Generació de mapes a partir d'imatges preses amb un UAV.

Guim Perarnau

Resum— En els darrers anys s'ha popularitzat l'ús dels UAVs (*drones*) en una gran varietat d'àmbits. Aquest treball se centra en el camp de generació de mapes (vistes àrees) òptima basat en imatges enregistrades amb un UAV. Per millorar la qualitat del mapa resultant es vol treure profit de la instal·lació de sensors en l'UAV que proporcionin informació que pugui ser útil en el procés, com la posició i l'orientació de l'UAV. Com a tal, es descriuran tots els passos emprats per tal d'aprofitar les dades del sensor, com sincronitzar el sensor amb les imatges enregistrades, l'avaluació i anàlisi de diferents descriptors i detectors de característiques, l'estimació de la transformació geomètrica entre parells d'imatges i finalment la confecció del mapa ajuntant totes les imatges en una de sola. Els resultats obtinguts han sigut dos mètodes per generar el mapa: basat només en imatges i basat només en les dades del sensor, cada un amb els seus avantatges i inconvenients. Com a treball futur queda cercar un mètode vàlid per poder aprofitar els avantatges d'ambdós mètodes, ja que en aquest treball el mètode provat per tal d'assolir aquest objectiu no ha millorat els resultats prèviament obtinguts.

Paraules clau— drone, feature matching, mapping, mosaic, ortomapa, rastreig, UAV, unió d'imatges.

Abstract— Over the last few years UAVs (drones) popularity has increased in a wide range of areas. This project is focused on the field of optimal map generation (aerial views) based on images taken from a UAV. To improve the resulting map quality, sensors have been installed in the UAV to profit from the information they provide that may be useful for the process, like UAV position and orientation. Therefore, all the steps done will be described in order to benefit from the sensor data, as the synchronization between sensors and recorded images, the evaluation and test of different feature detectors and descriptors, the estimation of geometrical transformation between image pairs and finally the making of the map, joining all the images into one. The results are two methods for generating the map: image based and sensor based, each one with its advantages and disadvantages. Future work could be focused on finding a valid method capable of taking avantage of both methods, as in this project the method used to achieve this goal has not improved the results previously obtained.

Index Terms— drone, feature matching, image stitching, mapping, mosaic, orthomap, tracking, UAV.

1 Introducció

AQUEST treball de fi de grau neix de la necessitat de supervisar l'estat dels incendis de forma eficient i econòmica. Quan es produeix un incendi i els bombers acudeixen per extingir-lo, pot passar que hi hagi zones on es tornin a revifar les flames, encara que els bombers l'haguessin extingit amb anterioritat. Ara per ara aquest control s'ha de portar a terme via assistència humana. Una forma alternativa de control d'incendis és l'ús d'UAVs (vehicle aeri no tripulat, conegut popularment com a *drone*) que generin mapes de les zones afectades.

La proposta d'aquest treball consisteix a utilitzar un UAV per a l'ajuda en l'extinció d'incendis. Aquest UAV es dirigirà cap al sinistre per supervisar tot l'incendi i ve equipat amb una càmera visible, una altra infraroja i diferents sensors que aporten informació com GPS, accelerò-metres i brúixola. L'objectiu final consisteix tant en la generació com en l'avaluació d'un mapa de la zona utilitzant la quantitat més gran d'informació disponible (imatges i sensors) per tal que tota la informació visual captada per l'UAV quedi reflectida en un mapa de vista zenital (ortomapa).

El punt de partida del projecte consisteix en un únic UAV que executarà un pla de vol especificat prèviament per obtenir informació tant visual (imatges) com registrant les dades dels sensors de posició. L'UAV enregistrarà tota la informació i en acabar el recorregut es processaran *offline* les dades obtingudes en un ordinador extern per confeccionar l'ortomapa de la zona. Les dades dels sensors, però, no estan sincronitzades amb les imatges enregistrades.

El treball es divideix en tres objectius diferenciats:

1. Sincronitzar les dades dels sensors amb les imatges captades per l'UAV: com que es vol aprofitar les dades dels sensors per generar l'ortomapa és necessari saber la informació dels sensors per a cada frame. La sincronització serà realitzada estimant la trajectòria generada per l'UAV a partir de les imatges per emparellar-la amb la donada pels sensors. Seguidament serà necessari trobar un mètode per posar en correspondència els punts representatius (rotacions del UAV) de les dues trajectòries.
2. Estimació de la transformació geomètrica entre pa-

rells d'imatges: s'explicaran tots els mètodes implementats per estimar el moviment entre imatges. Aquests es poden separar entre basats en imatge (*NCC* [12], descriptors i detectors de característiques [11]), basats en les dades dels sensors i basats en l'algorisme *Lucas-Kanade* [15]. En el cas de l'estimació basada en imatges es realitzaran experiments amb diferents detectors i descriptors de característiques per trobar el que realitza una estimació més precisa i robusta.

3. Generació de l'ortomapa: el procés d'ajuntar totes les imatges enregistrades en un sol mosaic. Aquesta part també inclou la validació dels resultats obtinguts, on s'explicarà quin mètode s'ha dissenyat per obtenir una mètrica per avaluar la qualitat d'un ortomapa. Aquest estarà basat en una anotació manual de correspondències (*ground truth*) i un mapa en vista zenital real de la zona sobrevolada.

Un cop explicada tota la feina realitzada s'exposaran els resultats obtinguts i finalment les conclusions extretes de tot plegat. A continuació s'explicarà el context en el qual s'ha desenvolupat aquest treball.

2 Estat de l'art

Actualment existeix software comercial dedicat exclusivament a la creació d'ortomapes a partir d'un UAV fent ús de sensors [1,2,3], que també ofereixen altres serveis més complexos com la generació de mapes en 3D. També existeixen algorismes recents (2013) que generen el mapa utilitzant només la informació de les imatges, com és el cas de *PTAM* [4] i *SVO* [5], però aquests estan restringits pel fet que han de donar els resultats a temps real. D'aquesta forma es pot confirmar que obtenir mapes a partir d'un UAV és viable, ja que actualment s'han aconseguit resultats al respecte en aquest àmbit.

Pel que fa al procés de crear mosaics o panorames donat un conjunt d'imatges basat en les seves característiques (*image matching* i *stitching*), té els seus orígens en el treball de Movarec (1981) [6], el qual va dissenyar el detector de característiques Movarec. Aquest detector després va ser millorat per Harris i Stephens (1988) [7] i en 1992 Harris [8] va demostrar la seva eficàcia per realitzar un rastreig (*tracking*) en imatges. Durant els següents anys hi va haver diferents avenços al respecte, com la detecció d'*outliers* (soroll) per eliminar-los de la solució final [9,10]. L'estat de l'art més recent i important en aquest àmbit, però, es troba en l'article de Brown i Lowe [11], on es detalla tota la metodologia de com passar d'imatges realitzades sobre la mateixa escena a un sol mosaic, el qual és un dels problemes principals a tractar en aquest treball.

- *E-mail de contacte: guimperarnau@gmail.com*
- *Menció realitzada: Computació*
- *Treball tutoritzat per: Felipe Lumbreras i Daniel Ponsa (departament de computació)*
- *Curs 2014/15*

3 Treball realitzat

En aquest apartat s'ha explicat què s'ha fet del projecte i com s'ha fet. Abans, però, s'ha d'especificar que tots els experiments, i en conseqüència totes les figures que contenen imatges del vol, es corresponen a una missió de vol realitzada al camp de vol del Club aeromodelisme Sant Cugat. En aquesta missió l'UAV es troba aproximadament a 50 metres d'alçada i la seqüència de vídeo enregistrada està composada per 3754 frames.

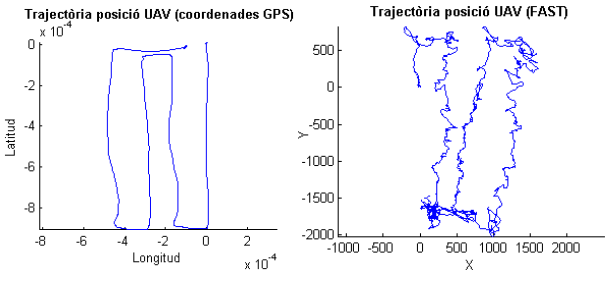
3.1 Sincronitzar les imatges adquirides per l'UAV amb les dades del sensor.

Tal com s'ha mencionat en la introducció, els sensors instal·lats en l'UAV són independents de la càmera que enregistra les imatges. Com a tal, es necessita realitzar una sincronització entre les dades i les imatges per tal d'aprofitar el possible avantatge que comportaria tenir més informació a l'hora de realitzar l'estimació. La idea que s'ha implementat per aconseguir-ho ha estat correlacionar la trajectòria que es pot extreure de les imatges de l'UAV amb la que se'n pot extreure dels sensors. Per exemple, es podria pensar a correlacionar la posició de latitud i longitud del GPS amb la posició de l'UAV estimada per les imatges.

El primer pas, llavors, consisteix a veure quin moviment s'ha produït entre dues imatges de forma local, és a dir, estimar la seva transformació geomètrica, ja que d'aquesta forma se'n podrà extreure una trajectòria. Per estructurar millor aquest treball, però, s'ha decidit agrupar en un mateix apartat tots els mètodes d'estimació entre imatges (3.2), encara que sigui un pas previ a la realització d'aquest objectiu. Per tant, aquest apartat se centrarà únicament en el procés de sincronització.

