dei dati sintetici per ridondanza e *anomaly correction* (e quindi anche ritrasmissione in caso di fault del sistema di comunicazione);

Per il sistema di comunicazione abbiamo indicato già possibili soluzioni tecniche anche *long range* compatibili con le specifiche tecniche.

Per lo specifico test funzionale si è scelto il protocollo wifi presente di default nelle schede utilizzate estendendone il range con l'utilizzo di dispositivi WIFI direzionali. In particolare è stato utilizzato allo scopo il dispositivo access point Wi-Fi direzionale Ubiquiti Nanostation 2 posto a terra e collegato alla rete internet.

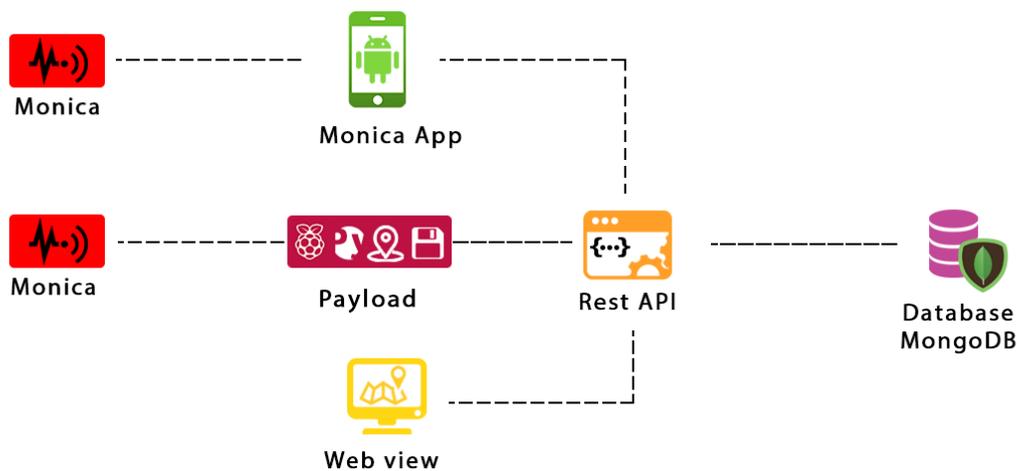


Figura 13. Schema infrastruttura predisposta alla gestione del payload

In Figura 13 è mostrato lo schema dell'infrastruttura predisposta per la gestione del payload. A sinistra c'è il dispositivo Monica, il produttore di dati, che è interfacciato con un App su smartphone oppure è interfacciato al payload composto da Raspberry e Wipy + PyTrack qui descritto. Entrambe le tipologie hanno la capacità di interagire con il server DB di back end per riversare o prelevare i dati delle sessioni prodotte. Completa lo schema l'accesso ai dati via web (a mappe e percorsi delle sessioni) con una doppia interfaccia grafica utente/amministratore.

2.4.2 Test case funzionale

Il payload così progettato e realizzato è stato incapsulato in un *case* ad-hoc di dimensioni di circa 15x15x15 cm e di peso di circa 500g ed è stato impiegato in una prova di volo ed acquisizione con l'array sensoriale classico del dispositivo Monica. Si veda più avanti il paragrafo '2.5 Campagne sperimentali e test di sistema'.

2.4.3 Adattamento sistema di calibrazioni sensori

L'attuale *array* sensoriale del dispositivo Monica è costituito da sensori elettrochimici (per gli inquinanti) ed "a circuito integrato" per l'umidità e la temperatura. I primi necessitano di calibrazione/ricalibrazione periodica per mantenere la corretta accuratezza nelle rilevazioni, deteriorata da normali e documentati fenomeni di invecchiamento del sensore. Tale fase viene realizzata attualmente senza certificazione in un sistema costituito da una camera di condizionamento sensori e da un sistema di condizionamento e acquisizione di dati sensoriali senza utilizzare la piattaforma Monica e il suo *backend*.

Esso è attualmente un sistema da laboratorio utilizzato per verificare calibrazioni di fabbrica ed eventualmente estrarre ulteriori curve di calibrazione (multivariata, *zero drift curve*, ecc.).

Nello specifico utilizzo, esiste la necessità di valutare con precisione l'impatto che l'impiego in veloce mobilità dei sensori o il loro incapsulamento in griglie di protezione, può avere sulla sensibilità all'analita. Inoltre è fondamentale l'utilizzo della piattaforma Monica/Mena per testare in laboratorio nuove tipologie di dispositivi HW derivanti da Monica da aggiungere alla piattaforma.

A tale scopo è stato progettato un *upgrade* alla dotazione standard della camera di condizionamento sensori per poter realizzare tali prove.

In particolare il sistema prevede l'utilizzo di una *board* android su cui viene fatta girare l'app standard Monica che interroga tramite un adattatore BT (o eventualmente possa fare da bridge Wifi) il dispositivo Monica, il payload, ecc., riversando i dati raw nel backend/DB MENA.

Per quanto riguarda l'impianto, esso permette di testare sensori di gas in un ambiente con atmosfera controllata. I sensori vengono inseriti in una camera di test in cui viene flussata un'atmosfera artificiale prodotta con flussi di bombole certificate opportunamente miscelati mediante l'uso di valvole e controllori di flusso anch'essi certificati.

Per poter tarare più dispositivi contemporaneamente, si usa una camera di test da 15Lt in cui possono essere posizionati fino a 8 dispositivi. La concentrazione e la stabilizzazione dei gas all'interno della camera segue delle leggi temporali ben descritte e modellate. Per una corretta taratura dei dispositivi è necessario iniettare costantemente e accuratamente la concentrazione massima a cui esporre i dispositivi per un tempo determinato.

In particolare per quanto riguarda l'analita NO₂ (gas molto critico da dosare a basse concentrazioni) i dispositivi Monica vanno tarati da 0 a 500ppb (parti per miliardo) che è un tipico intervallo di concentrazione a cui dovrebbero essere esposti nelle campagne di misura in campo.

Grazie alla catena certificata delle concentrazioni in bombola, dei misuratori e controllori di flusso, le tarature dei dispositivi all'NO₂ raggiungono precisioni al di sotto dei 5ppb. E' però necessario poter controllare e validare la calibrazione con un sistema di misura indipendente (sensore campione) con accuratezza e precisione nota. I misuratori a chemiluminescenza sono una tecnologia affermata e di riferimento per le misurazioni di NO₂ in aria ambiente. La certificazione e i metodi di riferimento delle normative EN 14211 permettono di utilizzare in laboratorio lo strumento T200 [27] della Project Automation S.p.A. per validare con la giusta accuratezza e precisione le calibrazioni effettuate in laboratorio. Si è scelto quindi di dotare il laboratorio di caratterizzazione di uno strumento come quello indicato utile e necessario per poter dare maggiore credibilità alle misurazioni di laboratorio.

2.4.4 Eventuali estensioni della dotazione sensoristica

Per quanto riguarda ipotesi di eventuali estensioni della dotazione sensoristica dell'array, era stata indicata una possibilità esplorativa per la rilevazione delle polveri sottili (PM), sono stati quindi individuati ed acquistati i sensori miniaturizzati possibili candidati all'estensione per le loro caratteristiche generali (Tabella 1).

