



Università degli Studi di Ferrara

FACOLTÀ DI SCIENZE MM. FF. NN.

CORSO DI LAUREA SPECIALISTICA IN INFORMATICA

SVILUPPO DI SOFTWARE PER L'ELABORAZIONE E LA MANIPOLAZIONE DI DATI SATELLITARI LANDSAT E SPOT

Relatore:

Dott. MIRCO ANDREOTTI

Laureando:

MARIO CAVICCHI

Correlatore:

Dott. SIMONE MANTOVANI

Anno Accademico 2006-2007

*Ai miei genitori
e a Clarissa, l'ammore della mia vita*

Indice

| | |
|--|-----------|
| Introduzione | 1 |
| 1 Dati satellitari LANDSAT e SPOT | 3 |
| 1.1 Overview dei servizi sviluppati | 3 |
| 1.2 Sensori satellitari | 4 |
| 1.3 Archivi dati satellitari | 6 |
| 1.3.1 Landsat Level 1G | 7 |
| 1.3.2 Landsat GeoCover collection | 10 |
| 1.3.3 Landsat CEOS | 11 |
| 1.3.4 SPOT DIMAP | 12 |
| 1.4 Formati di immagine | 12 |
| 1.4.1 Formato TIFF/GeoTIFF | 12 |
| 1.4.2 Formato CEOS | 14 |
| 2 Software I/O | 17 |
| 2.1 Software/librerie di I/O | 17 |
| 2.2 Librerie disponibili LANDSAT e SPOT | 18 |
| 2.2.1 Librerie libtiff | 18 |
| 2.2.2 Librerie libgeotiff | 19 |
| 2.2.3 Geospatial Data Abstraction Library (GDAL) | 19 |
| 2.3 Software design e prestazioni | 22 |
| 3 Il portale SSE, servizi di elaborazione ed integrazione | 25 |
| 3.1 Servizi KEI | 26 |

| | | |
|----------|---|-----------|
| 3.1.1 | Format Check | 26 |
| 3.1.2 | Format Conversion | 28 |
| 3.1.3 | Clipping | 31 |
| 3.1.4 | LIRCC | 39 |
| 3.1.5 | LISC | 41 |
| 3.2 | Script di esecuzione dei servizi | 43 |
| 3.3 | Gestione degli errori | 45 |
| 4 | Servizi concatenati | 47 |
| 4.1 | Analisi delle possibili catene di processamento | 47 |
| 4.2 | Reconditioning | 48 |
| 4.3 | Categorization | 48 |
| 4.4 | Modifiche software e script | 50 |
| 5 | Test di verifica sui servizi | 53 |
| 5.1 | Overview dei test eseguiti | 53 |
| 5.2 | Problemi durante lo sviluppo e i test | 58 |
| 6 | Conclusioni | 63 |
| A | Codici sorgente | 65 |
| A.1 | Programma per il clipping: clipgtiff | 65 |
| A.1.1 | clipgtiff.c | 65 |
| A.1.2 | LatLong-UTMconversion.c | 73 |
| A.1.3 | LatLong-UTMconversion.h | 77 |
| A.2 | Script per la concatenazione dei servizi: fullchain | 78 |
| A.2.1 | fullchain.sh | 78 |
| B | Acronimi | 85 |
| | Bibliografia | 88 |

Elenco delle figure

| | | |
|------|--|----|
| 1.1 | Sistema di georeferenziazione World Reference System (WRS-2) . . . | 8 |
| 1.2 | Schema dei riferimenti in un'immagine TIFF. | 13 |
| 1.3 | Schema dei record di un'immagine CEOS nel Leader File. | 15 |
| 2.1 | Schema del funzionamento delle librerie GDAL | 21 |
| 3.1 | Diagramma del servizio Format Check. | 27 |
| 3.2 | Diagramma del servizio Format Conversion. | 31 |
| 3.3 | Esempio di taglio di un'immagine ruotata. | 32 |
| 3.4 | Archivio Landsat Level 1G (L71039037_03720030325.zip) | 33 |
| 3.5 | Archivio Landsat Level 1G (L71039037_03720030325-ss.zip) | 34 |
| 3.6 | Diagramma del servizio di Clipping. | 38 |
| 3.7 | Diagramma del servizio di LIRCC. | 40 |
| 3.8 | Diagramma del servizio di LISC. | 42 |
| 3.9 | Archivio Landsat CEOS (LN5-cal.zip) | 43 |
| 3.10 | Archivio Landsat CEOS (LN5-cal-SpCI72.zip) | 44 |
| 4.1 | Diagramma del servizio di Reconditioning | 49 |
| 4.2 | Diagramma del servizio di Categorization. | 50 |
| 4.3 | Schema del servizio di Categorization | 52 |

Elenco delle tabelle

| | | |
|-----|--|----|
| 1.1 | Bande di acquisizione del sensore Thematic Mapper (TM) | 5 |
| 1.2 | Bande di acquisizione del sensore Enhanced Thematic Mapper Plus (ETM+) | 6 |
| 1.3 | Bande di acquisizione del sensore HRVIR | 6 |
| 1.4 | Bande di acquisizione del sensore High Resolution Geometric (HRG) | 6 |
| 5.1 | Lista degli Acceptance Test eseguiti sui servizi KEI. | 54 |
| 5.1 | Lista degli Acceptance Test eseguiti sui servizi KEI. | 55 |
| 5.1 | Lista degli Acceptance Test eseguiti sui servizi KEI. | 56 |
| 5.1 | Lista degli Acceptance Test eseguiti sui servizi KEI. | 57 |
| 5.2 | Acceptance Test per il servizio di Clipping | 58 |
| 5.3 | Acceptance Test per il servizio di Format Conversion | 59 |
| 5.4 | Acceptance Test per il servizio di Format Check | 60 |
| 5.5 | Acceptance Test per il servizio di LIRCC | 61 |
| 5.6 | Acceptance Test per il servizio di LISC | 62 |

Introduzione

Il lavoro di tesi proposto riguarda la progettazione e lo sviluppo di applicazioni software per l'elaborazione di immagini satellitari. L'azienda presso la quale è stato svolto il periodo di tirocinio, le cui attività sono oggetto del presente lavoro, si occupa di Osservazione della Terra da Satellite. Meteorological and Environmental Earth Observation (MEEO) progetta e realizza sistemi di analisi in grado di estrarre, elaborare e sintetizzare informazioni sul sistema Terra-Atmosfera partendo dai dati raccolti dai satelliti per l'Osservazione della Terra attualmente in orbita. In particolare il progetto per il quale ho sviluppato gli applicativi descritti in seguito ha come obiettivi la realizzazione di nuovi strumenti per l'elaborazione di dati per l'integrazione degli stessi in un portale dell'Agenzia Spaziale Europea. Tale portale, denominato Service Support Environment (SSE), è stato realizzato allo scopo di diffondere le applicazioni per l'elaborazione di dati satellitari rendendoli facilmente disponibili ed utilizzabili ad utenti esperti e non esperti. Il sistema ESA-SSE è infatti concepito per utenti come le pubbliche amministrazioni, le aziende private, le istituzioni internazionali che necessitano di qualunque tipo di informazioni legate al territorio, dall'inquinamento dell'aria e del suolo, allo sviluppo urbano, alle mappe di rischio di frane, esondazioni, ecc.. Il progetto a cui ho partecipato è denominato KIM Extensions and Installation (KEI) ed è stato promosso dall'ESA come estensione del progetto Knowledge based Information Mining (KIM). Si tratta di un programma pluriennale per una generazione ed una gestione più efficiente e più utile dell'enorme quantità di dati raccolti tramite le osservazioni satellitari. Le problematiche che sono state affrontate durante il tirocinio riguardano principalmente la manipolazione di immagini costituite da file di grandi dimensioni (tipicamente dell'ordine di centinaia di megabyte) e la grande diversità

dei formati di archiviazione utilizzati dai consorzi e dagli enti che si occupano della raccolta e distribuzione dei dati. Inoltre gli obiettivi operativi del tirocinio prevedevano lo sviluppo di software nel linguaggio C in ambiente Linux con lo specifico scopo di aumentare le performance di calcolo degli algoritmi; tuttavia tale linguaggio a differenza dei software proprietari specifici per le immagini satellitari, necessita di un lavoro d'implementazione di librerie *ad hoc*. Il software sviluppato è costituito da una serie di moduli che sono stati chiamati servizi in quanto ciascun modulo realizza un servizio di elaborazione pubblicato sul portale SSE. I pacchetti che sono stati interamente progettati e sviluppati sono il Format Check (fck) per effettuare il check del formato delle immagini, il Clipping (clipgtiff) per il taglio di sotto-scene da immagine originale ed il Format Conversion (fcv) per la conversione del formato delle immagini, mentre sono stati ricodificati dei moduli software per la calibrazione radiometrica (LIRCC) e la classificazione spettrale (LISC) di immagini allo scopo di migliorare le prestazioni rispetto a quelli esistenti. Inoltre ho realizzato a livello di scripting due catene di processamento costituite dai singoli servizi descritti in precedenza. Tutti i servizi sviluppati, sia quelli singoli che quelli concatenati, sono stati testati sia internamente all'azienda che direttamente da personale dell'ESA prima di essere pubblicati in modo operativo sul portale SSE.

Il lavoro di tesi è stato realizzato introducendo brevemente nel primo capitolo alcuni concetti sui satelliti per l'Osservazione della Terra, i sensori montati a bordo di tali satelliti e i dati da essi prodotti. Il secondo capitolo descrive le librerie specifiche utilizzate per la manipolazione dei dati satellitari, le modalità di accesso e gli schemi di funzionamento. Il terzo e il quarto capitolo descrivono i pacchetti sviluppati e la loro integrazione nel portale SSE; si è scelto di dividere questo argomento in due capitoli in quanto parte dei pacchetti sviluppati sono stati realizzati come concatenazione dei pacchetti di base accennati in precedenza. Si è ritenuto necessario descrivere a parte la realizzazione dei pacchetti concatenati allo scopo di esprimere i concetti con maggior chiarezza. Infine, nel capitolo dedicato alle conclusioni, si è cercato di focalizzare la discussione sugli aspetti qualificanti del lavoro svolto sia per l'esperienza acquisita sia per l'ambito tecnologico sicuramente innovativo nel quale è stato svolto il lavoro.

Capitolo 1

Dati satellitari LANDSAT e SPOT

1.1 Overview dei servizi sviluppati

I servizi sviluppati durante la tesi costituiscono un pacchetto di strumenti per la manipolazione e l'elaborazione di immagini satellitari. Ogni software implementato deve quindi poter accettare in input un archivio di immagini satellitari conforme agli standard descritti in dettaglio nei capitoli successivi e generare in output un altro archivio consistente con i medesimi standard.