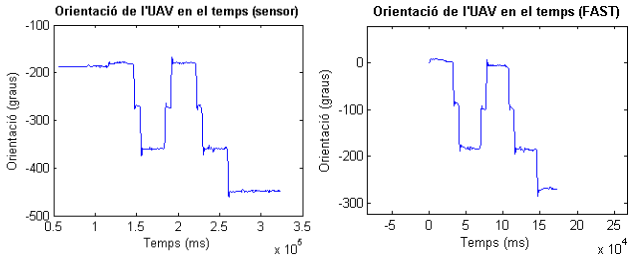
3.1.1 Tipus de trajectòria a sincronitzar.

Es parteix aleshores dels resultats que es detallen en l'apartat 3.2.1, on es conclou que el mètode basat en el detector de característiques *FAST* [13] és el més precís i robust per realitzar estimacions basades en imatges. El que es persegueix ara és generar, a partir de l'estimació, una seqüència de valors que pugui ser fàcilment posada en correspondència amb les dades del sensor. S'han plantejat dues opcions: considerar la posició X i Y de l'UAV, o només considerar les rotacions realitzades al llarg de la seqüència. Com es pot observar a la figura 1, la trajectòria segons la posició conté molt soroll i com a tal la tasca de sincronitzar aquesta trajectòria amb la generada pel sensor (GPS) seria molt complicada. En canvi, a la figura 2 trobem que els canvis en l'orientació de l'UAV són molt semblants amb els canvis d'orientació detectats pel sensor. El més important, però, és que els canvis de l'orientació es produeixen en ambdues seqüències en un temps relatiu molt semblant, de forma que es poden utilitzar aquests punts de referència per realitzar la sincronització. Per tant no importa tant si el grau de rotació de cada trajectòria és més elevat o menys, sinó el moment en el temps on es produeixen, de forma que es podria realitzar una sincronització basada en aquests instants.



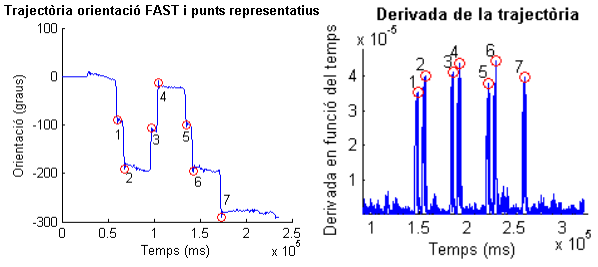
(a) (b)

Fig. 1. Comparació entre les diferents posicions X Y de l'UAV generada pel GPS del sensor (a) i la generada a partir del *FAST* (b). Pel càlcul de les posicions s'han utilitzat tots els frames enregistrats. Es pot observar un excessiu soroll en el cas del *FAST* pel fet que l'UAV no és completament estable mentre està enlairat.



(a) (b)

Fig. 2. La figura (a) mostra l'evolució de l'orientació de l'UAV en el temps mitjançant els sensors, mentre que en (b) s'ha calculat a partir del *FAST*. Les dues seqüències són molt semblants, i encara que no siguin exactament iguals, els punts on hi ha un gran canvi d'orientació en poc temps coincideixen, de forma que es podria utilitzar per posar les dues seqüències en correspondència.



(a) (b)

Fig. 3. En (b) es mostra en l'espai de la derivada de la trajectòria de (a). A l'espai de la derivada es poden detectar els pics més alts i per tant trobar els punts de canvi en els graus de la trajectòria original.

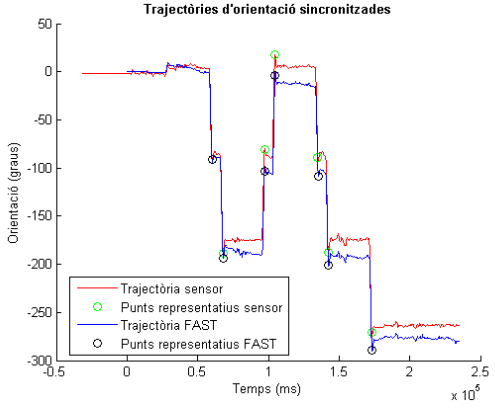


Fig. 4. Les dues trajectòries (la del sensor i la del *FAST*) sincronitzades en el temps mitjançant l'aparellament de punts calculant la diferència del temps.

3.1.2 Posada en correspondència (mapping) de les trajectòries.

Arribat a aquest punt es tenen dues seqüències que van en funció del temps i varien el seu grau de rotació. L'objectiu consisteix ara a trobar el desplaçament en el temps entre les dues orientacions estimades al llarg del temps per tal de fer-les coincidir. Amb això es podrà generar un fitxer on, per cada imatge de la seqüència enregistrada per l'UAV, es tindran totes les dades del sensor. Per extreure la posició dels punts on varia el grau de rotació s'ha calculat la derivada de l'evolució de l'orientació, aconseguint així poder mesurar i ordenar els canvis segons el grau de variació en el temps. Això també permet filtrar tot el soroll que hi pugui haver, ja que el criteri seguit és quedar-se només amb els pics de la derivada que tinguin un valor superior a la meitat del pic més alt. Tot aquest procés queda il·lustrat en la figura 3.

Amb els punts marcats en les dues trajectòries, només queda el pas de posar-los en correspondència. Malgrat això, es podria donar el cas on, a causa del soroll, es detectessin punts en una trajectòria que no figuressin en l'altra. Aleshores s'ha programat un algorisme basat en la idea de programació dinàmica [24] per tal de trobar l'aparellament de punts entre trajectòries de mínim cost. D'aquesta manera, si es detectessin punts inexistents en l'altra trajectòria, s'eliminarien. Es va considerar utilitzar *RANSAC* [14] (més detalls de *RANSAC* en la secció 3.2.1) com a alternativa, però s'ha implementat una solució de programació dinàmica degut a què en proves preliminars fetes amb *RANSAC* es van obtenir resultats menys precisos. Aquest algorisme queda definit de la següent forma:

$$d_{i\,j} = mínim \begin{cases} d_{i-1\,j-1} + De(ip_2, jp_1) (\text{unió } ip_2 \text{ i } jp_1) \\ d_{i-1\,j} + C \text{ (esborrar } ip_2) \\ d_{i\,j-1} + C \text{ (esborrar } jp_1) \end{cases}$$

On:

- d és una taula (M+1)x(N+1) on s'emmagatzemen els resultats parcials. La primera posició de d està inicialitzada a 0.
- M i N representen la quantitat de punts de canvi de rotació de les dues trajectòries.
- ip_x representa el punt i de la trajectòria x.
- C és el cost d'esborrar un punt, amb valor 1000.
- De és la distància euclidiana entre dos punts.

Amb els punts emparellats i restant la diferència en el temps s'obté la sincronització de les dues trajectòries (Fig. 4), permetent així generar un fitxer on per cada imatge captada per l'UAV s'obté la informació del sensor.

3.2 Estimació de la transformació geomètrica entre imatges

La base de tot aquest treball consisteix a estimar el moviment que es produeix entre dues imatges, és a dir, estimar la seva transformació geomètrica. Aquesta estimació es pot realitzar amb diferents mètodes i en aquest apartat es discutiran els següents:

- Basat en la imatge, en el qual únicament s'utilitza la informació extreta de les imatges.

Aquesta aproximació es divideix entre mètodes basats en regions de la imatge (*template*) o basats en detectors i descriptors de característiques.
- Basat en les dades del sensor, on no es requereix la informació de les imatges.
- Basat en l'algorisme *Lucas-Kanade* [15], que permet utilitzar tant les imatges com les dades del sensors.

El resultat ha de ser la transformació geomètrica, és a dir, la translació, rotació i escalat, entre parells d'imatges (Fig. 5). A continuació s'explicarà en detall cada un dels mètodes implementats.

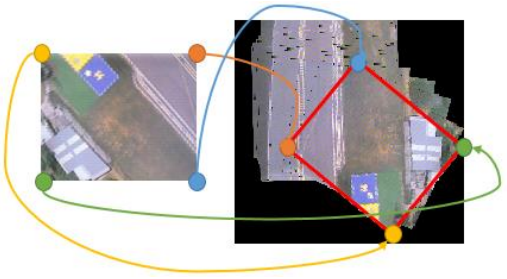


Fig. 5. Imatge essent transformada geomètricament per posar-se en correspondència amb la imatge anterior.

3.2.1 Estimació basada en imatges.

Abans d'explicar la metodologia seguida en l'estimació basada en imatges s'ha de concretar quin preprocessament tenen les imatges (o frames): totes les proves s'han realitzat amb les imatges convertides a escala de gris i agafant totes les files imparelles, ja que la captació d'imatges és en mode entrellaçat. S'ha provat a utilitzar les files parelles i també a convertir les imatges a intensitat (camp *value* de HSV), però no s'aprecia cap canvi significatiu en els resultats, de manera que es pot utilitzar qualsevol d'aquests criteris. Per altra banda les imatges contenen marges negres al voltant que s'han eliminat.

L'alineament entre imatges (trobar punts en comú, *feature matching*) s'ha realitzat seguint dues estratègies: basat en un *template* i basat en punts característics. Seguidament s'analitzaran diferents descriptors i detectors de característiques utilitzats per implementar cada estratègia.

Alineament basat en *template*.