Tabella 1. Selezione di sensori di polveri miniaturizzati

Produttore	modello	Tipologia	Principio di funzionamento	Dimensioni e peso
Amphenol Advanced Sensors	SM-PWM-01C	Doppio output proporzionale alla concentrazione di particolato sottile (ca. 1-2um) e particolato (ca. 3-10um)	ottico	59x46x18/20 g
Sharp	GP2Y1010AU 0F	Sensore di concentrazione di pm2.5	ottico	53x51x36mm/53g
Sharp	DN7C3CA007	Sensore di concentrazione di pm2.5 con uscita analogica	ottico	53x51x36mm/53g
Plantower	PMS 7003	Laser based PM2.5/10 sensor (0,3-1 um; 1,0-2,5 um; 2,5-10 um)	ottico	48x37x12mm
Wuhan cubic optoelectronics co.	PM2003	Laser Particle Sensor Module PM0.3, PM2.5, PM10 tre canali	ottico	71x55x13.6mm/25g
Alphasense	OPC/N2	Particulate Monitor 0.38 to 17um, con indicazione della distribuzione su 16 bins	ottico	64x75x60/105g
NOVA	SDS018/ SDS011	Laser PM2.5 sensor	ottico	58x20x46mm/100g

Questi sensori sono concepiti per postazioni comunque fisse e in nessun *datasheet* esiste il riferimento all'utilizzo in movimento. Per uno in particolare (Alphasense) viene solo indicato una dipendenza della sensibilità da eventuali variazioni di flusso imposto dal tiraggio forzato.

Particolare attenzione andrebbe quindi riposta nell'utilizzo di tali sensori nei nuovi scenari.

Gli approcci possibili definitivi potrebbero essere due:

- Prevedere un alloggiamento ed un posizionamento del sensore che lo protegga e limiti forti perturbazioni di flusso nelle prossimità della bocca di sensing (magari direzionato anche verso il basso per ridurre, a drone fermo, variazioni forzate del flusso dovute ad eventuali raffiche di vento orizzontali) e progettare test specifici per verificarne la reale efficacia e poter eseguire acquisizioni in lento movimento.
- In caso di fallimento del precedente punto o di non necessaria acquisizione in movimento, prevedere sempre un alloggiamento con le citate caratteristiche e una fase di acquisizione basata su un numero N di test point in cui il drone si ferma per un tempo superiore al setting time del sensore ed effettua l'acquisizione.

Naturalmente anche per tali ipotesi preliminari occorre prevedere dei test case specifici volti a validare in campo la soluzione definitiva.

2.5 Campagne sperimentali e test di sistema

Nel corso della attività del PAR 2017, sono state effettuate alcune campagne sperimentali e dei test dei sistemi realizzati. Le attività possono essere categorizzate nei seguenti punti:

- campagne per raccogliere dati video e termici;
- campagne di raccolta dati chimici;
- campagne di test per il payload di interfacciamento con i sensori aggiuntivi;
- test di funzionalità dell'algoritmo di ausilio al pilota in ambiente simulato.

Le campagne volte alla raccolta di dati video e termici sono state tre: due nel Centro Ricerche Casaccia in aree individuate opportunamente per permettere il volo in sicurezza e nel rispetto delle normative ed una presso un'aviosuperficie.

Durante le campagne di volo sono stati registrati filmati video e termografie. Il video è in risoluzione HD (1980 x 1080 pixel) mentre le immagini termiche sono 640 x 512 a toni di grigio.

Per verificare la funzionalità del sistema chimico è stata realizzata una campagna di misura all'interno del Centro Ricerche Casaccia utilizzando un veicolo terrestre elettrico in modo da evitare la perturbazione delle misure. Si è scelto di utilizzare il veicolo elettrico in quanto il volo su alcune zone della Casaccia è tassativamente vietato essendoci obiettivi sensibili all'interno del centro stesso.

La funzionalità del *payload* progettato e realizzato per interfacciarsi con gli ulteriori sensori chimici è stato testato in una prova di volo ed acquisizione con l'array sensoriale classico del dispositivo Monica.

Infine sono state realizzate delle sessioni di simulazione per verificare la funzionalità del sistema di supporto al pilota utilizzando due o tre droni pilotati contemporaneamente in un ambiente dove fosse presente un edificio.

2.5.1 Campagne dati video e termografie

Nel Centro Casaccia sono state individuate alcune zone come utilizzabili liberamente per il volo di droni. Queste sono poste a sufficiente distanza da edifici frequentati e da alcuni impianti di particolare riguardo. In queste zone sono stati compiuti i voli durante i quali sono stati raccolti i dati utilizzati poi nelle

ricostruzioni 3D qui riportate. Per la raccolta dei dati si ringraziano per la collaborazione gli ingegneri Pollino e Cosimi del Laboratorio ENEA DTE-SEN-APIC.

In Figura 14 è visibile il modello 3D di un'area di esercitazione per i vigili del fuoco, ripreso nel visibile. Sono facilmente individuabili due auto utilizzate per le esercitazioni ed alcuni container, il modello è stato creato sulla base di 36 immagini registrate su un percorso circolare, vedi Figura 15. Dalla medesima area di volo provengono i modelli mostrati precedentemente in Figura 4 ed in Figura 7 relativi all'edificio F65 nelle due bande: visuale e termica, rispettivamente con 126 e 116 immagini acquisite.



Figura 14. Il modello 3D nella prima area di volo in Casaccia

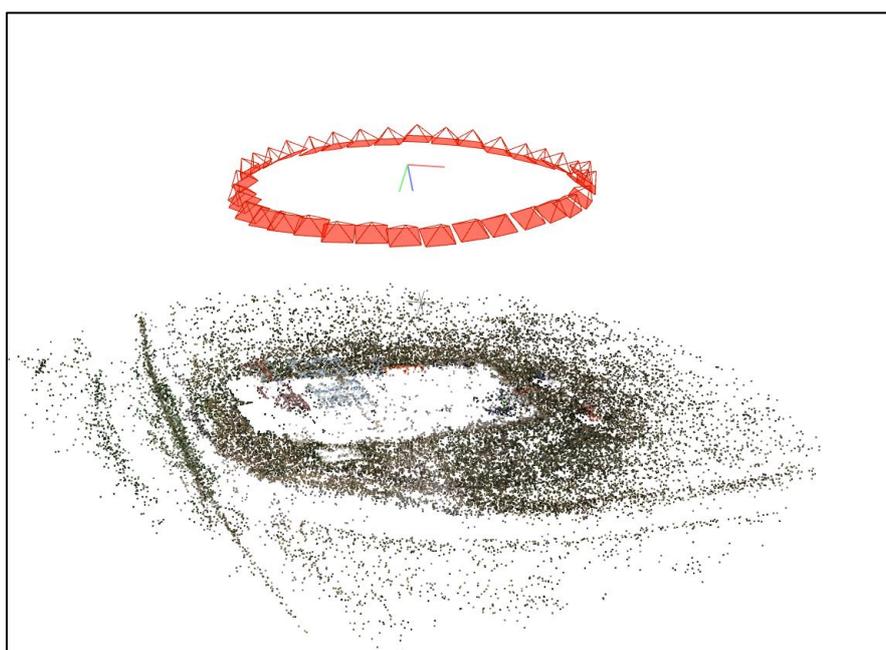


Figura 15. La traiettoria di ripresa delle immagini

In Figura 16 è invece mostrato il modello 3D visuale realizzato a partire da 100 immagini riprese in un'area di volo. In esso è visibile un magazzino ed alcuni container ad esso vicini.