Le soluzioni concepite durante lo sviluppo del software sono state per la maggiore parte rivolte a creare strumenti che processino o elaborino i dati in input in modo da poter ricreare dati di output sempre conformi e consistenti con le tipologie di archivi satellitari più diffusi, intendendo per archivi satellitari sia i file contenenti l'immagine vera e propria sia i file correlati contenenti le informazioni aggiuntive per la geolocalizzazione e la calibrazione radiometrica dei dati (metadata file).

Nei paragrafi successivi verranno descritti gli standard dei formati di immagine e di archivi di immagini utilizzati per l'implementazione dei vari strumenti software che compongono i servizi.

1.2 Sensori satellitari

Nell'ambito dell'Osservazione della Terra da Satellite sono stati sviluppati e realizzati sensori multi-spettrali in grado di raccogliere dati in diverse bande dello spettro elettromagnetico e che risultano utili allo sviluppo di applicazioni in ambito meteorologico e climatico, di applicazioni per il monitoraggio ambientale, o per il monitoraggio del territorio. A quest'ultimo ambito, in particolare, appartengono i sensori che consentono di ricavare informazioni approfondite sulla natura del suolo: attraverso l'applicazione di uno specifico algoritmo di processamento (sviluppato da MEEO e denominato *SOIL MAPPER*) di immagini multi-spettrali è possibile infatti determinare la natura del tipo di suolo, discriminando all'interno della scena osservata da satellite pixel di vegetazione, di suolo scoperto, di acqua, di ghiaccio, ecc.. Esistono inoltre sensori che, pur non essendo stati concepiti col preciso intento di osservare le proprietà del suolo, possiedono caratteristiche spettrali sufficienti per l'applicazione di una varietà di algoritmi di elaborazione. Viste le ampie possibilità offerte dalle numerose combinazioni satellite-sensore, si è dovuto operare una prima scrematura tra i satelliti esistenti, selezionando quelli che montano sensori che permettono l'acquisizione di bande spettrali utili all'osservazione delle proprietà del suolo. Successivamente si è operata un'ulteriore scrematura per individuare i sensori con risoluzioni geometriche di acquisizione (dimensione del pixel a terra) adatte alla generazione di mappe ad alta ed altissima risoluzione.

Sono stati scelte quindi due piattaforme satellitari: **LANDSAT** missione 5 e 7 e **SPOT** missione 4 e 5. A bordo di questi satelliti sono installati infatti sensori che acquisiscono dati multi-spettrali adatti all'applicazione degli algoritmi di osservazione del suolo con la risoluzione geometrica richiesta.

Le immagini prodotte dai sensori di questi satelliti vengono archiviate seguendo standard differenti a seconda del tipo di sensore utilizzato e del tipo di formato di immagine prodotto. I sensori da cui vengono prese le immagini sono:

- Thematic Mapper (TM), [10], montato sul satellite **LANDSAT-5**, [8];
- Enhanced Thematic Mapper Plus (ETM+), [7], montato sul satellite **LANDSAT-7**, [9];

- High Resolution in the Visible and Infra-Red (HRVIR), [5], montato sul satellite **SPOT-4**, [2];
- High Resolution Geometric (HRG), [5], montato sul satellite **SPOT-5**, [2].

Ogni sensore permette di acquisire in modo multi-spettrale le immagini, ovvero acquisisce in diverse bande dello spettro elettromagnetico la stessa zona osservata. Le bande di acquisizione coprono diverse lunghezze d'onda: visibile (*VIS*), infrarosso vicino (*NIR*), infrarosso nelle onde corte (*SWIR*), infrarosso nelle onde medie (*MWIS*) e infrarosso termico (*TIR*). La risoluzione geometrica del pixel osservato dipende dalla lunghezza d'onda di osservazione, pertanto immagini relative a bande differenti possono avere diversa risoluzione: è il caso, ad esempio, dell'archivio Landsat Level 1G, in cui la risoluzione della banda termica risulta essere doppia (per sensore ETM+) o addirittura quadrupla (per sensore TM) rispetto alle altre bande. I sensori montati sui satelliti LANDSAT acquisiscono immagini ad *Alta* risoluzione (da 30 fino a 15 metri), mentre i sensori montati sui satelliti SPOT acquisiscono immagini ad *Altissima* risoluzione (da 20 fino 2,5 metri).

I valori del range di lunghezza d'onda e di risoluzione per ogni banda dei diversi sensori sono riportati nelle Tabelle 1.1 (TM, [10]), 1.2 (ETM+), 1.3 (HRVIR) e 1.4 (HRG).

| Banda | Lunghezza d'onda (μ m) | Risoluzione (m) |
|---------------|--|------------------------|
| Band 1 (VIS) | da 0.45 a 0.52 | 30 |
| Band 2 (VIS) | da 0.52 a 0.6 | 30 |
| Band 3 (VIS) | da 0.63 a 0.69 | 30 |
| Band 4 (NIR) | da 0.76 a 0.9 | 30 |
| Band 5 (SWIR) | da 1.55 a 1.75 | 30 |
| Band 6 (TIR) | da 10.4 a 12.5 | 120 |
| Band 7 (SWIR) | da 2.08 a 2.35 | 30 |

Tabella 1.1: Bande di acquisizione del sensore Thematic Mapper (TM)

| Banda | Lunghezza d'onda (μ m) | Risoluzione (m) |
|----------------|--|------------------------|
| Band 1 (VIS) | da 0.45 a 0.515 | 30 |
| Band 2 (VIS) | da 0.525 a 0.605 | 30 |
| Band 3 (VIS) | da 0.63 a 0.69 | 30 |
| Band 4 (NIR) | da 0.75 a 0.9 | 30 |
| Band 5 (SWIR) | da 1.55 a 1.75 | 30 |
| Band 6 (TIR) | da 10.4 a 12.5 | 60 |
| Band 7 (MWIR) | da 2.08 a 2.35 | 30 |
| Band PAN (VIS) | da 0.52 a 0.9 | 15 |

Tabella 1.2: Bande di acquisizione del sensore Enhanced Thematic Mapper Plus (ETM+)

| Banda | Lunghezza d'onda (μ m) | Risoluzione (m) |
|---------------|--|------------------------|
| Band 1 (VIS) | da 0.5 a 0.59 | 20 |
| Band 2 (VIS) | da 0.61 a 0.68 | 20 (10) |
| Band 3 (NIR) | da 0.78 a 0.89 | 20 |
| Band 4 (SWIR) | da 1.58 a 1.75 | 20 |

Tabella 1.3: Bande di acquisizione del sensore HRVIR

| Banda | Lunghezza d'onda (μ m) | Risoluzione (m) |
|----------------|--|------------------------|
| Band 1 (VIS) | da 0.5 a 0.59 | 10 |
| Band 2 (VIS) | da 0.61 a 0.68 | 10 |
| Band 3 (NIR) | da 0.79 a 0.89 | 10 |
| Band 4 (SWIR) | da 1.58 a 1.75 | 20 |
| Band PAN (VIS) | da 0.48 a 0.71 | 5 (2.5) |

Tabella 1.4: Bande di acquisizione del sensore HRG

1.3 Archivi dati satellitari

I dati satellitari relativi ad un determinato sensore possono presentare una varietà di modalità di archiviazione che dipende dall'ente distributore e dal livello di elaborazione del dato. Nell'ambito del progetto KEI, sono state definite limitazioni specifiche sulle tipologie di archivi di immagini satellitari a cui applicare gli algoritmi di processamento di immagini. Tutte le immagini contenute dentro gli archivi satellitari scelti sono immagini georeferenziate, che permettono di collocare geograficamente l'immagine sul globo terrestre. Questo permette di associare ogni singolo pixel, di cui

l'immagine è costituita, ad una precisa porzione di suolo terrestre.

Nei prossimi paragrafi verranno descritti i diversi tipi di archivi di immagini utilizzati.

1.3.1 Landsat Level 1G

Gli archivi Landsat Level 1G sono composti da un numero di file immagine pari al numero di bande acquisite dal sensore satellitare, e la nomenclatura dei file segue uno standard stabilito dal fornitore delle immagini satellitari. Nello specifico del progetto KEI lo standard di nomenclatura adottato è quello della Global Land Cover Facility (GLCF), che prevede la seguente struttura [13]:

`[satellite][sensor][path][startrow]_[endrow][date]_[band].[format][compression]`

La prima parte della struttura, chiamata *satellite*, identifica il satellite con cui sono state acquisite le immagini: ad esempio, la dicitura *L7* si riferisce al satellite *Landsat 7*. Il blocco successivo è un numero che indica il sensore utilizzato: ad esempio, *1* rappresenta il sensore *ETM+*. I successivi sei caratteri numerici a blocchi di tre identificano rispettivamente *path* e *row* nel sistema **WRS-2**, un sistema di georeferenziazione adottato per collocare le scene satellitari all'interno di un sistema di riferimento formato da riquadri. Così nel sistema **WRS-2** ogni coppia di valori *path* e *row* permette in modo molto semplice e rapido di risalire alla collocazione della scena sul globo terrestre. Un esempio del sistema **WRS-2** è rappresentato in figura 1.1 riferito allo stato della Thailandia.

Il campo successivo identificato dal nome *date* rappresenta la data in acquisizione dell'immagini satellitari espressa nel formato **YYYYMMDD**: ad esempio, **20000409** riferisce alla data 9 Aprile 2000.