Aquesta estratègia consisteix a retallar una regió de la imatge (*template)* i utilitzar-la com a descriptor per buscar aquest *template* en una segona imatge. La NCC (*Normalized Cross-Correlation* [12]) ha sigut la mesura de similitud utilitzada per implementar aquest mètode. La NCC no és ni un descriptor ni un detector de característiques, però serveix per trobar correlacions entre dues imatges. La NCC, llavors, mesura el grau de correlació entre un *template* i una imatge. A causa d'aquest sistema la NCC no és invariant ni a rotació ni a escala, és a dir, només detecta translacions.

Per tal de buscar correlacions entre les dues imatges s'ha realitzat el següent procediment: retallar de forma prefixada i seqüencial diferents *templates* d'un frame 1, i per cada *template* realitzar un escombrat per tot el frame 2 mesurant la correlació amb la NCC. El *template* del frame 1 que doni un resultat més alt serà l'escollit com a punt de referència entre frame 1 i frame 2. Aquí, però, es troba un problema. Si només s'agafa el *template* amb un resultat més elevat (màxima correlació), això no hauria d'implicar que fos un bon punt de referència, que és el que realment es busca. En altres paraules, una correlació alta no implica que el *template* trobat sigui singular respecte la resta, que és el que interessa per realitzar una estimació del moviment entre imatges. La solució a aquest aspecte es troba explicada a l'apèndix (A1), on es contempla també que el *template* sigui singular respecte la resta de *templates*.

El resultat de tot plegat és la posició relativa X i Y per cada frame respecte a l'anterior, el qual mostra una correcta predicció de la trajectòria segons la translació. Per altra banda, com que no es contemplen les rotacions, quan se'n produeixen l'error augmenta de tal forma que resulta inviable considerar aquest mètode d'estimació. És per això que s'ha implementat la rotació en la NCC aplicant petites rotacions en el *template* original per comparar quin grau de rotació és el que dóna una correlació més elevada. Això, però, augmenta considerablement el temps d'execució, ja que per cada rotació aplicada en el *template* s'ha de repetir tot el procés. Ometent el factor del temps, s'han realitzat diverses proves amb aquesta implementació en un fragment de vídeo on després de realitzar una anotació manual la rotació total acumulada és de 91º, mentre que l'estimada per la NCC és de 48º, un error molt significatiu pel problema tractat.

Finalment es pot concloure que la NCC ha quedat descartada, primer per la seva poca precisió en les rotacions, i segon pel seu elevat temps d'execució, on s'ha mesurat fins 15 segons per trobar la correlació entre dos frames per un rang de rotació de -3º a 3º.

Alineament basat en punts característics.

Per realitzar i validar aquest alineament s'ha utilitzat de referència els articles de Brown et al. [11] i de Bekele et al. [16], on es descriu la metodologia a seguir per realitzar una estimació mitjançant descriptors i detectors de característiques. Els descriptors i detectors que s'han avaluat són els implementats per MATLAB, els quals són els popularment més usats en la bibliografia:

- *Speeded Up Robust Features* (*SURF*) [17]
- *Maximally Stable Extremal Regions* (*MSER*) [18]
- *Features from Accelerated Segment Test* (*FAST*) [13]
- *Minimum Eigenvalue* [19]
- *Harris-Stephens* (*Harris*) [7]
- *Binary Robust Invariant Scalable Keypoints* (*BRISK*) [20]

Aquests descriptors estan caracteritzats per ser invariants a rotació i a escala, de forma que també simplifiquen el problema que ha plantejat la NCC. El primer pas consisteix a detectar característiques entre dues imatges i emparellar-les segons el seu grau de semblança. Seguidament s'utilitza una variació de l'algorisme *RANSAC* [4] que incorpora MATLAB *(M-estimator SAmple Consensus,*

MSAC) per estimar una transformació geomètrica de forma aleatòria i descartar *outliers*. Aquesta transformació geomètrica s'expressa en forma de matriu (Eq. 1), la qual descriu com els punts detectats de la imatge 2 s'han de rotar, escalar i traslladar per coincidir amb els de la imatge 1. Aleshores, a partir de les següents fórmules (Eq. 2, Eq. 3) es pot aïllar la θ i la s per tal d'aconseguir tant l'angle de rotació entre els frames com el factor d'escala, amb els quals es poden establir certs llindars que permeten detectar i eliminar frames sorollosos. De la mateixa forma s'eliminen tots els frames que no aporten informació, és a dir, tots els frames capturats abans que l'UAV estigui completament enlairat.

$$T = \begin{bmatrix} a & -b & 0 \\ b & a & 0 \\ t_x & t_y & 1 \end{bmatrix}, \qquad (1)$$

on $a{=}s \cdot cos\theta$, $b{=}s \cdot sin\theta$, t_z=translació en la dimensió z, s=factor d'escala, θ=angle de rotació.

$$\theta = \arctan(a, b) \qquad (2)$$

$$s = \sqrt{a^2 + b^2} \qquad (3)$$

Avaluació dels descriptors i detectors.

A partir d'aquí s'analitzaran els descriptors i detectors amb mètriques objectives segons s'han definit en diferents articles de la mateixa temàtica [16,21]. Les mètriques a utilitzar seran:

- Capacitat del descriptor: mesurar quants *inliers* (punts ben correlacionats no sorollosos) detecta cada descriptor en una sèrie de frames.
- Precisió de la localització i l'estimació: veure la diferència entre la transformació real entre dues imatges reals i l'estimació del descriptor. Com que no tenim una referència real i fiable (les dades dels sensors poden contenir error), s'anotarà manualment un *ground truth*.

La quantitat d'*inliers* detectats es pot observar en la figura 6a, on amb molta diferència el *FAST* detecta i correlaciona correctament més punts que la resta. La quantitat d'*inliers* per si sola, però, no dóna suficient informació, i és per això que s'ha d'avaluar si els punts detectats per cada descriptor són precisos. Per tal d'avaluar aquesta precisió s'ha creat un *ground truth* dels frames 930-962 (corresponents a una rotació d'aproximadament 90º) anotant de forma manual 10 punts correlacionats entre cada parell de frames. A partir d'això es pot realitzar una estimació fiable de la translació, rotació i escalatge d'aquesta seqüència de frames. El següent pas consisteix en que cada un dels descriptors faci aquesta estimació per comparar l'error respecte de les dades del *ground truth*. Els resultats s'han separat entre la mitjana de l'error en la translació i en la rotació (Fig. 6b i 6c). S'ha omès el factor d'escalatge perquè a priori es manté gairebé constant.

En la figura 6b no apareixen el *Harris* ni el *MSER* perquè tenen una quantitat d'error tan elevada que no es pot apreciar l'error de la resta de descriptors amb un error més baix. En la rotació es dóna un cas semblant, però l'error no desvirtua la proporció de les gràfiques i per tant es poden incloure. Amb aquests resultats destaquen especialment l'*Eigenvalue* i el *FAST*. Sabent que el *FAST* també és el que detecta més punts, es pot concloure que és aquest el descriptor més adequat per les imatges d'aquest projecte, ja que no només és el més robust per tots els punts que detecta, sinó que és dels més precisos.

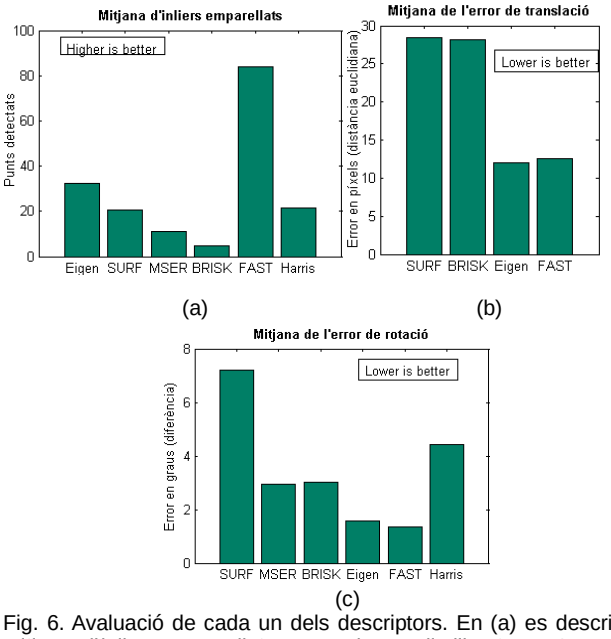


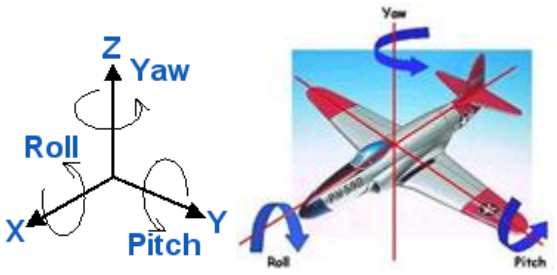
Fig. 6. Avaluació de cada un dels descriptors. En (a) es descriu la mitjana d'*inliers* emparellats per cada parell d'imatges. Com més *inliers* detectats, s'espera més robustesa al soroll en el resultat de l'estimació. En (b) i en (c) es mostra la quantitat d'error que es produeix de mitjana en l'estimació per cada parell d'imatges.

Un cop s'ha trobat el *FAST* com a detector òptim, s'ha d'avaluar quin tipus de transformació geomètrica és la més adequada per les imatges tractades, perquè pot ser projectiva (homografia, 8 graus de llibertat), afí (matriu arbitrària 2x3, 6 graus de llibertat) o similar (escalatge, rotació i translació, 4 graus de llibertat) [22]. Totes les proves que s'han realitzat fins ara han sigut amb la transformació similar perquè és la que a priori té els graus de llibertat adequats per les imatges enregistrades (perspectiva ortogonal). Per comprovar quin és el tipus de transformació òptima s'ha calculat la mitjana de l'error acumulat del *FAST* per cada una de les transformacions. Els resultats en la transformació projectiva respecte a la similar contenen un 17% més d'error en la translació i un 158% en la rotació. Per la transformació afí s'ha obtingut un 4% més d'error en la translació i 109% en la rotació. D'aquesta forma es confirma que la transformació similar és la més adequada per la seqüència d'imatges tractades.