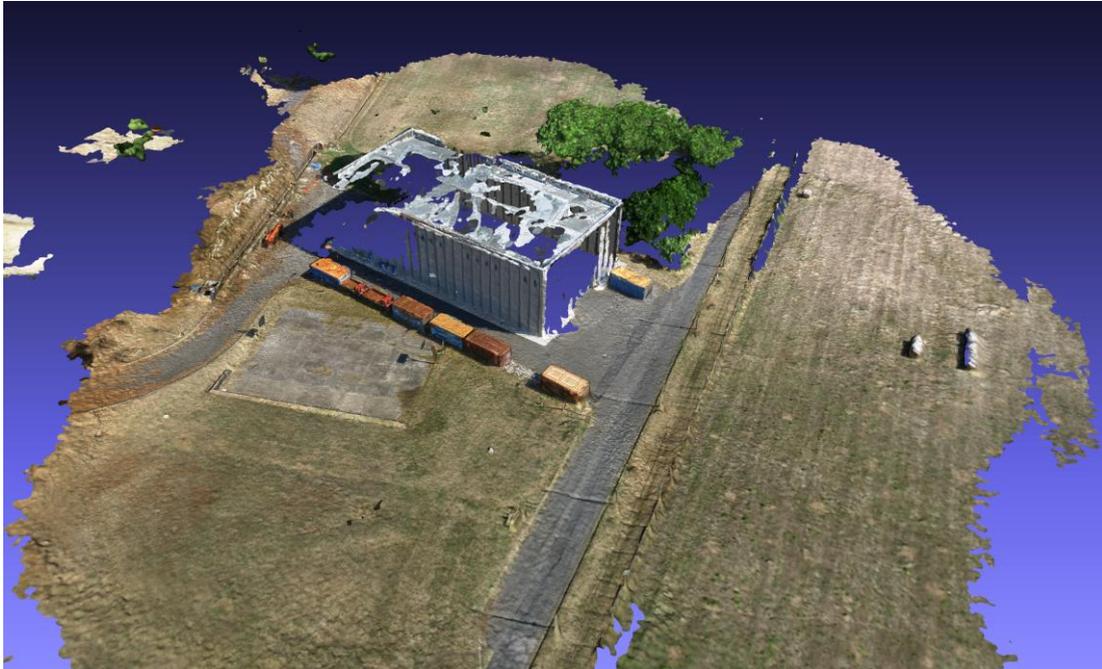


Figura 16. Il modello 3D di un magazzino in Casaccia

In Figura 17 è mostrato il modello 3D realizzato utilizzando i dati raccolti presso l'aviosuperficie "Palazzone" nelle vicinanze di Terni. Sono visibili due capannoni, sulla destra tre auto e tra i due capannoni è visibile con notevole dettaglio il pilota che indossa il giubbotto rifrangente giallo. Per realizzare il modello sono state utilizzate 51 immagini HD estratte dal video ripreso del drone.

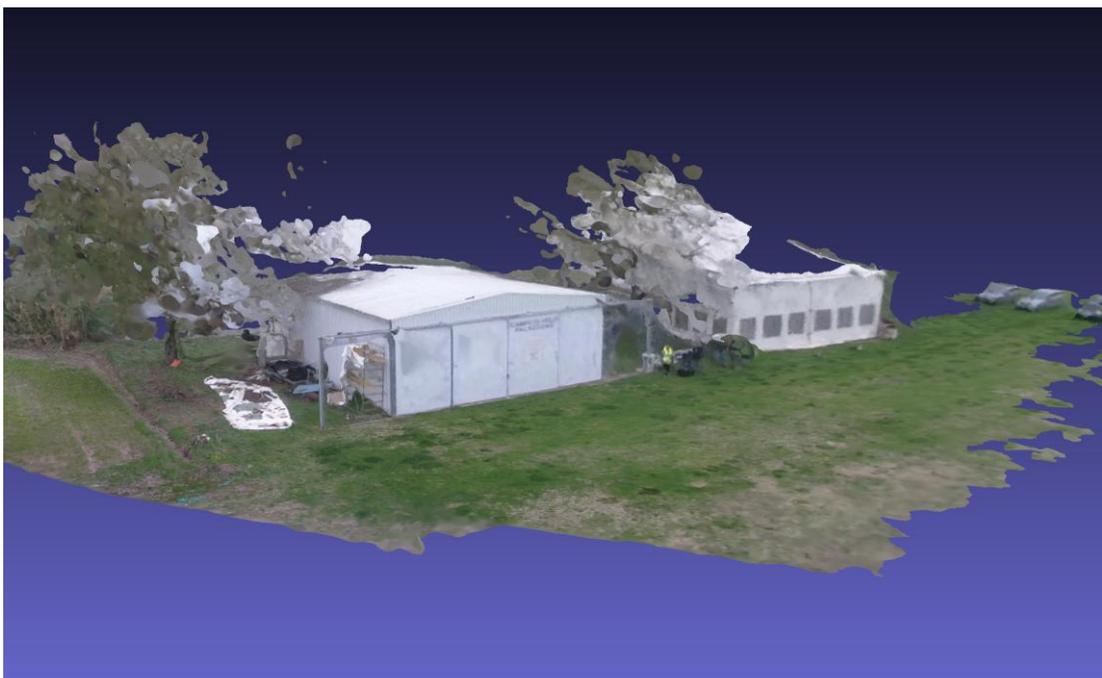


Figura 17. Il modello 3D dell'aviosuperficie "Palazzone"

2.5.2 Campagna sperimentale raccolta dati chimici in Casaccia

La funzionalità del sistema chimico è stata verificata tramite una campagna di misura all'interno del Centro Ricerche Casaccia utilizzando un veicolo terrestre elettrico in modo da evitare la perturbazione delle misure con eventuali gas di scarico prodotti dal veicolo. Si è scelto di utilizzare il veicolo elettrico in quanto il volo su alcune zone della Casaccia è tassativamente vietato essendoci obiettivi sensibili all'interno del centro stesso.

Il veicolo utilizzato è una Renault Twizy in dotazione al Laboratorio di Robotica, il payload chimico è stato alloggiato su di una tavola in plastica assicurata al tetto, si veda la Figura 18 cerchiato in rosso.



Figura 18. Il payload chimico installato sulla Twizy

In Figura 19 è mostrato il percorso compiuto con la Twizy all'interno del C. R. Casaccia. Sono stati percorsi circa 6 km, registrando 673 punti sperimentali, circa uno ogni 9 metri.