Il blocco della struttura identificato come *band* rappresenta la banda di acquisizione a cui corrisponde l'immagine. Nel caso in cui il sensore utilizzato sia *ETM+* (vedi tabella 1.2) il blocco *band* è sempre espresso con due cifre questo per poter differenziare le due modalità di acquisizione della banda 6, in cui la sensibilità del sensore

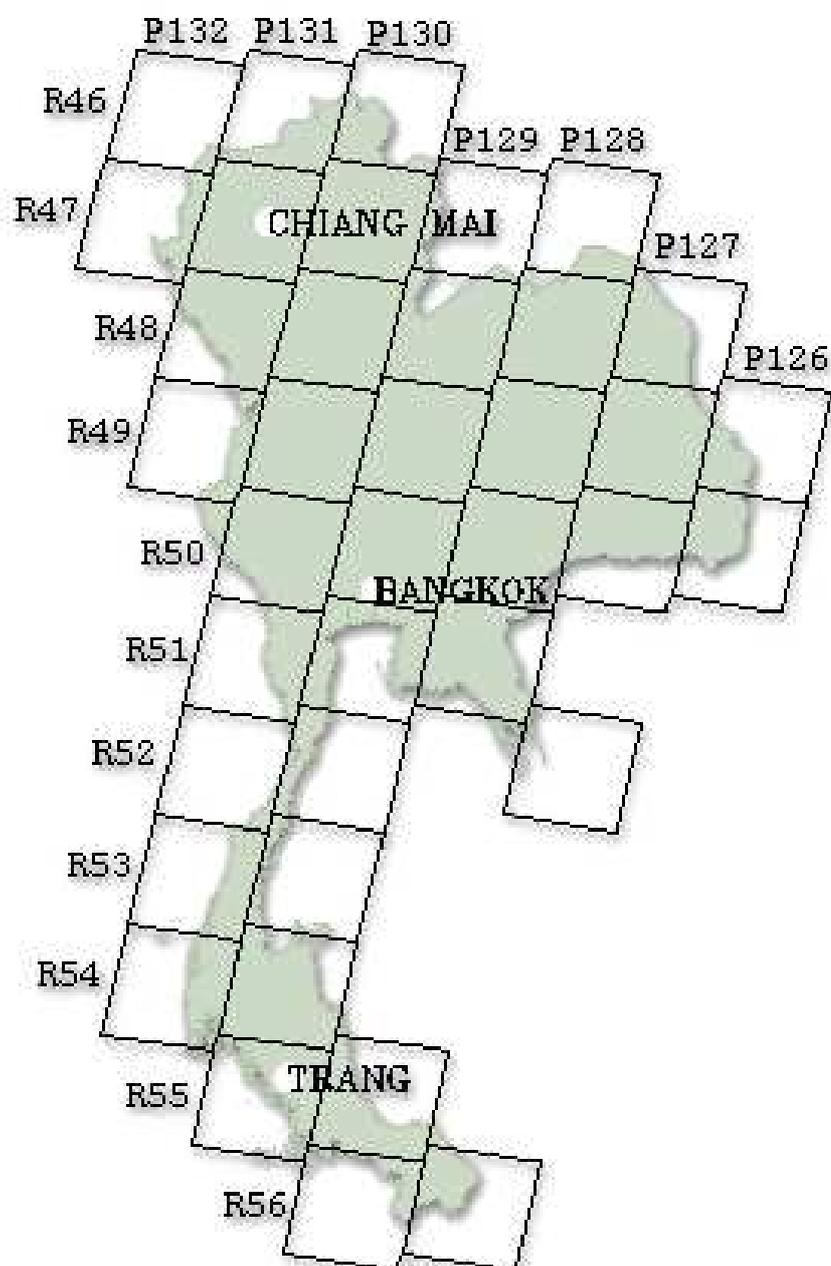


Figura 1.1: Sistema di georeferenziazione WRS-2 adottato per collocare le immagini satellitari sul globo terrestre utilizzando un sistema di riquadri identificati da una coppia di valori path e row. Esempio di sistema WRS-2 sullo stato della Thailandia.

presenta due valori di guadagno: così la banda 1 sarà indentificata con il numero 10, mentre per la banda 6 presenterà due immagini 61 e 62. Nel caso invece in cui il sensore di acquisizione sia **MSS** o **TM** (vedi tabella 1.1) viene utilizzato un solo carattere per esprimere la banda. La prima estensione del file immagine, ovvero il primo blocco dopo il punto indicato con l'etichetta *format*, indica il formato generato dal pre-processamento dell'immagine: ad esempio, l'estensione *.LIG* indica il formato *LIG*. L'estensione finale identificata dall'etichetta *compression* identifica invece la modalità di compressione del file immagine: ad esempio, *.gz* indica che è stato usato il software Gzip per comprimere il file.

Un caso particolare riguarda le scene acquisite usando il sensore **MSS** che adottano la nomenclatura dei file per *Landsat GeoCover collection* descritta nel capitolo successivo.

Un esempio di nomenclatura di un file in un archivio Landsat Level 1G è il seguente:

$$\underbrace{L7}_{\text{satellite}} \underbrace{1}_{\text{sensor}} \underbrace{015}_{\text{path}} \underbrace{033}_{\text{startrow}} \underbrace{033}_{\text{endrow}} \underbrace{20000409}_{\text{date}} \underbrace{B10}_{\text{band}} \underbrace{.L1G}_{\text{format}} \underbrace{.gz}_{\text{comp.}}$$

Le scene Landsat Level 1G sono accompagnate da un **metadata file** con estensione *.MTL* di tipo testuale in cui sono contenute informazioni aggiuntive alla scena osservata e informazioni riguardanti l'archivio stesso. Il *metadata file* ripropone alcune informazioni già contenute nella georeferenziazione dei GeoTIFF, ovvero il formato in cui sono memorizzate le immagini descritto nel paragrafo 1.4.1, e ne aggiunge altre utili per il processamento dell'immagine che il formato non permette di archiviare.

In risposta ai requisiti di standardizzazione dei servizi sviluppati durante il progetto KEI, l'utilizzo di immagini **Landsat Level 1G** tramite il portale **SSE** ha richiesto alcune semplici modifiche allo standard sopra citato. Nel formato prescelto le immagini dell'archivio in input ad un tipico servizio SSE non vengono compresse singolarmente, ma raccolte in un unico *Zip File* a cui viene aggiunto anche il **metadata file**. Questa scelta impone inoltre di dover omettere l'estensione nei nomi dei file in cui viene indicata la compressione utilizzata. Viene anche omessa nel nome l'estensione **format** e sostituita con l'estensione che identifica il tipo di file immagine, ovvero **.tif**.

1.3.2 Landsat GeoCover collection

Dal processamento delle immagini **Landsat Level 1G** viene costruita la collezione denominata *Landsat GeoCover*. Come per il *Landsat Level 1G* è stato creato dal fornitore, nello specifico sempre la GLCF, uno standard per la nomenclatura dei file contenuti nella collezione.

Il **Landsat GeoCover collection** è una collezione di immagini Landsat ortorettificate acquisite in tre differenti periodi di tempo: 1972-1980, 1990 e 2000, a cui corrispondono rispettivamente l'impiego dei sensori **MSS**, **TM** e **ETM+**. Ogni archivio **GeoCover collection** è composto da un numero di file pari al numero di bande che il sensore satellitare acquisisce. I file d'immagine contenuti all'interno degli archivi GeoCover collection seguono uno standard di nomenclatura così formato [13]:

$$[path][row]_{-}[sensor]_{-}[band].[format].[compression]}$$

La prima parte della struttura indica il numero di *path* e di *row* nel sistema **WRS-2**, come descritto nel paragrafo precedente. Per esempio una scena collocata dentro il riquadro che ha per riferimento path di 15 e row di 33, verrà indicato nel nome del file rispettivamente come *p015* e *r033*.

Il secondo blocco è costituito da un primo carattere che indica il satellite da cui è stata effettuata l'acquisizione, un secondo carattere che indica invece il sensore utilizzato seguendo (m = MSS, t = TM per piattaforma Landsat 5, o ETM+ per piattaforma Landsat 7). Il resto del blocco *sensor* indica la data di acquisizione espressa secondo il formato *YYYYMMDD* o *YYMMDD*. Ad alcuni file immagine viene aggiunto una parte che indica dove la scena è collocata rispetto al sistema di coordinate Universal Transverse Mercator (UTM). In modo del tutto analogo al sistema di georeferenziazione WRS-2, il sistema UTM consente di collocare velocemente la scena acquisita in una zona del globo terrestre: ad esempio se viene aggiunto **_z18** significa che l'immagine è localizzata nel sistema UTM nella zona 18.

La sezione *band* può essere preceduta dalla dicitura *nm* che indica se è stata adottata la tecnica di *resampling* (vedi [13]), mentre il numero indica la banda a cui l'immagine è associata. Il numero di banda è sempre espresso con due caratteri per poter

distinguere tra le due modalità di acquisizione della banda termica.

La parte del nome chiamata *format* identifica il formato dell'immagine. Ad esempio *.tif* indica che l'immagine è memorizzata nel formato GeoTIFF, mentre la parte del nome identificato dall'etichetta *compression* identifica come è stato compresso il file immagine. Ad esempio *.gz* identifica che è stato usato il software Gzip per comprimere il file. Un esempio di nomenclatura di un file in un archivio Landsat GeoCover è il seguente:

$$\begin{array}{ccccccc} \textit{path} & \textit{row} & \textit{sensor} & \textit{date} & \textit{band} & \textit{format} & \textit{compr.} \\ \hline \underbrace{p015r033}_{} & \underbrace{-}_{-} & \underbrace{5t}_{5t} & \underbrace{870516}_{870516} & \underbrace{nn10}_{nn10} & \underbrace{.tif}_{.tif} & \underbrace{.gz}_{.gz} \end{array}$$

Le scene Landsat GeoCover sono accompagnate da un **metadata file** con estensione *.met* di tipo testuale in cui sono contenute informazioni aggiuntive alla scena rappresentata e informazioni riguardanti l'archivio stesso. Il *metadata file* ripropone alcune informazioni già contenute nella georeferenziazione del formato GeoTIFF e ne aggiunge altre utili per il processamento dell'immagine che il formato non permette di archiviare.

Anche nel caso di archivi di immagini **Landsat GeoCover collection**, i requisiti di standardizzazione dei servizi SSE hanno richiesto semplici modifiche dello standard sopra citato. Le immagini dell'archivio non vengono compresse singolarmente, ma raccolte in un unico *Zip File* in cui viene anche inserito in **metadata file**. Questo impone inoltre di dover omettere l'estensione nei nomi dei file in cui viene indicata la compressione utilizzata.

1.3.3 Landsat CEOS

L'archivio di tipo **Landsat Committee on Earth Observation Satellites (CEOS)** è costituito da immagini nel formato **CEOS**, descritto in dettaglio nel paragrafo 1.4.2. Ogni archivio è composto da un numero variabile di sotto directory (una o più) che raccolgono i dataset di diverse immagini. Ogni sottodirectory relativa ad una scena contiene a sua volta un numero di dataset pari al numero di bande acquisite dal sensore del satellite (TM vedi tabella 1.1, ETM+ vedi tabella 1.2). Ogni file all'interno del-

l'archivio ha un suffisso di due caratteri numerici che identifica la banda, ad esempio il suffisso *_01* corrisponde alla banda *1*.

1.3.4 SPOT DIMAP

DIGital IMage maP (DIMAP), [1], è un formato di archivi di immagini provenienti da satelliti **SPOT** a cominciare dalla missione 5 lanciata nel Maggio del 2002. Questo tipo di archivio è stato sviluppato dal Centre National d'Etudes Spatiales (CNES), ovvero l'Agenzia Spaziale Francese.