Finalment s'ha provat de millorar la precisió de l'estimació mitjançant un pas més: refinament de l'estimació a nivell subpíxel utilitzant l'algorisme *Lucas-Kanade* [15], procés que es troba explicat a l'apartat 3.2.3.

3.2.2 Estimació basada en sensors de posicionament.

Gràcies al fet que s'ha pogut sincronitzar els sensors amb les imatges (apartat 3.1) ara es pot realitzar una estimació mitjançant la informació dels sensors per cada frame. Més concretament utilitzant la posició de l'UAV segons el GPS (latitud i longitud) i orientació de l'UAV a l'espai (angle de *yaw*, *roll* i *pitch*, representats a la figura 7). Les dades de posicionament (longitud, latitud i altitud) estan respecte



(a) (b)

Fig. 7. Representació de les rotacions *yaw*, *pitch* i *roll*. En (a) es poden observar els eixos sobre els quals es basa cada tipus de rotació, i en (b) es veuen aplicats a un avió.

al sistema de referència de coordenades esfèriques, mentre que les dades d'orientació (*yaw, pitch* i *roll*) estan respecte al camp magnètic terrestre. L'estimació entre parells d'imatges es realitza mitjançant els següents passos:

Càlcul de la posició.

En aquest pas és necessari crear una matriu de translació que defineixi la posició del frame N+1 en funció del frame N. Per fer això s'ha de passar de latitud i longitud (graus) a metres, i seguidament, de metres a píxels. Abans d'aquest últim pas, però, s'han de rotar els metres segons el *yaw* perquè es vol calcular la posició de la imatge segons el sistema de referència de la imatge (X,Y), i no respecte el sistema cartesià (longitud,latitud) (Eq. 4). Per realitzar la conversió de metres a píxels s'ha requerit calcular la focal de la càmera de l'UAV, amb el qual es pot obtenir la relació entre un píxel i un metre (Eq. 5) sempre i quan es conegui la distància entre la càmera i l'escena, en aquest cas l'altitud. Amb això s'obté la posició de l'UAV, però no la posició on es projecta la imatge (Fig. 8). Per simplicitat, i com que es projecten totes les imatges sobre un pla 2D, s'ha decidit utilitzar només la translació associada a les rotacions del *pitch* i el *roll*, i no realitzar la rotació en si. En altres paraules, les imatges no estaran esllavissades i seguiran el procés d'una transformació similar de 4 graus de llibertat (translació X i Y, rotació i escalatge). Per tant, a la posició de l'UAV s'afegeix la translació del *pitch* i el *roll*. Aquesta es calcula amb una simple funció trigonomètrica (Eq. 6).

$$[m'_x\ m'_y] = [m_x\ m_y] \cdot \begin{bmatrix} \cos -\theta & \sin -\theta \\ -\sin -\theta & \cos -\theta \end{bmatrix}, \quad (4)$$

on m_α indica els metres en el sistema cartesià, m'_α els metres en el sistema de referència de la imatge, α la dimensió (x o y) i θ l'angle de rotació del *yaw*.

$$p_\alpha = (f_\alpha \cdot m'_\alpha)\ /h\ , \qquad (5)$$

on p_α són els píxels, f_α el valor de la focal en píxels i h la distància a l'escena (altura).

$$a_x = h \cdot \sin \theta_{roll}$$
$$a_y = h \cdot \sin \theta_{pitch}\ , \qquad (6)$$

on a és el catet oposat (translació en x o en y), h és la hipotenusa (altura) i θ l'angle de rotació del *pitch* o el *roll*.

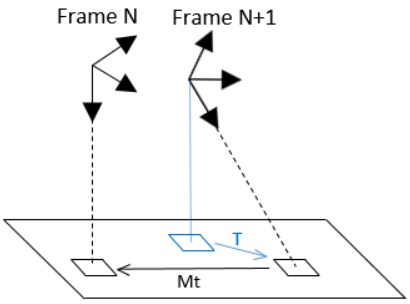


Fig. 8. Esquema de l'estimació realitzada entre dos frames. Els eixos superiors representen la posició i orientació de l'UAV en el frame N i N+1. La matriu Mt estima la transformació geomètrica entre la projecció de la imatge en el frame N+1 a la projecció del frame N. La matriu de translació T afegeix la translació causada pel *pitch* i el *roll* (Eq. 6), permetent calcular la projecció d'un frame.

Càlcul de la rotació i escalatge.

En el cas de la rotació només es tindrà en compte el *yaw*, mentre que per l'escala simplement s'ha d'observar el canvi en l'altura de l'UAV. Com a tal, només s'ha de crear la matriu de rotació i escalatge corresponent. S'ha de tenir en compte un detall, però. La rotació i l'escalatge s'han de realitzar respecte al centre de massa de la imatge (punt central), i no des del punt (0,0), ja que sinó en aplicar la rotació/escalatge es perdria la posició original de la transformació geomètrica. Per tant, abans de realitzar la rotació s'ha de traslladar la imatge al seu centre de massa, escalar i rotar la imatge, i finalment invertir la translació del centre de massa. El càlcul de la matriu de transformació resultant d'aplicar la translació, rotació i escalat es troba detallat en l'Eq. 7.

$$Mt_i^{i-1} = T_1' \cdot S_i^{i-1} \cdot R_i^{i-1} \cdot T_2' \cdot T_i^{i-1}, \quad (7)$$

on Mt_i^{i-1} és la matriu de transformació que posa en correspondència el frame i amb el frame i-1, T_1' la translació al centre de massa de la imatge al (0,0), T_2' la translació oposada de T_1', i S_i^{i-1}, R_i^{i-1} i T_i^{i-1} l'escalatge, rotació i translació del frame i-1 al frame i.

3.2.3 Estimació basada en Lucas-Kanade.

A continuació s'exposen dos mètodes basats en l'algorisme *Lucas-Kanade* [15], que comparteixen gairebé la mateixa implementació i només es diferencien en un pas en concret. Aquests mètodes són:

- Realitzar l'estimació intentant aprofitar els avantatges de les dues aproximacions (basat en imatges i basat en sensors). Els avantatges són: per la banda del *FAST* es busca aconseguir la correlació precisa de forma local entre parells d'imatges, i per la banda dels sensors es busca evitar la propagació d'error acumulat i aconseguir una estructura del mosaic homogènia corresponent al mapa real.
- Intentar millorar el resultat de *FAST+RANSAC* (basat en imatges) treballant a precisió subpíxel.

La idea bàsica de *Lucas-Kanade* consisteix a, donat un fragment d'una imatge 1 (*template*), trobar la transformació geomètrica (o homografia, depenent de la implementació) que posa en correspondència aquest *template* de la

imatge 1 en la imatge 2. Per trobar aquesta transformació es realitza un descens de gradient, és a dir, es va transformant el *template* cap a aquella direcció on disminueix l'error fins que convergeix. A més, l'algorisme permet situar l'origen del descens de gradient, o en altres paraules, la posició i orientació inicials del *template* en la imatge 2. Això últim és el que permet ajuntar en la mateixa estimació la informació tan extreta d'imatges (*FAST*+*RANSAC*) com la dels sensors. El procediment per realitzar l'estimació basada en *Lucas-Kanade* és el següent:

1. Detectar punts característics amb el *FAST* en la imatge N.
2. Crear *templates* segons els punts detectats en la imatge N. Com que en una imatge hi poden haver molts punts *FAST*, i com a conseqüència moltes *templates*, s'ha realitzat un filtre on s'eliminen aquells *templates* que puguin estar sobreposats per tal d'evitar redundàncies i estalviar temps de còmput.
3. Aquest pas varia segons si es busca ajuntar l'estimació basada en imatges i en sensors (a), o si es busca millorar el mètode *FAST*+*RANSAC* (b). Transformar geomètricament els punts *FAST* de la imatge N a la imatge N+1:
    a. mitjançant la informació dels sensors (apartat 3.2.2).
    b. detectant punts *FAST* en la imatge N+1 i aplicant *RANSAC* per extreure l'estimació (apartat 3.2.1).
4. Aplicar *Lucas-Kanade*, del qual s'obté l'estimació geomètrica entre la *template* i la imatge 2, a priori més precisa que l'estimació feta en el pas anterior.
5. Utilitzar *RANSAC* tenint en compte les estimacions realitzades per *Lucas-Kanade* per descartar *outliers* i guanyar robustesa.

Perquè *Lucas-Kanade* sigui més robust quan l'estimació que delimita l'origen del descens de gradient no sigui precisa i estigui lluny de la regió on ha de convergir, s'ha implementat diferents etapes d'aplicació de *Lucas-Kanade* en forma de piràmide multi resolució. Per tant, la primera aplicació de *Lucas-Kanade* es fa a una imatge a menys resolució que l'original, i a cada iteració la resolució va augmentant fins que s'arriba a l'original. D'aquesta forma s'aconsegueix més flexibilitat a l'hora de corregir l'origen del descens de gradient perquè finalment pugui acabar convergint.