Figura 19. Il percorso del payload chimico nel C. R. Casaccia

In Figura 19, a titolo di esempio, è mostrata la misura di particolato relativa al PM 2.5 in ppm (parti per milione), ovvero particelle di 2.5 μm . Al momento della misura nel casale visibile in basso a sinistra nelle immediate vicinanze della legenda, stavano bruciando le potature; ciò è evidente nelle misure molto elevate di particolato nella porzione di percorso più vicina al casale e come la presenza di un segnale elevato in altre zone, mostri che la ventilazione era proveniente da sud-ovest, spingendo il fumo verso nord-est.

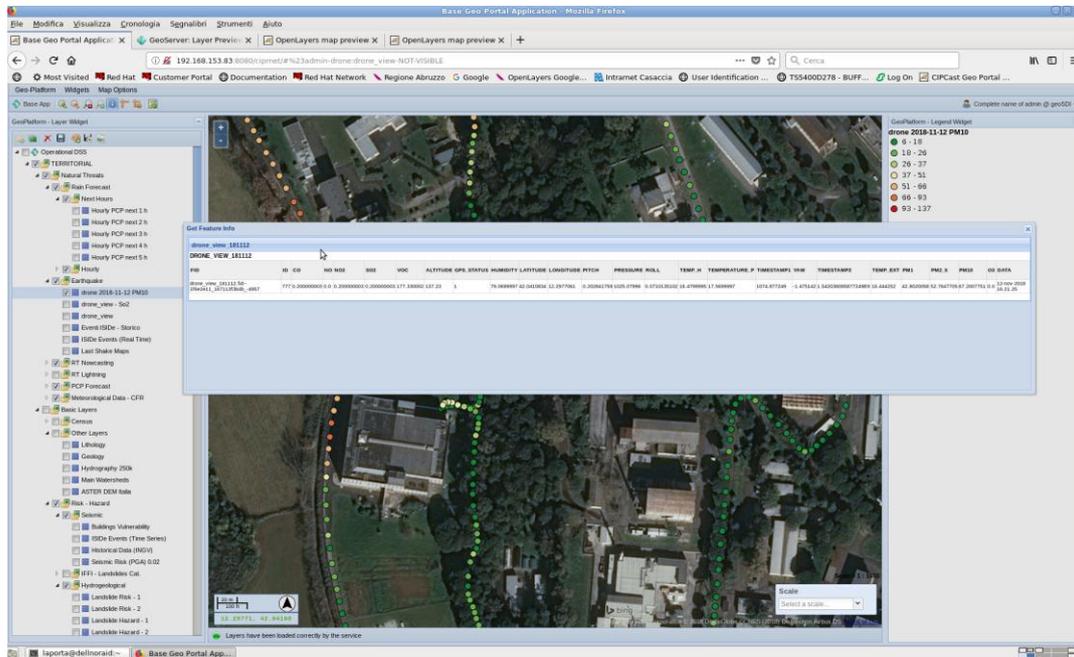


Figura 20. Visualizzazione dei dati chimici nel DSS

I dati sono poi stati trasmessi alla piattaforma DSS, per poterli integrare su di essa. In Figura 20 è mostrato il DSS con nello sfondo il percorso di misurazione e nella finestra in primo piano tutti i dati chimici registrati dal payload, relativi ad uno dei punti di misura.

2.5.3 Campagne di test per il payload di interfacciamento con i sensori aggiuntivi

Il payload progettato e realizzato per l'interfacciamento con gli ulteriori sensori chimici è stato incapsulato in un case ad-hoc di dimensioni di circa 15x15x15 cm e di peso di circa 500g (Figura 21) ed è stato impiegato in una prova di volo ed acquisizione con l'array sensoriale classico del dispositivo Monica. In particolare ci si è focalizzati sul dato dell'analita CO (monossido di carbonio). L'UAV impiegato per il test è un Drone della TopView customizzato, su base DJI S900 con centralina di volo N3 e del peso al decollo di 8kg.

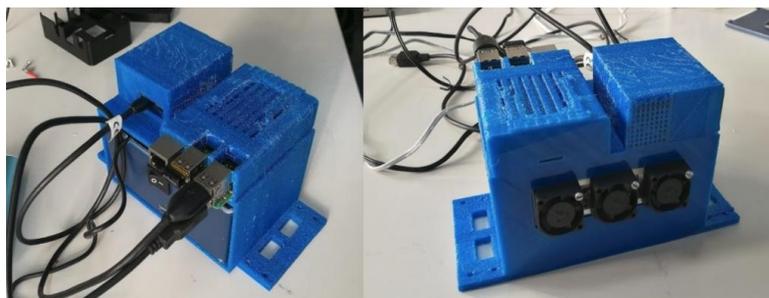


Figura 21. Prototipo del payload

Il test di volo, è stato effettuato realizzando una fonte di monossido di carbonio accendendo un fuoco composto da semplici sterpaglie in modo da simulare una criticità ambientale da monitorare (Figura 22).



Figura 22. Scenario del test di volo

In tutte le sessioni di test sul campo il payload ha lavorato in maniera ottimale, senza mostrare problemi di sorta, sono stati simulate criticità legate a problematiche di connettività (interruzione comunicazione bluetooth o disturbi sulla trasmissione generanti pacchetti dati corrotti) e l'applicazione è stata sempre in grado di lavorare, come da requisiti. Sono state inoltre eseguite sessioni simulando l'assenza di connessione ad Internet e che quindi sono state salvate in locale, nello spazio di archiviazione del Raspberry, ed inviate in un secondo momento sul backend di Monica (MENA).

Come ci si aspettava, i parametri atmosferici hanno influito sulle rilevazioni: in occasione di sporadiche raffiche di vento infatti il sensore ha registrato un aumento delle concentrazioni anche quando il drone volava sottovento ed in lontananza rispetto al fuoco acceso, ed una diminuzione delle concentrazioni con il drone sopravento al fuoco. Nella Figura 23 è mostrato il tracciato GNSS registrato dal payload. Il grafico è stato ottenuto prelevando i dati grezzi ed effettuando un plot tridimensionale.