Un archivio DIMAP è composto essenzialmente di due parti:

- **Image:** un file immagine nel formato GeoTIFF multi-livello che contiene l'immagine vera e propria. Ogni banda acquisita viene memorizzata come un livello dell'immagine.
- **Metadata:** un file nel formato eXtensible Markup Language (XML) in cui sono raccolte tutte le informazioni utili al processamento dell'immagine (informazioni di georeferenziazione, parametri di calibrazione radiometrica, parametri di acquisizione dell'immagine, ecc.).

1.4 Formati di immagine

1.4.1 Formato TIFF/GeoTIFF

Il formato Tagged Image File Format (TIFF) è un formato di immagine di tipo raster composto sostanzialmente da una parte chiamata Image File Directory (IFD), in cui sono archiviate tutte le informazioni e le caratteristiche dell'immagine (numero di righe, numero di colonne, tipo di dato, profondità cromatica, livello di compressione, ecc.) e da una parte di *dati* in cui è contenuta l'informazione per ogni pixel dell'immagine. Le specifiche del formato TIFF permettono un'estrema flessibilità che richiede l'utilizzo di riferimenti standard posti all'inizio del file (*header*) per poter reperire le informazioni contenute nella parte IFD e gestire la parte dedicata ai *dati*. Uno schema

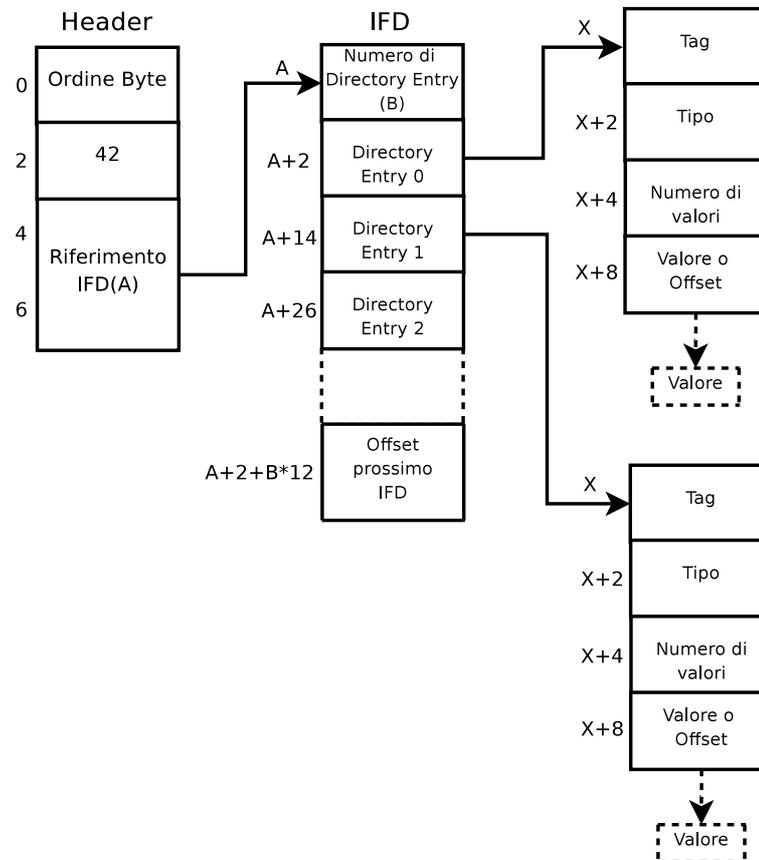


Figura 1.2: Organizzazione delle strutture di Header e di IFD. Il valore alfanumerico indicato a sinistra di ogni blocco fa riferimento alla posizione espressa in byte all'interno del file.

riassuntivo di come sono costruiti i riferimenti all'interno di un TIFF file è rappresentato in figura 1.2. Il carattere alfanumerico a sinistra dei blocchi rappresenta l'offset, ovvero la posizione all'interno del file espressa in numero di byte in cui si trova il valore del parametro.

L'header collocato all'inizio del file TIFF, oltre all'informazione sulla formattazione del file (Ordine Byte) ed al codice di identificazione del formato TIFF (42), contiene il riferimento del primo IFD (Riferimento IFD (A)). I primi 2 byte indicano quante informazioni sono contenute nell'IFD (Numero di Directory Entry(B)), a cui seguono i blocchi contigui (Directory Entry 0, 1, 2, ...), ciascuno composto di 12 byte. Ogni Directory Entry e' composto da due parti: una parte relativa all'informazione sul valore del dato (TAG, Tipo, Numero di Valori) ed una parte in cui risiede il valore del dato

vero e proprio o il riferimento per individuarlo all'interno del file (Valore o Offset).

Il formato d'immagine **GeoTIFF** eredita tutti i meccanismi implementati dalle **TIFF** per la rappresentazione dell'immagine, garantendo così la compatibilità del formato con i vari software di visualizzazione. Inoltre il formato GeoTIFF aggiunge al contenuto di informazioni standard (numero di righe, numero di colonne, tipo di dato, profondità cromatica, livello di compressione, ecc.) le informazioni sulla georeferenziazione dell'immagine come latitudine, longitudine, proiezione, ecc..

1.4.2 Formato CEOS

Il formato di immagine CEOS, [4], è stato sviluppato da un'ampia associazione di agenzie spaziali[12]. Questo formato di immagine si prefigge lo scopo di uniformare e standardizzare la rappresentazione di un dataset di immagini satellitari multi-banda. Ogni immagine CEOS è formata da tre file, denominati **Imagery File (DAT)**, **Leader File (LEA)** e **Trailer File (TRA)**. I tre file, sebbene contengano informazioni differenti riguardanti la stessa immagine, presentano la medesima formattazione, facilitando l'implementazione di strumenti per accedere al contenuto di ciascun file.

L'**Imagery File** contiene la parte dei dati relativa ai valori di ogni pixel di cui è costituita l'immagine. Essendo il formato **CEOS** un formato di tipo raster, l'informazione è rappresentata come in una matrice.

Il **Leader File** contiene le informazioni relative alla georeferenziazione e tutte le informazioni relative all'acquisizione dell'immagine, informazioni utili per una successiva manipolazione. Sostanzialmente il **Leader File** è composto da 4 parti:

- **File Descriptor record:** parametri relativi ai descrittori del formato.
- **Scheme Header Record:** parametri relativi alla scena, alla missione, al sensore utilizzato e valori utili per il processamento dell'immagine.
- **Map Projection Ancillary Record:** parametri riguardanti le caratteristiche geometriche dei dati grezzi di input e dell'immagine processata.
- **Radiometric Ancillary Record:** parametri e tabelle di trasformazione radiometrica.

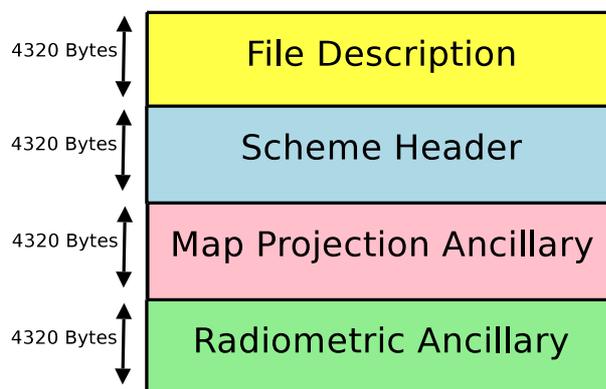


Figura 1.3: Schema dei record di un'immagine CEOS nel Leader File.

Il **Trailer File** contiene tutte le informazioni aggiuntive ed accessorie riguardanti l'immagine, come ad esempio la qualità dei dati e la qualità dell'acquisizione.

Per estrarre le informazioni dai vari file basta posizionarsi all'offset dove è localizzata l'informazione e leggere una quantità di byte pari alla grandezza del dato selezionato. La posizione all'interno del file e la dimensione di ogni dato è standardizzata dal formato **CEOS**. Nello schema in figura 1.3 è rappresentato il Leader File con i suoi quattro blocchi sopra descritti.

Come si può vedere nello schema la dimensione dei blocchi è sempre di 4320 bytes e i blocchi sono posti in modo consecutivo. Per reperire le informazioni in ogni blocco bisogna tenere conto dell'offset dell'informazione all'interno del blocco a cui va sommata la dimensione dei blocchi precedenti. Una descrizione dettagliata della struttura di file CEOS e di tutte le informazioni contenute può essere consultata in [4].

Capitolo 2

Software I/O

2.1 Software/librerie di I/O

Lo scopo di questa tesi riguarda lo sviluppo di applicativi per la manipolazione e il processamento di immagini satellitari multispettrali in ambiente Linux.

All'interno del progetto KEI, il formato immagine di riferimento per manipolare o processare immagini è il formato **GeoTIFF**. Durante lo sviluppo del software si è cercato quindi di implementare strumenti che facilitassero l'interazione con questo tipo di immagine. Quando possibile, ogni formato immagine in ingresso ai vari programmi sviluppati è stato fatto convergere al formato GeoTIFF creando opportuni software di conversione.

Dopo una prima idea di creare ex-novo strumenti per la manipolazione di immagini GeoTIFF si è pensato di rivolgersi a strumenti di interfaccia già sviluppati e consolidati negli anni, in modo da poter disporre già da subito di strumenti testati e affidabili. Le librerie utilizzate per la gestione delle immagini nel formato GeoTIFF utilizzate sono **libtiff**, [6], e **libgeotiff**, [3]. Per la gestione di altri formati d'immagine invece si sono usate le librerie **GDAL**, [11], che comunque danno anche la possibilità di interagire con le immagini TIFF e GeoTIFF.

2.2 Librerie disponibili LANDSAT e SPOT

2.2.1 Librerie libtiff

Le librerie **libtiff** forniscono delle interfacce astratte per la gestione di immagini in formato TIFF multi-livello. Ogni livello dell'immagine può essere letto usando campioni di 8 bit senza tenere conto di come sia costruita l'immagine TIFF internamente, ovvero vengono resi trasparenti all'utilizzatore i meccanismi e i controlli legati al colorspace o la compressione dei dati. Le librerie TIFF permettono, attraverso una serie di Application Programming Interface (API), di acquisire i valori dei dati grezzi relativi ai valori dei pixel dell'immagine, in modo da facilitare la creazione di strumenti per la manipolazioni dell'immagini TIFF. Le principali funzioni per interfacciarsi alle libtiff sono le seguenti:

- **TIFFSetWarningHandler()**: Imposta il livello di warning comunicati durante l'uso delle librerie.
- **XTIFFOpen()**: Apre il file GeoTIFF.
- **TIFFGetField()**: Legge un tag TIFF associato da una determinata chiave.
- **TIFFSetField()**: Scrive un tag TIFF associato da una determinata chiave.
- **TIFFNumberOfStrips()**: Ritorna il numero di strip¹ da cui è formata l'immagine TIFF.
- **_TIFFmalloc()**: Ritorna il puntatore dell'area allocata come buffer di lettura dell'immagine TIFF.
- **TIFFReadEncodedStrip()**: Legge una determinata strip.
- **TIFFWriteEncodedStrip()**: Scrive una determinata strip.
- **_TIFFfree()**: Libera la memoria occupata dal buffer di lettura.