Igual que en els altres mètodes, el resultat d'aplicar aquesta estimació consisteix en una transformació geomètrica que descriu l'escalatge, rotació i translació que posa en correspondència el frame N+1 amb el frame N. S'espera, però, que aquesta estimació sigui més precisa, ja que s'ha aplicat amb l'objectiu de millorar les estimacions ja provades.

3.3 Generació ortomapa

Un cop s'ha pogut estimar totes les transformacions entre parells d'imatges, ja es pot passar finalment a el·laborar el mosaic (ortomapa). Aquest apartat se centra amb el procés necessari per confeccionar un mosaic, també anomenat *image stitching*. Els passos es divideixen en tres: per una part el preprocessament de les estimacions realitzades, després la inicialització del mosaic i finalment la seva confecció col·locant cada una de les imatges en el mateix sistema de referència. Per acabar també serà explicat el mètode de validació dissenyat per tal d'avaluar la qualitat dels mosaics resultants.

3.3.1 Preprocessament de les matrius de transformació.

Per tal de generar un mosaic tots els frames han d'estar en el mateix sistema de referència. Una manera d'aconseguir-ho és posant tots els frames en correspondència amb el frame 1 de la seqüència (Eq. 8).

$$Mt_i^1 = \prod_{k=1}^{i} Mt_k^{k-1}, \qquad (8)$$

on Mt_k^{k-1} està definit en l'Eq. 7 i Mt_1^0 és la matriu identitat (el frame 1 està respecte ell mateix).

Existeix llibertat de posar els frames en referència de qualsevol frame de la seqüència, no només del primer. Escollir quin és el frame amb el qual se situen la resta de frames pot suposar en què es propagui menys error en el mosaic, ja que s'evita que l'error entre cada parell de frames s'acumuli excessivament. Per això mateix una bona forma d'evitar aquesta propagació és utilitzar de referència un frame en el centre del mosaic. El procés de posar tots els frames en referència d'un frame *k* està en l'Eq. 9.

$$Mt_i^k = Mt_i^1 \cdot \text{inv}(Mt_k^1)\,, \qquad (9)$$

on la funció inv(A) calcula la inversa de la matriu A.

3.3.2 Inicialització mosaic

Abans de començar a ajuntar totes les imatges en una sola és necessari delimitar el "contenidor" o "espai" on aniran totes aquestes imatges. Com que s'ha enfocat aquest treball de forma *offline* no ens trobem en un cas on s'hagi d'anar creant el mosaic de forma incremental a mesura que s'obtenen imatges, sinó que des del principi es compta amb totes les imatges que formaran part del mosaic. Com a tal, es pot saber que el mosaic tindrà una mida fixe durant tota l'execució, i per tant només és necessari calcular-la un únic cop. Per fer-ho només s'ha d'aplicar a cada un dels vèrtexs de cada frame (és a dir, els punts [1,1],[1,amplitud imatge],[altura imatge,1] i [altura,amplitud]) les matrius de transformació que posicionen cada un dels frames en el mateix sistema de referència (Eq. 10).

$$[x_i' \quad y_i' \quad 1] = [x_i \quad y_i \quad 1] \cdot Mt_i^j, \qquad (10)$$

on x_i i y_i són les coordenades respecte al frame *i* i x_i' i y_i' són les coordenades respecte al mosaic.

Un cop s'han calculat tots aquests punts, es pot saber la mida del mosaic resultant amb els valors màxims i

mínims per X i per Y, sent l'amplitud del mosaic $X_{màxim} - X_{mínim}$ i l'altura $Y_{màxim} - Y_{mínim}$. Amb això, llavors, ja es pot crear el contenidor on es crearà el mosaic. Durant aquest pas també és important tenir en compte que $X_{mínim}$ o $Y_{mínim}$ poden tenir valors inferiors a 1. En el cas de MATLAB la indexació sempre ha de ser igual o superior a 1, de forma que cal saber el desplaçament entre els X o Y mínims per posteriorment corregir l'accés al mosaic.

3.3.3 Confecció mosaic.

Un cop es té tot el mosaic inicialitzat amb les mides i els desplaçaments ja es pot realitzar el procés d'*image stitching*. El procés és el mateix realitzat en l'equació 10, però en comptes de fer-ho només pels vèrtexs de cada imatge, es realitza per cada un dels píxels que conformen les imatges. En el moment de transformar les imatges s'ha de tenir en compte quin tipus de posada en correspondència (*mapping*) s'utilitza: *input-to-output* o *output-to-input*.

El primer cas, *input-to-output*, consisteix a iterar per cada píxel del frame (*input*) per transformar-lo a un píxel del mosaic (*output*). Aquest mètode, però, té el problema que es poden quedar en l'*output* píxels en negre com a conseqüència del fet que la transformació, en cas que sigui escalatge o rotació, requeriria més píxels dels que disposa l'*input*. Per solucionar-ho es pot aplicar la transformació *output-to-input*: iterar per cada píxel del mosaic (*output*) i realitzar la transformació inversa respecte al frame (*input*), ja que el sentit de la transformació és l'oposat. Per evitar iterar innecessàriament per tots els píxels del mosaic amb l'*output-to-input* es calcula una màscara mitjançant *input-to-output* abans de començar el procés per delimitar la regió del mosaic on iterar, estalviant així temps d'execució.

3.3.4 Validació.

Per tal de poder comparar la qualitat dels diferents ortomapes generats s'ha dissenyat un mètode de validació amb una mètrica objectiva. Aquesta mètrica consisteix en l'error basat en la distància euclidiana (Eq. 11) entre punts representatius del mapa real captat per una foto satèl·lit i l'ortomapa generat. Per tant el primer pas ha sigut crear un *ground truth* que relaciona cada frame de la seqüència amb un punt del mapa real. Els punts seleccionats han sigut aquells que fossin característics i fàcilment reconeixibles en l'escena per tal de facilitar la correlació entre mapa real i mapa generat.

$$error\ mosaic = \sum_{i \in N} \sqrt{\left(x_1^i - x_2^i\right)^2 + \left(y_1^i - y_2^i\right)^2}, \quad (11)$$

on N és el conjunt de frames del mosaic, 1 i 2 marquen el conjunt de punts anotats (mapa real o mapa estimat).

Amb el *ground truth* el següent pas consisteix a inicialitzar manualment el primer frame de l'ortomapa a una orientació i a una escala coincidents amb el mapa del satèl·lit, ja que els dos mapes han d'estar en el mateix sistema de referència. Seguidament se situa el primer punt del frame 1 a distància 0 del seu punt corresponent en el mapa del satèl·lit. A partir d'aquí, i projectant els punts dels següents frames, es pot emmagatzemar l'error (distància euclidiana) per cada frame, podent així mesurar la qualitat d'un ortomapa.

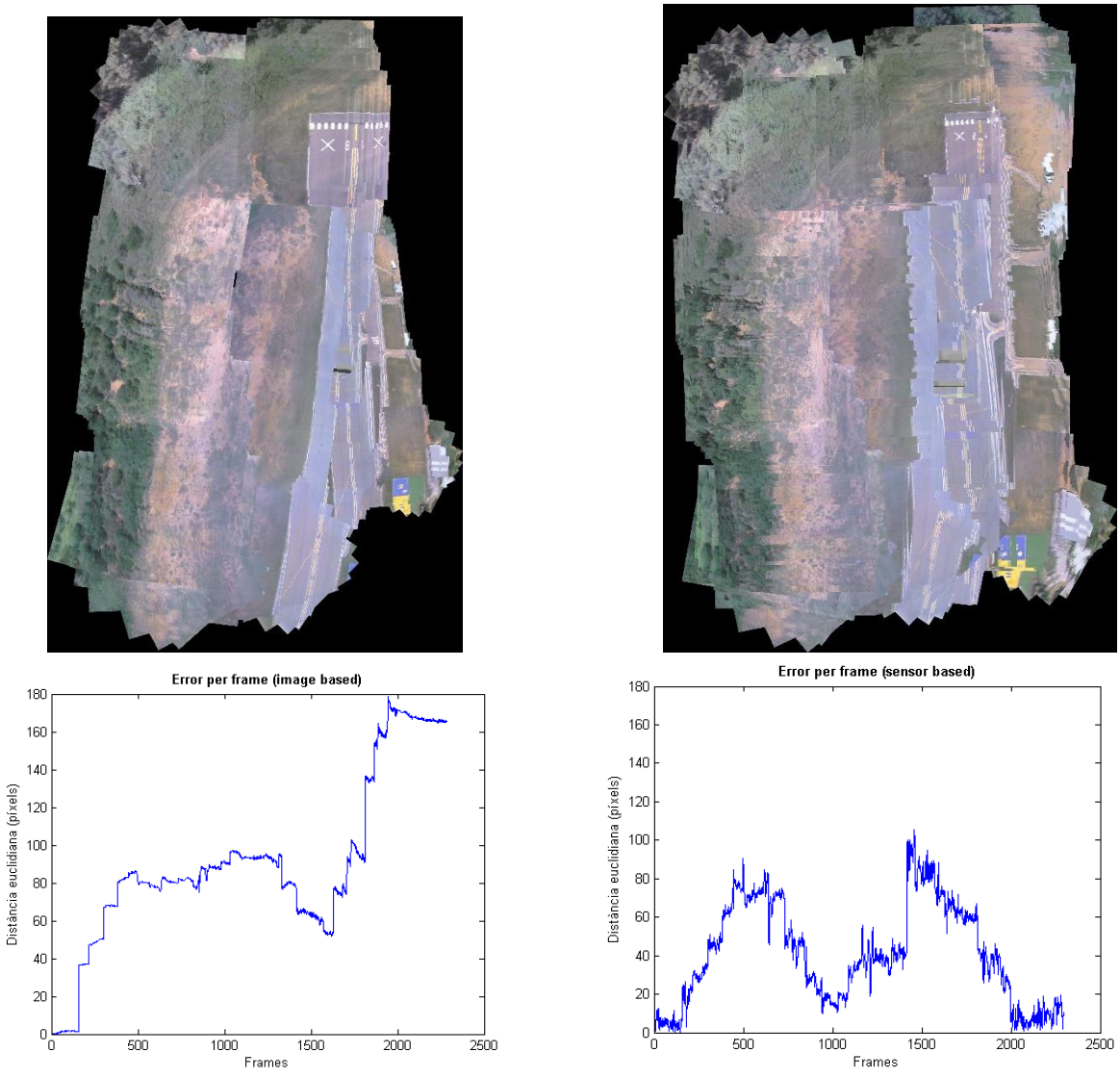
4 RESULTATS FINALS

(Nota: a l'apèndix A2 es mostra un esquema de tot el flux d'execució del treball).