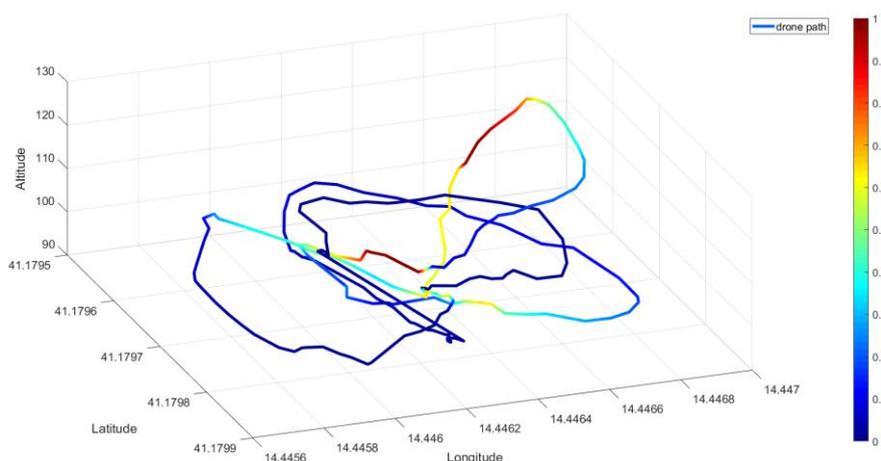


Figura 23. Tracciato della sessione di test del drone, visto in 3D. CO normalizzata tra 0 e 1

In Figura 24 è mostrato il tracciato in 2D come visualizzato direttamente nell'area di backend Monica (MENA)

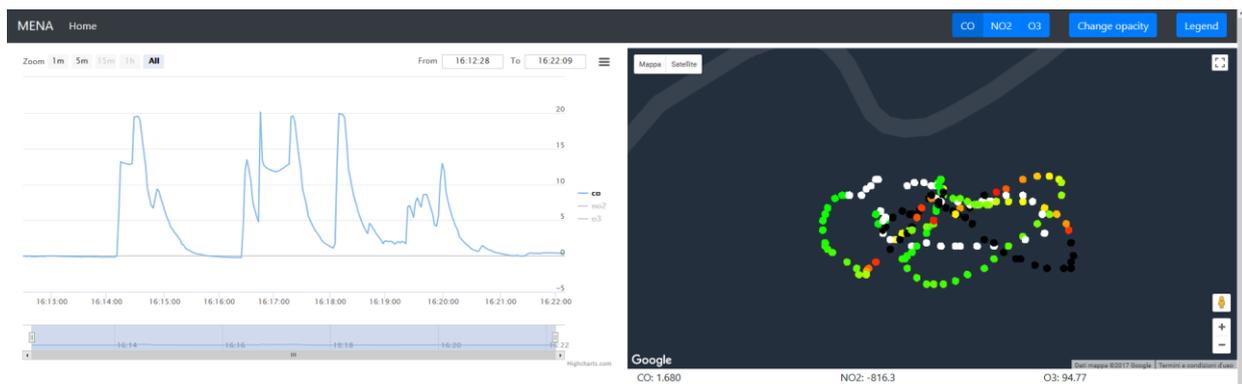


Figura 24. Tracciato CO (valore assoluto e 2D) della sessione visualizzata nel backend di Monica

2.5.4 Collaudo algoritmo sotto simulatore ROS

Allo scopo di verificare la funzionalità del sistema di supporto al pilota realizzato nella corrente annualità, sono state effettuate delle sessioni di volo simulate utilizzando il simulatore sviluppato nella precedente annualità, come modificato in questa annualità.

Lo scopo di queste sessioni è stato quello di cercare di mettere in crisi l'algoritmo che suggerisce al pilota una traiettoria sicura rispetto alla distanza da altri droni, dagli edifici e da eventuali zone di maltempo.

La configurazione di una sessione sperimentale è così schematizzabile: il drone 1 è quello che utilizza l'algoritmo ed è pilotato tramite un joystick dal pilota; a contendere lo spazio aereo del drone 1 possono esserci uno o due altri droni, entrambi pilotati da due piloti per i quali non è attivo l'algoritmo; a completare il teatro delle operazioni, può essere presente un edificio.

In Figura 25 sono mostrate tre configurazioni di conflitto: il drone con le eliche rosse è quello per il quale è attivo l'algoritmo (drone 1).

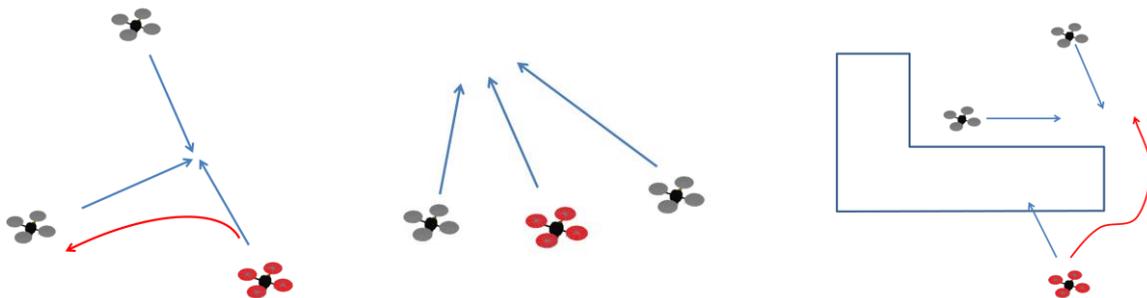


Figura 25. Tre configurazioni di conflitto per i droni

La configurazione sulla sinistra vede due droni che convergono da due direzioni perpendicolari al punto verso il quale sta viaggiando il drone 1. E' anche mostrata in rosso una possibile traiettoria che prevenga il conflitto, ovvero la traiettoria che potrebbe indicare l'algoritmo. La configurazione al centro mostra un conflitto nel quale tutti e tre i droni convergono al medesimo punto, ma provenendo da direzioni molto simili. Quella di destra è complicata dalla presenza di un edificio: il drone 1 potrebbe avere quale suggerimento la traiettoria in rosso che lo porterebbe ad evitare l'edificio, ma la stessa lo porterebbe subito dopo ad essere in conflitto con gli altri due droni.

Le sessioni di prova si sono dunque svolte utilizzando tre piloti che realizzassero condizioni di conflitto simili a quelle mostrate in Figura 25. In Figura 26 è mostrata l'architettura generale delle sessioni di prova con i

tre joypad ed i tre RPAS, l'ADSB topic che fornisce al nodo Sen-drone i dati per richiamare l'algoritmo i cui risultati vengono mostrati infine sul display.

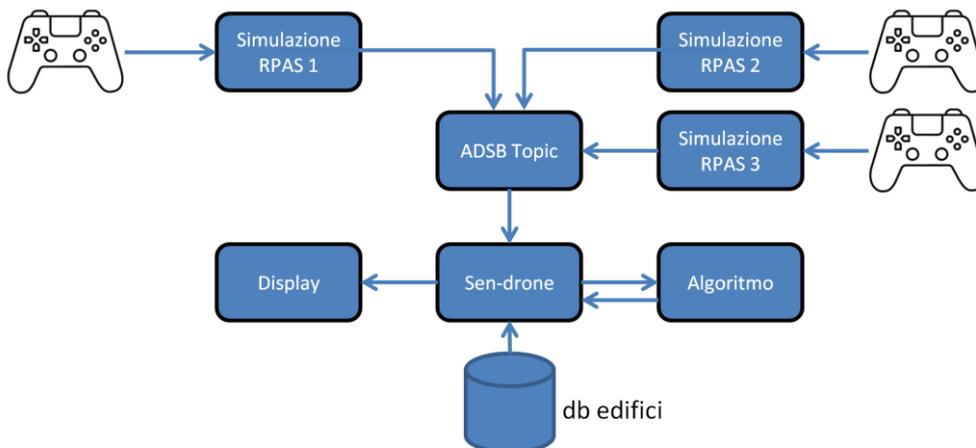


Figura 26. Tipica architettura delle prove nel simulatore

I risultati di queste prove hanno mostrato l'affidabilità dell'algoritmo che ha sempre suggerito una traiettoria che evitasse le situazioni di conflitto con edifici o droni.