¹Una **STRIP** è un elemento del formato TIFF in cui sono memorizzate i valori dei pixel di una o più righe dell'immagine.

- **TIFFStripSize():** Ritorna la grandezza in byte della strip.
- **TIFFClose():** Chiude il file TIFF.

2.2.2 Librerie libgeotiff

Le librerie **libgeotiff** forniscono un'interfaccia per accedere ai tag di georeferenziazione contenuti nelle immagini GeoTIFF. L'utilizzo delle librerie libgeotiff si sovrappone e si affianca all'utilizzo delle librerie **libtiff** per la manipolazione delle immagine satellitari. Infatti le librerie libgeotiff forniscono solo gli strumenti per accedere alle informazioni contenute nei tag GeoTIFF, ma non hanno strumenti né per l'accesso ai tag TIFF, né per la manipolazione dei valori associati ai pixel dell'immagine. Le principali funzioni per interfacciarsi alle **libgeotiff** sono le seguenti:

- **XTIFFOpen():** Apre il file GeoTIFF.
- **GTIFNew():** Ritorna il puntatore alla struttura dati contenente i parametri di georeferenziazione.
- **GTIFKeyGet():** Legge un tag GeoTIFF associato da un determinato parametro di georeferenziazione come ad esempio la zona UTM.
- **GTIFKeySet():** Scrive un tag GeoTIFF associato da un determinato parametro di georeferenziazione.
- **XTIFFClose():** Chiude il file GeoTIFF.
- **GTIFFree():** Libera la memoria occupata dal puntatore alla georeferenziazione.

2.2.3 Geospatial Data Abstraction Library (GDAL)

Le librerie **Geospatial Data Abstraction Library (GDAL)**, [11], forniscono un modello di dati astratto per accedere a numerosi formati geografici *raster*².

²Per **raster** si intendono i formati di immagine in cui le informazioni dei pixel vengono memorizzate per linee e colonne, come se fosse una matrice.

Le librerie GDAL offrono strumenti per leggere, scrivere, manipolare le immagini ed i parametri di georeferenziazione rendendo trasparente dal punto di vista dell'utente l'uso di diversi formati di immagine, dando così la possibilità di sviluppare semplicemente applicativi compatibili con tutti i formati supportati. In figura 2.1 è rappresentato lo schema di funzionamento delle librerie GDAL. Come si può notare dallo schema è presente un layer, ovvero uno strato software di intercomunicazione tra i vari componenti della libreria, chiamato **Driver** deputato all'input/output dell'immagine. Questo layer riferisce ad un **Dataset** collocato in memoria comune per tutti i formati di immagine sia in input che in output. Nello schema è presente anche un altro componente chiamato **Spatial Reference** che comunica con il **Dataset** per la manipolazione delle informazioni di georeferenziazione in essa contenute.

Le funzioni più utilizzate per interfacciarsi alle librerie **GDAL** sono le seguenti:

- **GDALAllRegister():** Inizializza l'environment per l'utilizzo delle librerie.
- **GDALGetDriverByName():** Restituisce il driver per la gestione di un determinato formato.
- **GDALGetDatasetDriver():** Restituisce il driver utilizzato per aprire un determinato Dataset.
- **GDALCreateCopy():** Crea una copia di un'immagine dando la possibilità di utilizzare un driver differente da quello usato per aprirla. In pratica viene eseguita una conversione di formato con il trasferimento anche delle informazioni di georeferenziazione e proiezione.
- **GDALSetGeoTransform():** Imposta la matrice di georeferenziazione in un Dataset.
- **GDALSetProjection():** Imposta le informazioni di proiezione in un Dataset.
- **GDALRasterIO():** Legge o scrive un Dataset.
- **GDALCreate():** Crea un Dataset vuoto associato ad un Driver.
- **GDALOpen():** Apre un'immagine.

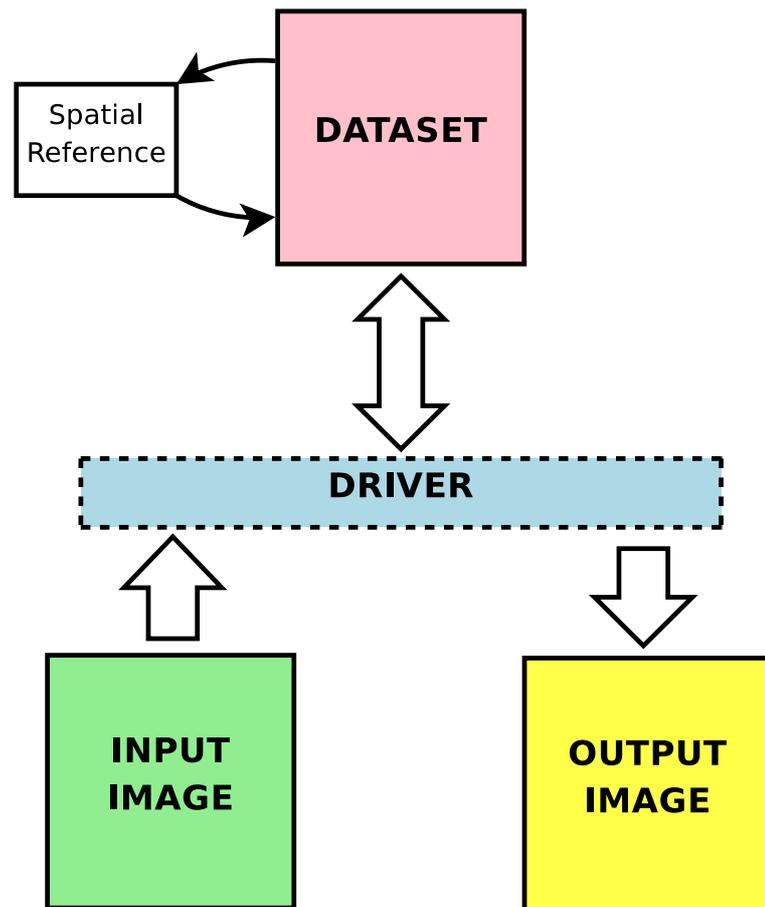


Figura 2.1: Schema del funzionamento delle librerie GDAL

- **GDALClose():** Chiude un'immagine.
- **OSRNewSpatialReference():** Crea un oggetto di tipo SpatialReference per la manipolazione delle informazioni di referenziazione.
- **OSRImportFromWkt():** Estrae dal Dataset le informazioni di referenziazione per poterle manipolare.
- **OSRExportToWkt():** Esporta le informazioni di referenziazione manipolate per poterle impostare nel Dataset.

2.3 Software design e prestazioni

Nello sviluppo del software per la manipolazione delle immagini si è cercato di ottimizzare le prestazioni per ridurre il più possibile i tempi di elaborazione. Da un'analisi dei vari step che i programmi devono affrontare durante l'elaborazione, si è potuto notare che le fasi che incidono di più nel conteggio totale del tempo di elaborazione sono due: la fase di I/O verso l'hardisk per l'acquisizione dell'immagine e la successiva riscrittura, e la fase di parsing³ di ogni elemento della matrice che rappresenta l'immagine.

La fase di I/O verso l'hardisk rappresenta un collo di bottiglia nelle comunicazione interne di un computer, questo perché il reperimento delle informazioni non avviene in modo diretto, come ad esempio verso la RAM, ma deve seguire tre fasi:

- **Seek time:** tempo necessario a spostare la testina sulla traccia;
- **Latency time:** tempo necessario a posizionare il settore desiderato sotto la testina;
- **Transfer time:** tempo necessario al settore per passare sotto la testina.

Quindi si è scelto di implementare il software in modo che la fase di lettura sia unica per minimizzare il tempo delle fasi di Seek e Latency, copiando temporaneamente

³Per **parsing** si intende il processo atto ad analizzare un flusso continuo di dati.

tutte le informazioni nella RAM che offre dei tempi di accesso notevolmente inferiori rispetto l'hardisk.

Il parsing degli elementi della matrice che rappresenta l'immagine, dal punto di vista procedurale, ha una complessità computazionale quadratica, ovvero, nel contesto di una elaborazione su di un computer monoprocesso il tempo di elaborazione aumenterebbe in modo quadratico con l'aumentare della dimensione dei dati da elaborare. Potendo disporre di computer multiprocesso per la manipolazione di immagini si è quindi deciso di sfruttare la possibilità di suddividere l'elaborazione totale in sotto elaborazioni simultanee ed indipendenti, affidandole ognuna ad un processore e solo alla fine del computo ricongiungere i risultati parziali ottenuti per ricostituire il risultato finale. Questa metodica permette di ridurre il tempo di elaborazione di un fattore proporzionale al numero di CPU di cui il computer dispone.

Capitolo 3

Il portale SSE, servizi di elaborazione ed integrazione

Il portale web SSE è un portale sviluppato e mantenuto da European Space Agency (ESA) per fornire ad un qualsiasi utente finale una serie di strumenti per la manipolazione e il processamento di immagini satellitari, garantendo interoperabilità tra servizi sviluppati da aziende differenti che intendono pubblicare i propri prodotti/servizi sul portale.

La compatibilità di tutti gli strumenti disponibili all'interno del portale SSE è assicurata dall'implementazione di metodi standard per la comunicazione tra i vari servizi e per l'accesso ai dati satellitari supportati.

Ogni servizio all'interno del portale SSE prevede come input un archivio con una struttura standardizzata a seconda del tipo di dati contenuti nell'archivio, e restituisce come output un archivio con le immagini processate, anch'esso conforme al medesimo standard. Questa compatibilità tra gli archivi di input ed output consente di utilizzare i vari servizi in una catena di lavoro e quindi di poter usufruire in modo sequenziale di diverse elaborazioni.