Tal com s'ha mencionat en els anteriors apartats, s'han realitzat tres aproximacions de com realitzar les estimacions del moviment entre imatges per confeccionar l'ortomapa. La primera ha sigut sense utilitzar les dades del sensor, només amb el descriptor de característiques *FAST* i *RANSAC* per estimar la matriu de transformació entre parells d'imatges i descartar *outliers*, la segona només tenint en compte la informació proporcionada pel sensor, i finalment una versió on s'usa l'algorisme *Lucas-Kanade* per millorar els dos mètodes anteriors. A la figura 9 es mostren dos dels tres orto mapes generats amb els seus respectius errors. El tercer mapa que no es mostra és el que utilitza *Lucas-Kanade*, ja que el resultat obtingut en aquest cas és un mosaic on les correspondències són exageradament errònies. Aquest últim aspecte succeeix perquè la transformació del *template* de la imatge 1 a la imatge 2 no és suficientment precisa, resultant en estimacions amb molt error. Aquest error inviable es dóna tant en el refinament a precisió subpíxel de l'estimació basada en imatges, com en l'ús simultani de les dades dels sensors i les imatges. Per aquest motiu l'estimació basada en *Lucas-Kanade*, a falta d'analitzar més profundament si és possible solucionar aquesta falta de precisió, queda descartada de la solució final.

Si s'observa l'error dels dos mosaics restants, es pot comprovar que el mapa generat amb les dades del sensor acaba tenint un error final acumulat tres cops menor respecte el basat en imatges. Això és comprensible perquè el mètode de validació avalua l'estructura del mosaic i el del sensor manté una estructura homogènia, ja que les mesures del sensor són absolutes (la transformació del frame N no depèn de l'estimada als frames precedents), i per això l'error no es propaga ni s'acumula. Per això mateix l'últim frame està molt pròxim al primer (l'UAV fa un circuit tancat), tal com es pot veure en la poca quantitat d'error en els últims frames. Tot i això, els sensors no aconsegueixen bona precisió local, ja que també contenen error i no s'arriba a donar una precisió molt elevada entre imatges. Justament passa el contrari amb el mosaic basat en imatges: entre imatges la precisió és molt bona, ja que és un mètode centrat a correlacionar dues imatges de forma local. Com a conseqüència d'això, l'error es propaga a mesura que es van acumulant imatges, creant una deriva en l'error que es veu reflectida sobretot en els últims frames, on l'error per frame és màxim. Per això l'estructura del mapa no és tan consistent com la del basat en el sensor.

Cal remarcar també que els mosaics mostrats en la figura 9 contenen tota la seqüència de frames, i com que els últims frames se sobreposen amb els anteriors, hi ha frames que poden quedar ocultats per frames posteriors. Per això a l'apèndix A3 es troben els mapes en diferents fases de creació i el mapa original. Per altra banda, el resultat



(a) (b)

Fig. 9. Ortomapes generats per diferents mètodes i el seu respectiu error. En (a) el mosaic s'ha generat a partir de la informació de les imatges (*FAST+RANSAC*), mentre que en (b) només s'ha utilitzat la informació del sensor. Els gràfics de l'error detallen l'error (distància euclidiana entre mapa real i mapa estimat) per cada frame individualment. En (a) es pot observar com l'error es va acumulant, arribant al seu màxim en els últims frames, mentre que en (b) l'error és més inestable (poca precisió local), però no es produeix deriva en l'error.

final podria semblar més imprecís del que es podria esperar, però s'ha de mencionar que el mosaic està format per més de 2500 frames, i és en casos com aquest, amb un nombre de frames tan elevat, on l'error és més notable (en especial el cas del mapa basat en imatges, on l'error s'acumula). Amb seqüències de frames de l'ordre de 100-300, per exemple, visualment s'aprecia un mapa molt més consistent i semblant al mapa real. Per tant, en analitzar els resultats visuals, s'ha de tenir en compte la quantitat de frames amb la que s'està treballant.

5 CONCLUSIONS

En la realització d'aquest treball s'ha pogut observar tot el procés necessari per a confeccionar un ortomapa a partir d'imatges enregistrades amb un UAV. Tot i ser aquest l'objectiu principal, també s'han tractat altres aspectes com la sincronització automàtica de les dades del sensor amb la seqüència de vídeo. Per dur a terme aquest projecte també s'ha après a utilitzar la metodologia adequada de planificació i organització per tal de facilitar el desenvolupament, seguiment i correcció d'errors del treball.

S'ha pogut avaluar cada un dels descriptors i detectors de característiques de l'estat de l'art més utilitzat, resultant en el *FAST*, en combinació amb l'algorisme *RANSAC*, el que millors resultats ha donat per l'alineament d'imatges consecutives. Aquests resultats han permès poder realitzar la sincronització amb les dades del sensor amb la precisió requerida.

Respecte al procés de creació de l'ortomapa s'ha vist que les diferents aproximacions per realitzar-lo tenen cada una diferents avantatges i inconvenients:

- Basat en imatges:
  - Precisió local entre imatges molt bona.
  - L'estructura del mapa es veu deformada per la deriva que suposa l'acumulació d'error.

- Basat en dades dels sensors:
  - Ofereix una estructura del mapa ben definida.
  - Poca precisió de forma local entre imatges.

- Basat en *Lucas-Kanade*:
  - Inviable amb la implementació actual per falta de precisió significativa en l'estimació.

Com que el procés de validació utilitzat té en compte l'estructura de tot el mapa, ha sigut el mosaic basat en els sensors el que ha donat el millor resultat. El criteri per decidir quin mapa és millor, però, dependrà de les necessitats de l'usuari final (prioritat en detalls locals entre imatges, prioritat en què l'estructura sigui fidel al mapa real, etc).

Finalment hi ha diferents aspectes que són susceptibles a millorar enfocats a una continuació d'aquest treball:

- Actualment l'aproximació utilitzada per l'algorisme *Lucas-Kanade* no dóna els resultats esperats, però faria falta una anàlisi més profunda per veure el motiu d'aquesta imprecisió, i un cop trobat, veure si es pot solucionar o si finalment es descarta aquest algorisme per a la generació d'ortomapes.
- En un principi s'havia planificat l'aplicació de l'algorisme *bundle adjustment* [23], que finalment no s'ha pogut realitzar a causa de l'elevada complexitat que suposava (llibreries poc intuïtives i molts canvis en la implementació de codi ja fet). El mètode *bundle adjustment*, per tant, és una oportunitat que encara queda per provar, on s'esperaria una optimització global de l'error acumulat que es produeix en el mapa basat en imatges.
- Optimitzar tot el codi passant-lo a un llenguatge de baix nivell com C o C++ i aplicar paral·lelisme. Tot i que *MATLAB* ofereix la comoditat de tenir moltes funcions implementades que han afavorit enormement el desenvolupament d'aquest projecte, el seu rendiment és millorable, i de cara a un producte final real el programa hauria de tenir una eficiència més elevada.

AGRAÏMENTS

Vull mostrar el meu agraïment a Felipe Lumbreras i Daniel Ponsa, tutors d'aquest treball de fi de grau. He rebut el seu suport en forma d'idees, recomanacions d'articles i valuosos consells, tant en els informes entregats com en tot el programari implementat. Gràcies a aquesta ajuda he pogut aprendre molt més i augmentar la qualitat d'aquest treball.

BIBLIOGRAFIA

[1] EnsoMOSAIC for UAVs, Web. 25 Gen. 2015 <http://www.mosaicmill.com/products/software/ensomosaic_uav.html>

[2] Pix 4D, Web. 25 Gen. 2015. <http://pix4d.com/>

[3] CATUAV, Web. 25 Gen. 2015. <http://www.catuav.com>

[4] S. Weiss, M. and W. Achtelik *et al.*, "Monocular Vision for Long-term Micro Aerial Vehicle", *Journal of Field Robotics*, Volume 30, Issue 5, Autonomous Systems Lab, Zurich, Germany, August 2013, pp. 803-831.

[5] C. Foster, M. Pizzoli and D. Scaramuzza, "SVO: Fast Semi-Direct Monocular Visual Odometry", Robotics and Perception Group, University of Zurich, Switzerland, 2013

[6] H. Movarec, "Rover visual obstacle avoidance", *International Joint Conference on Artificial Intelligence*, Vancouver, Canada, 1981.