A titolo di esempio In Figura 27 è mostrato una configurazione del genere di quella in Figura 25 a sinistra ovvero il drone 1 è opposto a due droni che convergono ad un medesimo punto.



Figura 27. Situazione di due droni convergenti

Nella Figura 27 è visibile la posizione dei due droni opposti e del drone 1 con la traiettoria suggerita dall'algoritmo per evitare un passaggio troppo ravvicinato, costruita per punti secondo l'algoritmo. In questa e nelle successive figure i punti della traiettoria hanno un intervallo di 3 secondi di volo. Naturalmente questo valore insieme al numero massimo di punti possono essere cambiati a piacimento. Il

drone per evitare il drone in alto nella Figura 27 si allarga verso sinistra e questa manovra permette al drone più a sinistra di passare prima evitando anche qui una situazione di conflitto.



Figura 28. Il caso di due droni ed un edificio

In Figura 28 è mostrata invece una situazione di conflitto del genere di quella in Figura 25 a destra, ovvero il drone 1 è opposto ad un edificio e due droni.

Nelle prove sperimentali il pilota del drone 1 ha seguito i suggerimenti di traiettoria dell'algoritmo, naturalmente al meglio delle proprie capacità. Infatti va ricordato che, per seguire i suggerimenti dell'algoritmo, il pilota cercherà di procedere lungo i punti della traiettoria mostrata sul display che, però, cambieranno mano a mano che la configurazione di conflitto evolverà, la miglior traiettoria viene infatti calcolata ogni 3 secondi di volo, sulla base della configurazione corrente.

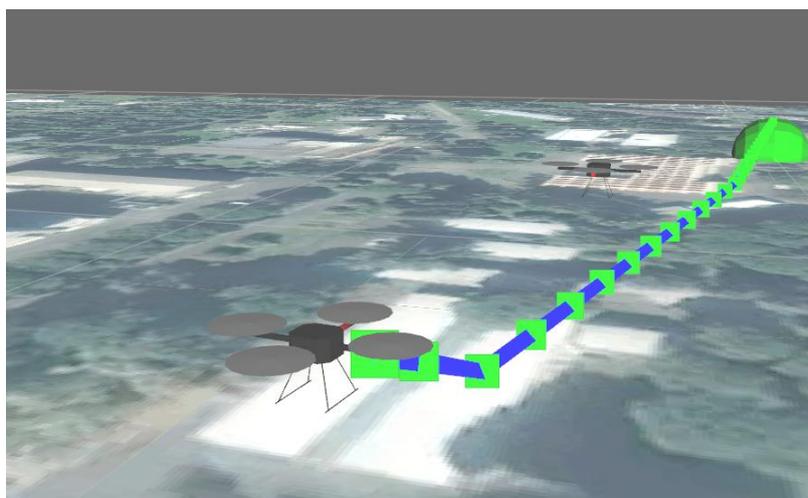


Figura 29. Il caso di cambio di livello di volo

Per completezza in Figura 29 è mostrato il caso di cambio di livello di volo per il drone1. Nella maggior parte dei casi, quando le condizioni di sicurezza lo permettono, il suggerimento da parte dell'algoritmo è una

traiettoria nello stesso livello di volo, ovvero una virata. Se però l'algoritmo non trova una traiettoria sufficientemente sicura nel corrente livello di volo del drone, può suggerire al pilota di cambiare il livello di volo. Nella Figura 29 viene mostrato questo caso, nel quale l'eccessiva vicinanza tra i due droni fa suggerire un abbassamento al drone 1 per evitare la situazione di pericolo.

3 Conclusioni

In questo report sono state presentate le attività del Piano Annuale di Realizzazione 2017 (PAR 2017) relative al sotto obiettivo "Monitoraggio aereo dello smart district" dell'obiettivo "Sicurezza infrastrutture critiche e monitoraggio aereo dello Smart District" del progetto "Sviluppo di un modello integrato di Smart District urbano".

Le attività relative al precedente Piano Annuale di Realizzazione 2016 hanno portato alla realizzazione di un dimostratore per il monitoraggio energetico degli edifici energivori e per quello ambientale per ciò che attiene alla qualità dell'aria; tale dimostratore si basa sulla tecnologia degli aeromobili a pilotaggio remoto multi elica usualmente indicati con il termine drone. Esso è equipaggiato con due diversi *payload* al variare della missione di monitoraggio. Il primo è una *suite* di sensori chimici sensibili ad alcuni inquinanti dell'aria, il secondo è un sistema composto da una termocamera e da una telecamera video, accoppiate. Tramite questi due *payload* è possibile monitorare le prestazioni energetiche degli edifici sia dal punto di vista delle dispersioni termiche, che da quello della produzione di inquinanti legati alle attività energetiche quali il riscaldamento.

Nel Piano Annuale di Realizzazione 2017 le attività si sono sviluppate dunque lungo tre filoni principali:

- l'elaborazione dei dati forniti dai sensori a bordo del drone, in modo da renderli più facilmente fruibili dall'utilizzatore finale, con particolare riferimento al payload termico/visibile;
- l'adattamento e la sperimentazione di un algoritmo per il mantenimento della separazione tra aeromobili sviluppata in ENEA allo scopo di fornire supporto al pilota del drone in ambiente urbano;
- Lo sviluppo di sensori chimici aggiuntivi e la loro integrazione per il payload del drone, sulla base dell'esperienza pregressa del sistema Monica.

Questi tre filoni di attività sono stati affiancati dalle relative prove sperimentali in campo. Il primo ed il secondo argomento sono stati principalmente posti in essere dal Laboratorio di Robotica dell'ENEA (DTE-SEN-IDRA) in collaborazione con il Dipartimento di Ingegneria Elettronica dell'Università di Roma "Tor Vergata"; il terzo filone è stato seguito dal Laboratorio ENEA Dispositivi Innovativi (DTE-FSN-DIN).

Nel primo filone le attività si sono incentrate sugli algoritmi detti di Structure from Motion (SfM), ovvero struttura dal movimento, cioè quella classe di algoritmi di computer vision che permettono di inferire la tridimensionalità degli oggetti mettendo in relazioni diverse immagini riprese da punti di vista diversi degli oggetti stessi. Sono stati confrontati e messi alla prova alcuni software open source per la realizzazione di modelli, realizzando modelli 3D visuali e, soprattutto, aprendo la strada a termografie tridimensionali che permettono un'analisi semplificata di oggetti quali gli edifici.