3.1 Servizi KEI

I servizi KEI sono stati sviluppati per processare solo determinati archivi di immagini al cui interno vi siano solo determinati formati di immagine. Le combinazioni di tipologie di archivi con i formati d'immagine sono i seguenti:

- Archivio **CEOS** con immagini **CEOS** da piattaforma **Landsat-5**
- Archivio **CEOS** con immagini **CEOS** da piattaforma **Landsat-7**
- Archivio **GeoCover** con immagini **GeoTIFF** da piattaforma **Landsat-7**
- Archivio **Level 1G** con immagini **GeoTIFF** da piattaforma **Landsat-7**

I servizi di base implementati sono cinque:

- **Format Check**
- **Format Conversion**
- **Clipping**
- **Landsat Image Radiometric Conversion and Correction (LIRCC)**
- **Landsat Image Spectral Categorization (LISC)**

e sono descritti in dettaglio nelle sezioni successive.

3.1.1 Format Check

Il servizio di Format Check rappresenta lo strumento principale di controllo per l'accesso agli altri servizi o strumenti implementati. Questo strumento controlla infatti se l'archivio di immagini satellitari fornito in input è conforme allo standard stabilito come input ai vari servizi. Al Format Check vengono fornite in ingresso l'informazione del formato (*inputFormat*) che dovrebbero avere le immagini contenute nell'archivio e la piattaforma (*platform*), ovvero da quale satellite sono state acquisite le immagini. Questo servizio controlla la conformità dell'archivio in input ai seguenti standard:

- **CEOS Landsat-5**
- **CEOS Landsat-7**
- **GeoTIFF Landsat-7**

Terminata l'elaborazione il servizio restituisce un messaggio dove informa l'utente se l'archivio di immagini è conforme alle informazioni specificate come input; in caso di esito negativo viene specificato in quale parametro l'archivio non è conforme all'input fornito per il test.

In figura 3.1 è rappresentato il diagramma di funzionamento del servizio di Format Check, in cui a seconda degli input forniti vengono seguiti cammini differenti per il test del pacchetto. Dallo schema si vede una prima differenziazione tra i formati GeoTIFF e CEOS: nel caso il formato sia GeoTIFF non vi sono altre alternative perché lo script accetta solo la piattaforma Landsat-7, mentre per il formato CEOS vi è un'ulteriore ramificazione per differenziare le piattaforme Landsat-5 e Landsat-7.

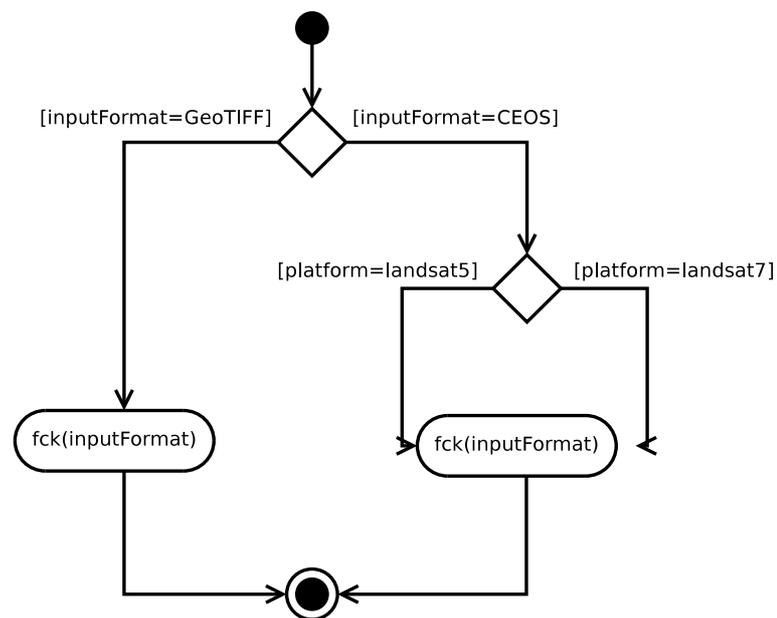


Figura 3.1: Diagramma del servizio Format Check.

Il riconoscimento del formato dell'immagini contenute dentro l'archivio è stato implementato in un programma chiamato **fck** che, data un'immagine di input, ne riconosce il formato. Il programma **fck** essendo implementato con le librerie GDAL riconosce tutti i formati supportati dalle librerie stesse.

Per eseguire il servizio di Format Check sull'immagine in input, il programma **fck** è basato sulla chiamata alla funzione *GDALOpen()* che fallisce (restituendo NULL, e stampando a monitor un messaggio di errore) nel caso in cui il file di input non sia un file immagine riconosciuto. Diversamente, nel caso in cui il contenuto nella variabile *hDataset* sia diverso da NULL, e venga quindi riconosciuto un formato di immagine valido, l'esecuzione procede con la chiamata alla funzione *GDALGetDatasetDriver* che restituisce il riferimento al *Driver* (memorizzato nella variabile *hDriver*) associato al dataset in fase di elaborazione. Usando il Driver dell'immagine si può richiamare la funzione *GDALGetDriverLongName* che permette di stampare il nome del formato dell'immagine. Nelle seguenti linee di codice è riportato come viene fatto il controllo:

```
hDataset = GDALOpen( argv [ i ], GA_ReadOnly );
if( hDataset != NULL ){
    hDriver = GDALGetDatasetDriver( hDataset );
    printf ( "%s : %s\n" , argv [ i ], GDALGetDriverLongName (
        hDriver ) );

} else{
    printf ( "%s : Not image\n" , argv [ i ] );
}
```

3.1.2 Format Conversion

Il Format Conversion è lo strumento preposto alla conversione del formato delle immagini contenute in un archivio in un altro formato. Naturalmente oltre alla conversione del file immagine bisogna generare o convertire tutte le informazioni contenute dentro i metadata file.

Le opzioni di conversione contemplate dal servizio sono le seguenti:

- da CEOS a:
 - GeoTIFF
 - JPEG2000
 - JPEG

- da GeoTIFF a:
 - JPEG2000
 - JPEG

Le immagini GeoTIFF mantengono le informazioni di calibrazione radiometrica contenute nel file originale. I formati GeoTIFF e JPEG2000 mantengono le informazioni di georeferenziazione contenute nel file originale. Agli archivi JPEG e JPEG2000 è associato un file di quicklook, ovvero un file immagine in formato JPEG, realizzato combinando le bande *rosso*, *verde* e *blu* per creare un'immagine in colori reali della scena convertita. Nelle operazioni di conversione da formato CEOS, gli archivi risultanti (indipendentemente dal formato di uscita) conterranno un singolo file per ogni banda.

Per la realizzazione di questo servizio sono stati creati sotto programmi che svolgono specifici compiti durante le operazioni di conversione. I programmi con la loro funzione sono di seguito elencati:

- **ceos2geotiff**: Converte le immagini di tipo di CEOS a immagini di tipo GeoTIFF
- **fcv**: Converte le immagini di tipo GeoTIFF a immagini JPEG2000 e JPEG
- **metgenerator5**: Genera il metafile partendo da CEOS Landsat-5 per un archivio GeoCover
- **metgenerator7**: Genera il metafile partendo da CEOS Landsat-7 per un archivio GeoCover

- **rgbquicklook:** Crea per gli archivi JPEG e JPEG2000 una quicklook per la visualizzazione a colori della scena.

Il programma che esegue la conversione vera e propria dell'immagine raster in input è **fcv**. Questo programma si basa sulla chiamata a funzione *GDALCreateCopy()*, che, dato in input il Driver del formato di immagine che si vuole in output, converte automaticamente l'immagine. Nelle seguenti linee di codice è riportata come viene utilizzata la funzione di conversione.

```

hDriver = GDALGetDriverByName( form );
if ( (hSrcDS = GDALOpen( file_in , GA_ReadOnly )) == NULL ){
    printf( "Unable to open source file %s\n" , file_in );
    exit(1);
}
hDstDS = GDALCreateCopy( hDriver , file_out , hSrcDS , FALSE,
    papszCreateOptions , NULL, NULL);
if( hDstDS == NULL ){
    printf( "Unable to create %s file %s\n" , form , file_out );
    exit(1);
}

```

La variabile **form** contiene il nome del driver che si vuole creare, in **hSrcDS** il dataset in input e in **papszCreateOptions** le opzioni di conversione.

Le informazioni in input al servizio devono essere il formato (*inputFormat*), la piattaforma (*platform*) e il formato di uscita (*outputFormat*). In figura 3.2 è rappresentato il diagramma di funzionamento di questo servizio in cui a seconda degli input forniti vengono seguiti cammini differenti per la conversione del pacchetto. Nel diagramma vi è una prima ramificazione in cui viene diversificato il cammino di processamento a seconda del formato, CEOS o GeoTIFF. Nel caso il formato di input sia GeoTIFF vi è solo un'altra ramificazione in cui si differenzia il formato di output, ovvero JPEG o JPEG2000. Nel caso, invece, il formato di input sia CEOS vi è una prima ramificazione tra le due piattaforme, Landsat-5 o Landsat-7, per differenziare come effettuare la conversione a GeoTIFF. Successivamente nel caso il formato di output sia Geo-

TIFF, il processamento termina, mentre se il formato di output è JPEG o JPEG2000 vi un'ulteriore ramificazione per scegliere tra questi dei due formati.

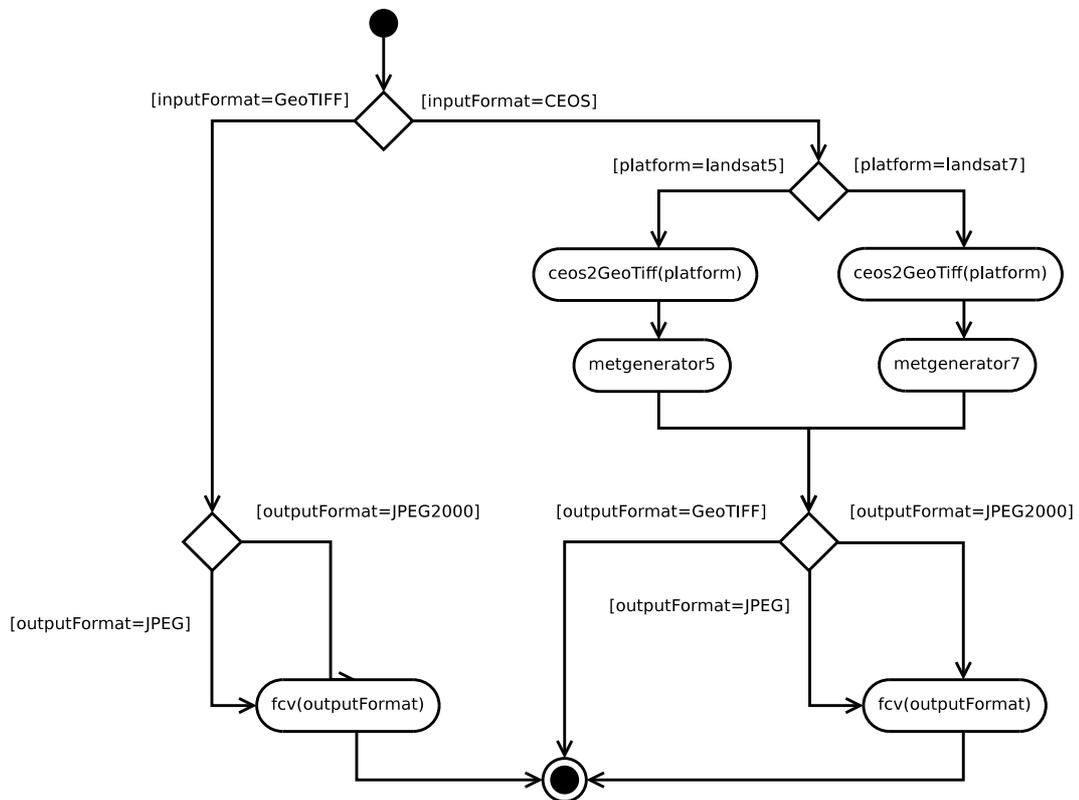


Figura 3.2: Diagramma del servizio Format Conversion.