[7] C. Harris and M. Stephens, "A combined corner and edge detector", *Fourth Alvey Vision Conference*, Manchester, UK, 1988.

[8] C. Harris, "Geometry from visual motion", Active Vision, *MIT Press*, 1992.

[9] Z. Zhang and R. Deriche *et al.*, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry", *INRIA*, nº 2273, France, 1994.

[10] T. Torr, "Motion segmentation and outlier detection", *Ph.D. Thesis*, Dept. of Engineering Science, University of Oxford, UK, 1995.

[11] M. Brown and D. Lowe, "Automatic Panoramic Image Stitching using Invariant Features", *International Journal of Computer Vision 74*, Department of Computer Sciene, University of British Columbia, Canada, 2007.

[12] J. P. Lewis, "Fast Normalized Cross-Correlation", *Vision interface 10*, 1995, pp. 120-123.

[13] E. Rosten and T. Drummond, "Fusing Points and Lines for High Performance Tracking", *tenth IEEE International Conference*, University of Cambridge, Cambridge, UK, 2005.

[14] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography", *Communications of the ACM*, 1981.

[15] S. Baker and I. Matthews, "Lucas-Kanade 20 Years on: A Unifying Framework", *International Journal of Computer Vision 56(3)*, The Robotics Institute, Carnegie Mellon University, Pittsburgh, USA, 2003, pp. 221–255.

[16] D. Bekele, M. Teutsch, T. Schuchert, "Evaluation of binary keypoint descriptors", Karlsruhe Institute of Technology, Fraunhofer Institute of Optronics, System Technologies and Image Exploitation, Karlsruhe, Germany, 2013, pp. 1-4.

[17] H. Bay, T. Tuytelaars and L. Van Gool, "SURF: Speeded Up Robust Features", *Lecture Notes in Computer Science Volume 3951*, ETH Zurich, Katholieke Universiteit Leuven, 2006, pp. 404-417.

[18] J. Matas and O. Chum *et al.*, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions", *In Proceedings of the British Machine Vision Conference (BMVC)*, UK, 2002.

[19] J. Shi and C. Tomasi, "Good Features to Track", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593-600.

[20] S. Leutenegger, M. Chli and R. Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints", *Proceedings of the IEEE International Conference*, ICCV, 2011

[21] K. Mikolajczyk and C. Schmid, "Scale & Affine Invariant Interest Point Detectors", *INRIA Rhne-Alpes*, GRAVIR-CNRS, France, 2004, pp. 13-16.

[22] B. Morse, "Basics of Geometric Transformations", CS 650: Computer Vision, pp. 10-14

[23] W. Triggs and P. McLauchlan, *et al.*, "Bundle adjustment: A modern synthesis", *Vision Algorithms: Theory and Practice, number 1883 in LNCS*, Springer-Verlag, Corfu, Greece, 2002, pp. 1-7.

[24] R. Bellman, "The theory of dynamic programming", *Bulletin of the American Mathematical Society 60*, nº 6, 1954, pp.503-515.

APÈNDIX

A1. PROBLEMA NCC

Tal com es comentava en l'apartat 3.2.1, agafar el *template* amb màxima correlació no assegura que aquell *template* sigui un punt ben correlacionat. El que es busca, per tant, és la màxima correlació i a la vegada que el *template* sigui singular en tota la imatge. Un exemple d'aquest problema es pot observar en la figura A1.1, on el *template* de màxima correlació és una regió de la imatge poc singular (tota la línia de la carretera és molt semblant), que podria dur a ambigüitats en l'estimació de moviment. La solució a aquest problema, aleshores, consisteix a crear un llindar, amb el qual només s'acceptarà un *template* en cas que la quantitat de valors elevats (*templates* amb alta correlació) no sigui més alta que tal llindar. S'han realitzat diferents proves amb diferents llindars fins a arribar a un compromís entre robustesa i exclusió de *templates*. A la figura A1.2 es pot observar el *template* trobat després d'aplicar aquest llindar.

Tot aquest resultat també s'hauria pogut aconseguir amb la mateixa estratègia utilitzada amb els descriptors i detectors de característiques + *RANSAC*, és a dir, aparellar cada una de les relacions entre *templates* trobades, i descartar els resultats erronis mitjançant *RANSAC*. Aquesta alternativa, però, no es va considerar en el moment d'experimentar amb el mètode NCC perquè el projecte es trobava encara en una etapa inicial on encara no s'havia provat aquest mètode alternatiu. De qualsevol forma, haver aplicat el mètode NCC mitjançant *RANSAC* no hauria canviat la conclusió final en la qual es descarta aquest mètode per realizar estimacions, ja que l'alt temps d'execució i la no invariancia a rotació i a escala no depenen d'aquest factor.

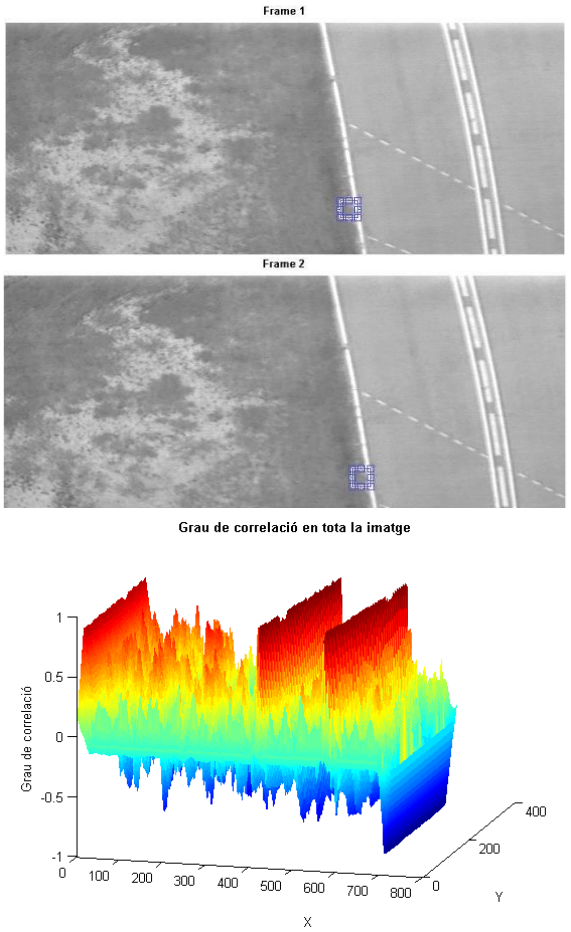


Fig. A1.1. Al frame 1 es mostra el *template* de màxima correlació amb una regió del frame 2. Aquest *template*, però, és poc singular, tal com es pot observar a la gràfica del grau de correlació, on hi ha molts pics. Això pot portar a ambigüitats en la correlació.

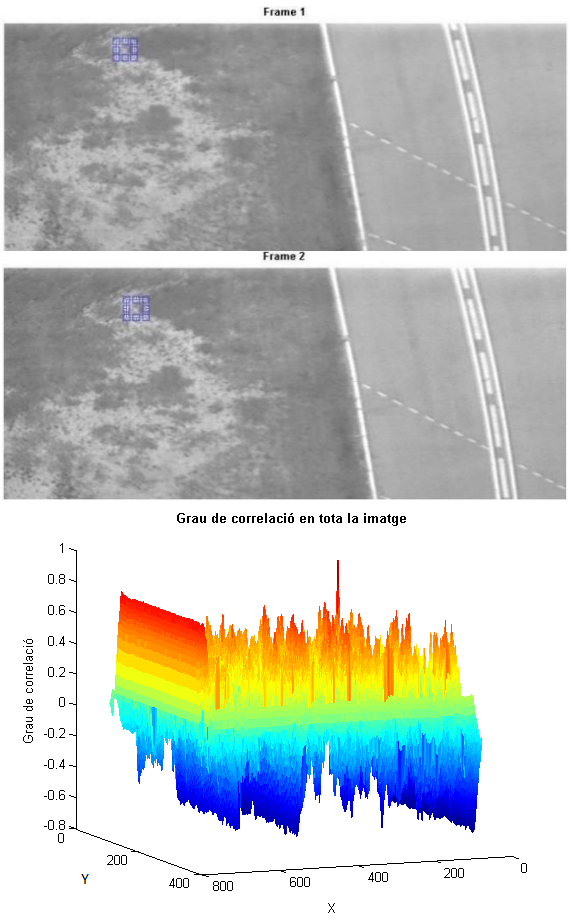


Fig. A1.2. Afegint un llindar per arribar a un compromís entre singularitat i grau de correlació s'aconsegueix un *template* més adequat per fer una correlació més robusta i menys sensible a patrons repetitius, tal com es pot apreciar en l'únic pic de la gràfica de correlació.

A2. ESQUEMA D'EXECUCIÓ GENERAL DE TOT EL TREBALL

A continuació es mostren dues figures que resumeixen en forma d'esquema els aspectes més importants d'aquest treball, per una fàcil comprensió de tota la idea general que comprèn. En la figura A2.1 es mostra l'esquema general del treball, mentre que en la figura A2.2 es detalla el funcionament de cada una de les estimacions del moviment entre imatges.