Un aspetto direttamente connesso al precedente è stato quello di una presentazione efficace dei modelli tridimensionali realizzati. Si è percorsa la via della presentazione tramite una interfaccia di tipo web sviluppata in ambiente *WebGL*, tramite l'approccio *three.js*. Questa scelta permette una più semplice ed efficace presentazione all'interno del Decision Support System (DSS) sviluppato nel sotto obiettivo "DSS protezione e sicurezza delle infrastrutture critiche" condotto principalmente dal Laboratorio ENEA Analisi e Protezione delle Infrastrutture Critiche (DTE-SEN-APIC).

Nel secondo filone è stato studiato ed implementato l'adattamento di un algoritmo, basato sulla Teoria dei Giochi, per la separazione tra aeromobili, al problema di suggerire una traiettoria di volo sicura al pilota del

drone al fine di evitare un eccessivo avvicinamento agli edifici da monitorare o ad eventuale altro traffico aereo nella zona oggetto del monitoraggio. Questo algoritmo sulla base delle posizioni degli attori presenti sulla scena (droni, velivoli, edifici, maltempo, etc.) suggerisce visualmente al pilota la traiettoria più sicura, quella, cioè, che evita l'insorgere di situazioni di conflitto ovvero di permettere un volo a sufficiente distanza da situazioni di pericolo in modo predittivo, prima che il conflitto sia in atto.

Nel terzo filone sono proseguite le attività per equipaggiare il drone con ulteriori sensori chimici provenienti da un sistema portatile di misura recentemente sviluppato in ENEA con finalità simili. In particolare si è realizzata e verificata la portabilità del sistema di sensoristica in un ambiente applicativo quale quello di un payload su drone. Si è anche esplorata la possibilità di equipaggiare il payload con ulteriori sensori per polveri sottili.

Le attività sopra descritte sono state integrate con attività sperimentali per il test operativo delle soluzioni sviluppate. La ricostruzione tridimensionale è stata valutata in più missioni di volo, mentre l'algoritmo per il suggerimento di una traiettoria sicura è stato valutato nell'ambiente di simulazione sviluppato nel corso delle attività del PAR 2016. E' stata condotta una campagna di monitoraggio ambientale all'interno del Centro Ricerche Casaccia, per verificare le funzionalità della soluzione di monitoraggio dell'aria. La portabilità del sistema sensoristico sviluppato nel Laboratorio DTE FSN DIN è stata verificata con un volo su drone in presenza di fumi.

Eventuali sviluppi futuri delle attività possono essere rappresentate dalla realizzazione di ulteriori campagne sperimentali, ad esempio una campagna volta al monitoraggio di una sottostazione elettrica dello Smart District mediante le tecniche sviluppate nel corso delle annualità precedenti, nei tre aspetti di raccolta, di elaborazione dei dati e di interfacciamento verso il DSS. Scopo di questa attività è quello di verificare l'utilità del mezzo termografico per l'analisi delle infrastrutture elettriche di distribuzione. Ulteriori sviluppi dell'algoritmo di ausilio al pilota possono essere diretti all'introduzione di ulteriori elementi di rischio quali il modello del terreno; parallelamente si possono studiare ulteriori configurazioni di conflitto oltre a quelle già sperimentate. Ulteriori attività nel campo dell'introduzione di nuovi sensori chimici possono passare attraverso l'ottimizzazione delle procedure di calibrazione del parco sensori ed ulteriori sessioni sperimentali.

4 Riferimenti bibliografici

- [1] S. Taraglio, L. Blasi, G. Cupertino, C. Moriconi, V. Nanni, S. De Vito, F. Formisano, G. Zanini, F. Russo, M. Villani e L. Vitali, «Sviluppo di un sistema di monitoraggio aereo per lo smart district,» 2017. Rds/PAR2016/021.
- [2] C. Wu, «Towards Linear-time Incremental Structure From Motion,» in *Proceedings of the International Conference on 3D Vision, 3DV '13*, 2013.
- [3] C. Wu, «VisualSFM: A Visual Structure from Motion System,» [Online]. Available: <http://ccwu.me/vsfm/>. [Consultato il giorno 28 09 2018].
- [4] E. Rupnik, M. Daakir e M. Pierrot-Deseilligny, «MicMac – a free open-source solution for photogrammetry,» *Open Geospatial Data, Software and Standards*, vol. 2, n. 14, 2017.
- [5] «MicMac,» [Online]. Available: <http://micmac.engr.eu/index.php/Accueil>. [Consultato il giorno 28 09 2018].
- [6] J. Schönberger e J. Frahm, «Structure-from-Motion Revisited,» in *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [7] J. Schönberger, E. Zheng, M. Pollefeys e J. Frahm, «Pixelwise View Selection for Unstructured Multi-View Stereo,» in *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.
- [8] «Colmap Structure-from-Motion and Multi-View-Stereo,» [Online]. Available: <http://demuc.de/colmap/>. [Consultato il giorno 28 09 2018].
- [9] R. Tsai, «An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision,» in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Miami Beach, FL, 1987.
- [10] S. Betti e S. Chiesa, «Studio e sviluppo di algoritmi per l'ausilio al pilota e l'elaborazione dei dati sensoriali di bordo,» 2018, Rds/PAR2017/065.
- [11] S. Taraglio e S. Chiesa, «Evolutionary Approach to Epipolar Geometry Estimation,» in *Advances in Theory and Applications of Stereo Vision*, A. Bhatti, A cura di, InTech Publisher, 2011, pp. 1 - 16.
- [12] T. Tuytelaars e K. Mikolajczyk, «Local invariant feature detectors: A survey,» *Foundations and Trends in Computer Graphics and Vision*, vol. 3, n. 3, pp. 177-280, 2008.
- [13] D. Lowe, «Object recognition from local scale-invariant features,» in *Proceedings of the International Conference on Computer Vision*, 1999.
- [14] D. Lowe, «Distinctive image features from scale-invariant keypoints,» *International Journal of Computer Vision*, vol. 60, n. 2, pp. 91 - 110, 2004.
- [15] M. Kazhdan, M. Bolitho e H. Hoppe, «Poisson surface reconstruction,» in *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006.
- [16] F. Chiabrando, E. Donadio e F. Rinaudo, «SfM for orthophoto generation: a winning approach for cultural heritage knowledge,» in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 25th International CIPA Symposium 2015*, Taipei, Taiwan, 2015.
- [17] Z. Champion, F. Chiabrando e J. Harrington, «3D modeling of the michigan tech husky statue using a close-range photogrammetric approach,» in *ASPRS Annual Conference and co-located JACIE Workshop*, Tampa, Florida, USA, 2015.
- [18] O. Moutinho, «Evaluation of photogrammetric solutions for rpas: commercial vs open source,» MSc Thesis, University of Porto, Faculdade de Ciencias, Portugal, 2015.
- [19] «three.js - Javascript 3D library,» [Online]. Available: <http://threejs.org>. [Consultato il giorno 28 9 2018].
- [20] D. Taurino, S. Taraglio, A. Tedeschi, A. Pasquini e V. Nanni, «Satisficing Game Theory for enhancing autonomy in unmanned aerial vehicles,» *International Journal of Artificial Intelligence*, vol. 7, n. A11, pp. 316 - 328, 2011.