3.1.3 Clipping

Il servizio di Clipping dà la possibilità di creare sotto scene delle immagini contenute nell'archivio di input specificando le coordinate geografiche degli angoli in basso a sinistra (Low Left (LL)) ed in alto a destra (Upper Right (UR)) del box della sotto scena. Il servizio di Clipping accetta in input dati CEOS e GeoTIFF e produce in output dati in formato GeoTIFF. Le coordinate passate come input allo script devono descrivere un box di taglio rettangolare in coordinate geografiche espresse in latitudine e longitudine dei due punti.

Le immagini contenute all'interno di un archivio possono essere ruotate rispetto al sistema di coordinate geografiche N-S/E-W: dovendo eseguire un taglio rettangolare dell'immagine occorre pertanto determinare il più piccolo sovra-box nel sistema N-S/E-W che contenga il box di taglio in modo da avere comunque la zona selezionata all'interno della sotto-scena. In figura 3.3 si può vedere un esempio di taglio di una sotto-scena nel caso di immagine ruotata.

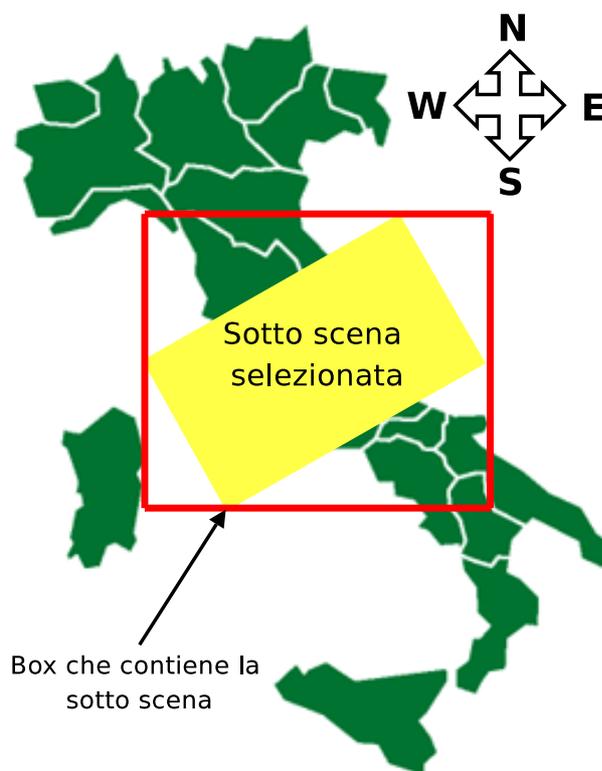


Figura 3.3: Esempio di taglio di un'immagine ruotata.

L'area in giallo rappresenta la sotto-scena selezionata nel sistema di riferimento dell'immagine che risulta ruotato rispetto al nord geografico; prima di procedere al taglio viene pertanto creato il sovra-box (bordo rosso in figura 3.3) allineato al nord geografico e contenente la sotto-scena.

Un esempio pratico del servizio di Clipping è mostrato nelle figure 3.4 - 3.5: la prima immagine (ottenuta come combinazione RGB delle bande 3,2,1) rappresenta la zona attorno al lago di **Salton Sea** nel deserto del Colorado (California-USA). Nella

medesima figura è evidenziato con un rettangolo verde il box di taglio dell'immagine di coordinate $LLlat = 32.7360, LLlon = -116.4184$ e $URlat = 33.7465, URlon = -115.0828$. La seconda figura 3.5 mostra invece la combinazione RGB del risultato del servizio di Clipping.

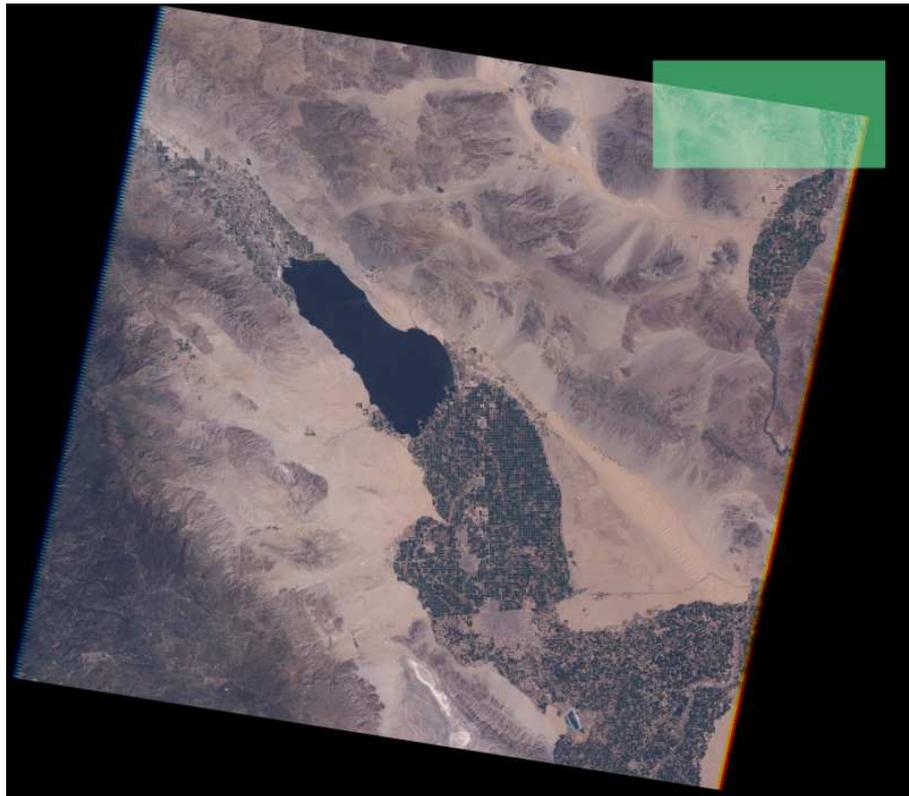


Figura 3.4: Archivio Landsat Level 1G (L71039037_03720030325.zip): combinazione RGB delle bande 3,2,1. Immagine acquisita dal sensore ETM+ il 25 Marzo 2003.

Per la realizzazione di questo servizio sono stati creati sotto programmi che svolgono operazioni specifiche durante l'esecuzione del taglio della sotto scena. Alcuni componenti sono comuni al servizio di Format Conversion per poter convertire il formato dell'archivio di input nel formato compatibile al programma che realizza effettivamente il taglio. I programmi che compongono il servizio sono di seguito elencati:

- **ceos2geotiff**: Converte le immagini di tipo di CEOS a immagini di tipo GeoTIFF



Figura 3.5: Archivio Landsat Level 1G (L71039037_03720030325-ss.zip): combinazione RGB delle bande 3,2,1. Immagine processata dal servizio di Clipping.

- **metgenerator5:** Genera il metadata file partendo da CEOS Landsat-5 per un archivio GeoCover
- **metgenerator7:** Genera il metadata file partendo da CEOS Landsat-7 per un archivio GeoCover
- **clipgtiff:** Crea la sotto-scena di un'immagine GeoTIFF
- **Metconverter:** Converte il metadata file di un archivio GeoCover, adattando le informazioni alla sotto-scena selezionata.

Il programma che realizza il taglio vero e proprio della sotto-scena è **clipgtiff**. Questo programma accetta in ingresso le coordinate espresse in latitudine e longitudine dell'angolo in basso a sinistra e in alto a destra del box di taglio dell'immagine. Partendo dalle coordinate locali si ricavano le coordinate espresse in pixel corrispondenti nell'immagine utilizzando l'inversa della matrice di conversione C . La matrice di conversione C consente di eseguire la trasformazione:

$$\begin{aligned} \begin{bmatrix} x' \\ y' \end{bmatrix} &= C \begin{bmatrix} n_x \\ n_y \end{bmatrix} \\ &= pxSize \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} n_x \\ n_y \end{bmatrix} \end{aligned}$$

dove x' e y' sono le coordinate locali in UTM, n_x e n_y sono le coordinate di partenza espresse in numero di pixel, $pxSize$ è la risoluzione geometrica del pixel e θ è l'angolo di rotazione dell'immagine.

A partire dalla matrice di conversione C il calcolo dell'inversa viene effettuato utilizzando il metodo standard per matrici 2×2 :

$$\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \rightarrow \begin{pmatrix} \frac{C_{22}}{C_{11}C_{22}-C_{12}C_{21}} & \frac{-C_{12}}{C_{11}C_{22}-C_{12}C_{21}} \\ \frac{-C_{21}}{C_{11}C_{22}-C_{12}C_{21}} & \frac{C_{11}}{C_{11}C_{22}-C_{12}C_{21}} \end{pmatrix}$$

Di seguito riportato un estratto di codice del programma **clipgtiff** riportato in appendice A.1.1 dove viene calcolata la matrice inversa:

```
matrix_inv[0] =  adfGeoTransform[5]/( adfGeoTransform[1] *
    adfGeoTransform[5] - adfGeoTransform[2]* adfGeoTransform
    [4]);
matrix_inv[1] = -adfGeoTransform[2]/( adfGeoTransform[1] *
    adfGeoTransform[5] - adfGeoTransform[2]* adfGeoTransform
    [4]);
matrix_inv[2] = -adfGeoTransform[4]/( adfGeoTransform[1] *
    adfGeoTransform[5] - adfGeoTransform[2]* adfGeoTransform
    [4]);
matrix_inv[3] =  adfGeoTransform[1]/( adfGeoTransform[1] *
    adfGeoTransform[5] - adfGeoTransform[2]* adfGeoTransform
    [4]);
```

L'array di float **matrix_inv** contiene i quattro valori della matrice inversa, mentre l'array **adfGeoTransform** contiene sei valori float che corrispondono ai seguenti parametri:

- **adfGeoTransform[0]:** Coordinata X espressa in UTM dell'angolo in alto a sinistra dell'immagine.
- **adfGeoTransform[1]:** Elemento C_{11} della matrice di conversione.
- **adfGeoTransform[2]:** Elemento C_{12} della matrice di conversione.