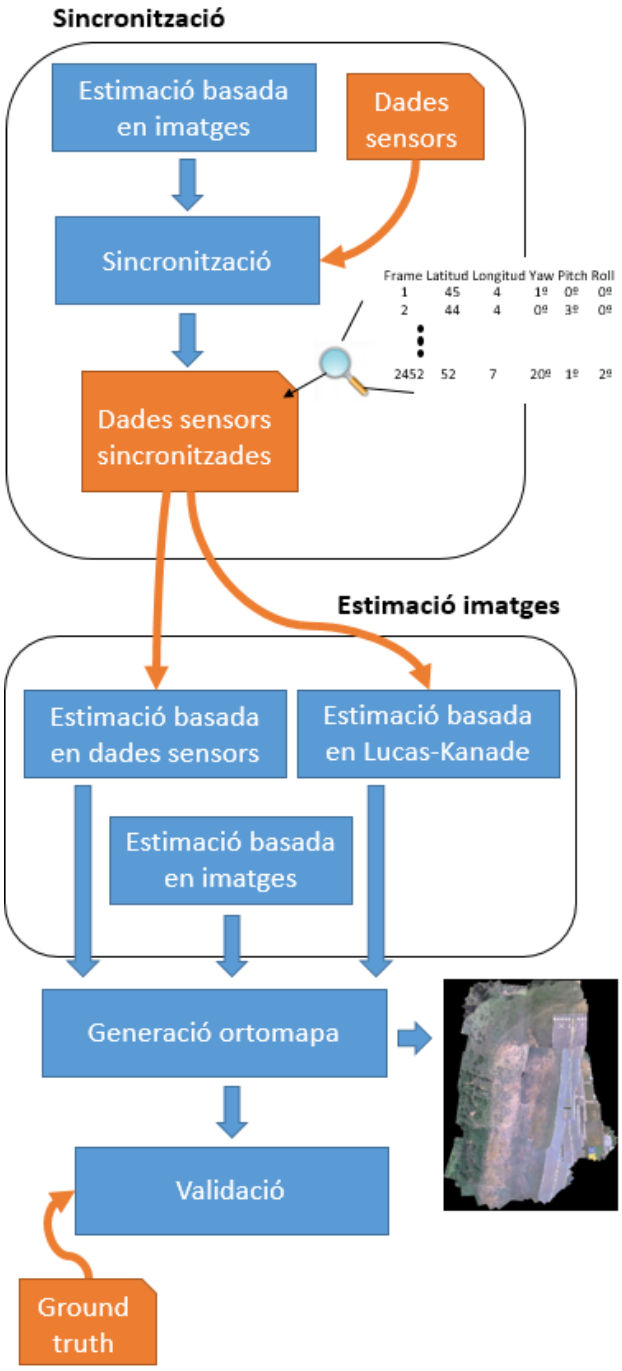


Fig A2.1 Fluxe d'execució de tot el treball: sincronització, estimació del moviment entre imatges i generació de l'ortomapa.

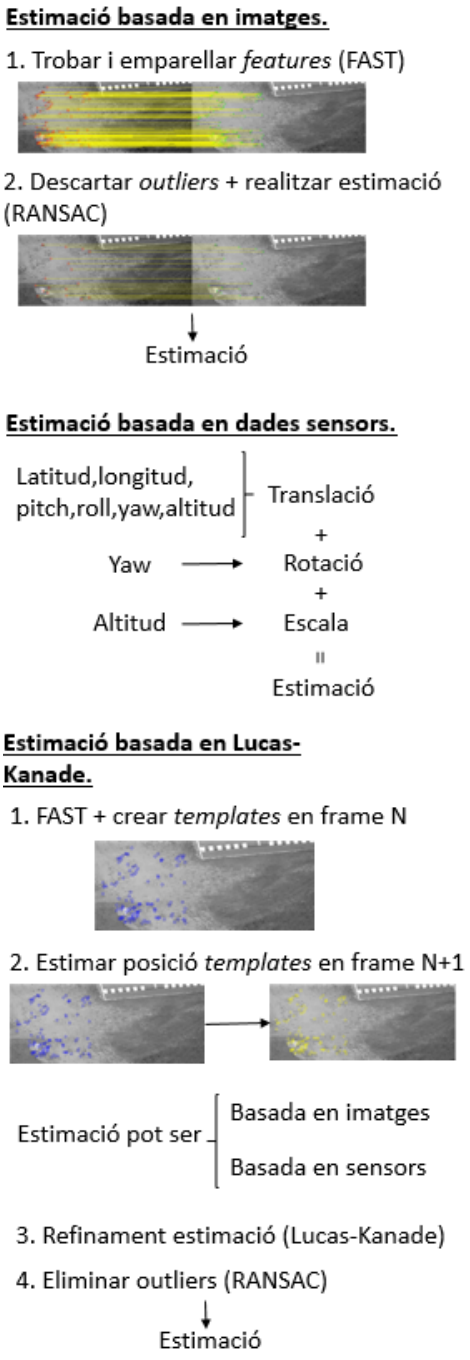
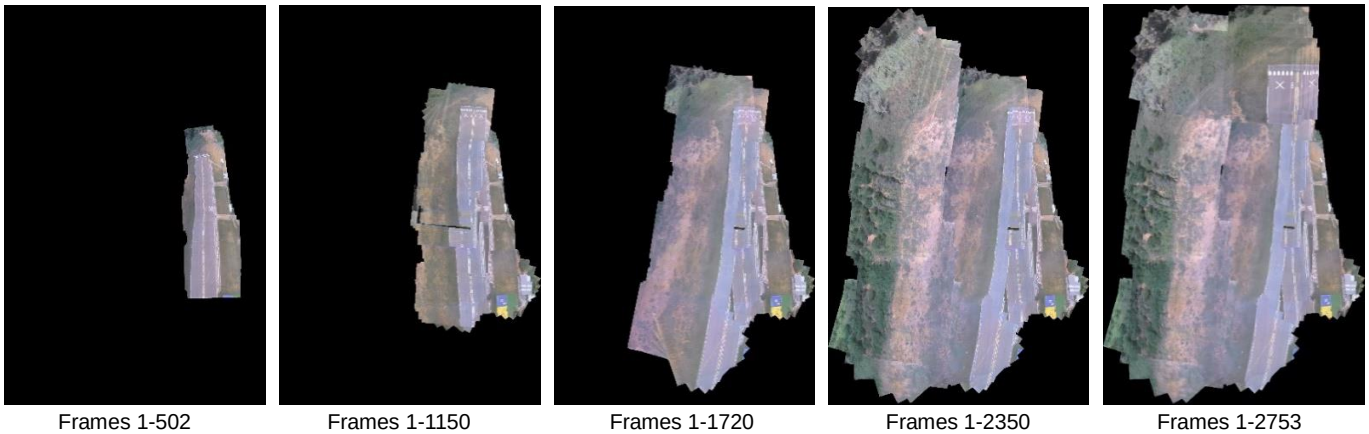


Fig. A2.2 Detall del procés que realitza cada una de les estimacions provades.

A3. DIFERENTS FASES DE CREACIÓ DEL MOSAIC

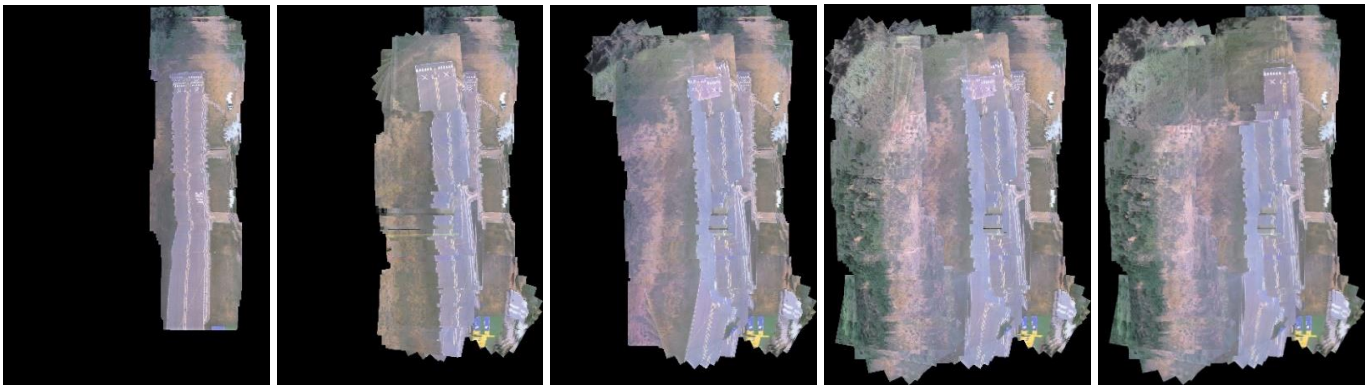
En aquest apartat de l'apèndix es mostren les diferents fases de confecció de l'ortomapa (Fig. A3.2, A3.3). D'aquesta forma es poden apreciar aquelles regions que, en el resultat final, podrien quedar ocultes per frames posteriors. Per altra banda també es mostra el mapa real de la zona sobrevolada (Fig. A3.1) per tal que es pugui comparar el nivell de semblança entre el mapa estimat i el real. Aquest mapa real és també el que s'ha utilitzat per fer l'anotació manual del *ground truth* utilitzat per avaluar la qualitat dels ortomapes estimats.

Fig. A3.1. Mapa real de la regió on es realitza la missió de vol de l'UAV.



Frames 1-502 Frames 1-1150 Frames 1-1720 Frames 1-2350 Frames 1-2753

Fig. A3.2. Evolució de la creació de l'ortomapa basat en imatges.



Frames 1-502 Frames 1-1150 Frames 1-1720 Frames 1-2350 Frames 1-2753

Fig. A3.3. Evolució de la creació de l'ortomapa basat en les dades dels sensors.