- [21] «Teoria dei Giochi - Wikipedia,» [Online].
- [22] «Satisficing - Wikipedia,» [Online]. Available: <https://en.wikipedia.org/wiki/Satisficing>. [Consultato il giorno 28 9 2018].
- [23] M. Carrozzo, S. De Vito, E. Esposito, M. Salvato, F. Formisano, E. Massera, G. Di Francia, P. Delli Veneri, M. Iadaresta e A. Mennella, «UAV Intelligent Chemical Multisensor Payload for Networked and Impromptu Gas Monitoring Tasks,» in *5th IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, 2018.
- [24] «Raspberry Pi 3 Model B - Raspberry Pi,» [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. [Consultato il giorno 28 9 2018].
- [25] «WiPy 3.0 - Pycom,» [Online]. Available: <https://pycom.io/product/wipy-3-0/>. [Consultato il giorno 28 9 2018].
- [26] «Pytrack - Pycom,» [Online]. Available: <https://pycom.io/product/pytrack/>. [Consultato il giorno 28 9 2018].
- [27] «Manuale operativo Teledyne Model T200,» [Online]. Available: <http://www.teledyne-api.com/prod/Downloads/06858E%20-%20MANUAL,%20T200.pdf>.

5 Abbreviazioni ed acronimi

I seguenti acronimi sono presenti nel testo e sono qui elencati in ordine di apparizione.

PAR Piano Annuale Realizzativo.

SfM Structure from Motion. Struttura da movimento: algoritmica per la ricostruzione di modelli 3D.

DSS Decision Support System. Sistema di support alle decisioni: un sistema software volto alla presentazione di dati complessi, usualmente in modo georeferenziato.

RPAS Remotely Piloted Aerial Systems. Sistemi aerei a pilotaggio remoto: i droni.

HTML HyperText Markup Language. Linguaggio a marcatori per ipertesti: linguaggio per la realizzazione di contenuti sul web.

GPS Global Positioning System. Sistema di Posizionamento Globale: sistema basato su satelliti per la rilevazione di posizione e altitudine.

CCD Charge-Coupled Device. Dispositivo ad accoppiamento di carica: l'elemento sensibile di un sistema di ripresa di immagini.

SVD Single Value Decomposition. decomposizione ai valori singolari: in algebra lineare, un tipo di fattorizzazione di una matrice basata sull'uso di autovalori e auto vettori.

SIFT Scale-Invariant Feature Transform. Trasformazione di caratteristiche invariante di scala. Algoritmo della computer vision che permette di rilevare e descrivere caratteristiche locali (feature) in immagini.

DoG Difference of Gaussian. Differenza di Gaussiane: algoritmo di miglioramento delle feature di una immagine che coinvolge la sottrazione di una versione sfocata dell'immagine origine da un'altra meno sfocata.

PLY PoLYgon file format. Formato di file poligono: formato di file per la descrizione di modelli 3D.

GIS Geographic Information System. Sistema informativo geografico: insieme di strumenti informatici concepito per descrivere ed analizzare elementi, eventi e fenomeni inerenti alla superficie terrestre.

WebGL Web-based Graphics Library. Una libreria grafica per i browser web.

API Application Programming Interface. Interfaccia per la programmazione di applicazioni: librerie software di un linguaggio di programmazione.

GPU Graphics Processing Unit. Unità grafica di elaborazione: coprocessore specializzato nel rendering di immagini grafiche.

UAV Unmanned Aerial Vehicle. Veicolo aereo senza equipaggio: drone tipicamente autonomo.

SGT Satisficing Game Teory. Teoria dei giochi soddisfacente: strategia di decision-making che non cerca l'ottimalità ma un dato livello di accettabilità.

ROS Robot Operating System. Sistema Operativo per Robot: un *middleware* che fornisce una serie di strumenti informatici per la realizzazione di software per la programmazione e la gestione di robot.

HW SW FW Hardware, Software, Firmware: la parte fisica, il programma di controllo di alto livello e il programma di controllo di basso livello di un dispositivo.

IoT Internet of Things. Internet delle cose: neologismo riferito all'estensione di Internet al mondo degli oggetti e dei luoghi concreti.

GNSS Global Navigation Satellite System. Sistema satellitare di navigazione globale: esempi: GPS, Galileo in UE.

USB Universal Serial Bus. Bus seriale universal: interfaccia di comunicazione seriale.

MENA il server di *backend* del sistema Monica di monitoraggio chimico ambientale.

SD Secure Digital. Formato di schede di memoria flash.

- HD** High Definition. Alta definizione: usualmente video digitali con 1920x1080 pixel.
- PM** Particulate matter. Particolato: sostanze sospese in aria sotto forma di aerosol atmosferico.

6 Indice delle figure

Figura 1. La geometria epipolare.....	9
Figura 2. I sei passi necessari alla costruzione di un modello 3D	11
Figura 3. Una tipica sequenza di comandi per una ricostruzione 3D in MicMac	12
Figura 4. Il modello 3D di un edificio della Casaccia come ricostruito da MicMac (sinistra) e Colmap (destra)	13
Figura 5. Un esempio di scheletro di file in three.js per il display di un modello 3D. Nei riquadri le chiamate alla libreria.....	14
Figura 6. Un esempio di modello 3D visualizzato in una pagina web	15
Figura 7. Il modello 3D in infrarosso dell’edificio F65 visualizzato nella pagina web	16
Figura 8. Il modello 3D richiamato nel DSS	16
Figura 9. Simulazione di quattro UAV in rotte di collisione controllate e risolte dall'algorithm.....	17
Figura 10. Il sistema OLGA: in alto la simulazione del volo di un Cessna, in basso lo strumento OLGA dove è visibile in alto un aereo (in verde) in rotta di collisione ed il suggerimento di rotta in rosso.....	18
Figura 11. I due diversi riferimenti temporali: tempo reale e tempo di fast forward.....	20
Figura 12. Schema architetturale del payload.....	22
Figura 13. Schema infrastruttura predisposta alla gestione del payload.....	23
Figura 14. Il modello 3D nella prima area di volo in Casaccia	26
Figura 15. La traiettoria di ripresa delle immagini	26
Figura 16. Il modello 3D di un magazzino in Casaccia	27
Figura 17. Il modello 3D dell’aviosuperficie “Palazzone”	27
Figura 18. Il payload chimico installato sulla Twizy	28
Figura 19. Il percorso del payload chimico nel C. R. Casaccia	28
Figura 20. Visualizzazione dei dati chimici nel DSS.....	29
Figura 21. Prototipo del payload	29
Figura 22. Scenario del test di volo	30
Figura 23. Tracciato della sessione di test del drone, visto in 3D. CO normalizzata tra 0 e 1.....	30
Figura 24. Tracciato CO (valore assoluto e 2D) della sessione visualizzata nel backend di Monica	31
Figura 25. Tre configurazioni di conflitto per i droni.....	31
Figura 26. Tipica architettura delle prove nel simulatore	32
Figura 27. Situazione di due droni convergenti.....	32
Figura 28. Il caso di due droni ed un edificio.....	33
Figura 29. Il caso di cambio di livello di volo	33