- **adfGeoTransform[3]:** Coordinata Y espressa in UTM dell'angolo in alto a sinistra dell'immagine.
- **adfGeoTransform[4]:** Elemento C_{21} della matrice di conversione.
- **adfGeoTransform[5]:** Elemento C_{22} della matrice di conversione.

Il calcolo per ricavare le coordinate espresse in pixel è riportato nelle seguenti linee di codice:

```

pxLR[0] = (int)(( matrix_inv[2] * (LL[2] - adfGeoTransform[3]) )
    + ( matrix_inv[0] * (UR[3] - adfGeoTransform[0]) ));
pxLR[1] = (int)(( matrix_inv[3] * (LL[2] - adfGeoTransform[3]) )
    + ( matrix_inv[1] * (UR[3] - adfGeoTransform[0]) )) * -1;
pxLL[0] = (int)(( matrix_inv[2] * (LL[2] - adfGeoTransform[3]) )
    + ( matrix_inv[0] * (LL[3] - adfGeoTransform[0]) ));
pxLL[1] = (int)(( matrix_inv[3] * (LL[2] - adfGeoTransform[3]) )
    + ( matrix_inv[1] * (LL[3] - adfGeoTransform[0]) )) * -1;
pxUR[0] = (int)(( matrix_inv[2] * (UR[2] - adfGeoTransform[3]) )
    + ( matrix_inv[0] * (UR[3] - adfGeoTransform[0]) ));
pxUR[1] = (int)(( matrix_inv[3] * (UR[2] - adfGeoTransform[3]) )
    + ( matrix_inv[1] * (UR[3] - adfGeoTransform[0]) )) * -1;
pxUL[0] = (int)(( matrix_inv[2] * (UR[2] - adfGeoTransform[3]) )
    + ( matrix_inv[0] * (LL[3] - adfGeoTransform[0]) ));
pxUL[1] = (int)(( matrix_inv[3] * (UR[2] - adfGeoTransform[3]) )
    + ( matrix_inv[1] * (LL[3] - adfGeoTransform[0]) )) * -1;

```

Le coordinate in pixel vengono memorizzate in array di due elementi dove il primo elemento è la coordinata X e il secondo elemento è la coordinata Y, mentre gli array LL e UR contengono le coordinate espresse in UTM degli angoli, rispettivamente, in basso a sinistra ed in alto a destra del box di taglio.

Nell'implementazione del programma **clipgtiff** si è dovuto risolvere il problema in cui l'area di taglio selezionata dall'utente non rientra completamente all'interno dell'immagine. Per far questo si sono dovuti implementare una serie di controlli sulle

coordinate di taglio. Nel caso gli angoli del box di taglio siano esterni all'immagine vengono riportati sui bordi dell'immagine, proiettando la coordinate sui vari lati della scena. In pratica si individua la zona di taglio come il risultato dell'intersezione del piano che contiene l'immagine e il piano che contiene l'area selezionata dall'utente.

Nel caso in cui la scena di taglio sia completamente esterna all'immagine viene comunicato un messaggio di errore. Se il box di taglio comprende tutta l'immagine viene restituita l'immagine stessa. I controlli applicati sono riportati nelle seguenti linee di codice:

```
if( ( BoxLL[0] > pxSizeX ) || ( BoxLL[1] > pxSizeY ) ){
    printf("Invalid coords of subscene\n");
    exit(2);
}
if( ( BoxUR[0] < 0 ) || ( BoxUR[1] < 0 ) ){
    printf("Invalid coords of subscene\n");
    exit(2);
}
if( BoxLL[0] < 0 ) BoxLL[0] = 0;
if( BoxLL[1] < 0 ) BoxLL[1] = 0;
if( BoxUR[0] > (pxSizeX - 1) ) BoxUR[0] = pxSizeX - 1;
if( BoxUR[1] > (pxSizeY - 1) ) BoxUR[1] = pxSizeY - 1;
```

Le variabili `pxSizeX` e `pxSizeY` contengono i valori delle dimensioni dell'immagine in pixel, mentre le variabili `BoxLL` e `BoxUR` contengono le coordinate del box di taglio.

Le informazioni in input al servizio devono essere il formato (*inputFormat*), la piattaforma (*platform*), le coordinate espresse in latitudine e longitudine dell'angolo in basso a sinistra e dell'angolo in alto a destra della sotto-scena selezionata. In figura 3.6 è rappresentato il diagramma di funzionamento di questo servizio in cui a seconda degli input forniti vengono seguiti cammini differenti per il processamento dell'archivio. Dal diagramma si vede come il percorso di processamento dell'immagine da tagliare converga al formato di immagine GeoTIFF, ovvero l'unico formato accettato dall'eseguibile *clipgtiff*. Infatti nel caso il formato di input sia GeoTIFF il

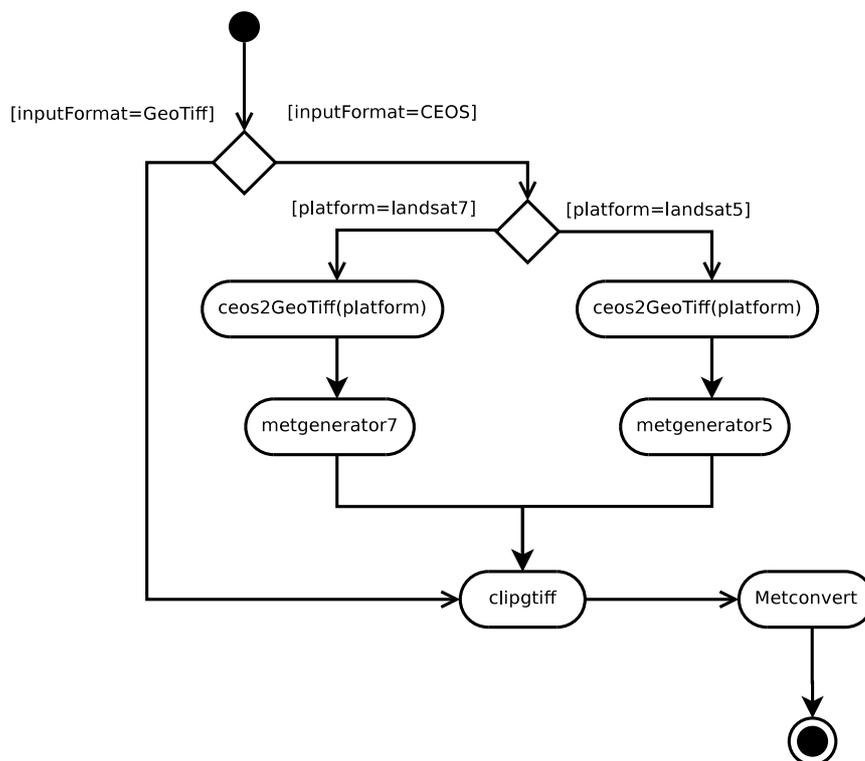


Figura 3.6: Diagramma del servizio di Clipping.

diagramma porta direttamente al *clipgtiff*, mentre se il formato di input è CEOS, vi è una prima differenziazione della piattaforma Landsat-5 o Landsat-7 per la conversione dell'archivio e poi viene eseguito il programma per il taglio *clipgtiff*.

Una prima versione del servizio di Clipping è stata realizzata in linguaggio Integrated Numerical Analysis & Data Visualization Solution (IDL). Questa prima soluzione è stata però abbandonata perché non coerente con la linea di sviluppo del software per i servizi. Infatti usando IDL ci si doveva appoggiare per l'esecuzione del servizio sul motore interno di questo software e questo limitava le performance e la flessibilità. In oltre per lo sviluppo dei servizi non si voleva essere legati a software proprietari e quindi si è deciso di sviluppare il Clipping usando il linguaggio di programmazione C.

3.1.4 LIRCC

Il servizio chiamato LIRCC esegue la conversione e correzione radiometrica (di seguito indicata con il termine calibrazione) sulle immagini contenute nell'archivio di input. I dati accettati in ingresso possono essere in formato CEOS oppure GeoTIFF; l'archivio di dati in uscita è formato da file in formato GeoTIFF floating point (un file per ogni banda). L'operazione di calibrazione utilizza i parametri contenuti all'interno dei metafile per trasformare il valore numerico relativo alle singole bande acquisite, ovvero Digital Number (DN), in valori che esprimono le grandezze fisiche associate da cui è possibile operare una successiva elaborazione dell'immagine. In particolare le bande nello spettro dal Visible (VIS) al Short-wavelength InfraRed (SWIR) vengono convertite in riflettanza al Top Of Atmosphere (TOA), mentre le bande nel Thermal InfraRed (TIR) vengono convertite in Brightness Temperature (BT) in gradi K.

Per la realizzazione di questo servizio sono stati creati sotto programmi che svolgono compiti specifici durante le operazioni di calibrazione dell'immagine. Alcuni componenti sono comuni al servizio di Format Conversion per poter convertire il formato dell'archivio di input nel formato compatibile al programma che realizza la calibrazione delle immagini. I programmi che compongono il servizio sono di seguito elencati:

- **ceos2geotiff:** Converte le immagini di tipo CEOS a immagini di tipo GeoTIFF
- **metgenerator5:** Genera il metadata file partendo da CEOS Landsat-5 per un archivio GeoCover;
- **metgenerator7:** Genera il metadata file partendo da CEOS Landsat-7 per un archivio GeoCover;
- **ln5rcc:** Calibra le immagini provenienti da un archivio Landsat-5;
- **ln7rcc:** Calibra le immagini provenienti da un archivio Landsat-7.

Le informazioni in input al servizio devono essere il formato (*inputFormat*) e la piattaforma (*platform*), tutte le informazioni aggiuntive necessarie alla calibrazione vengono estratte dal metafile contenuto nell'archivio o generato nel caso l'archivio sia

di tipo CEOS. In figura 3.7 è rappresentato il diagramma di funzionamento di questo servizio in cui a seconda degli input forniti vengono seguiti cammini differenti per il processamento dell'archivio. Nel diagramma si nota che vi è una prima ramificazione

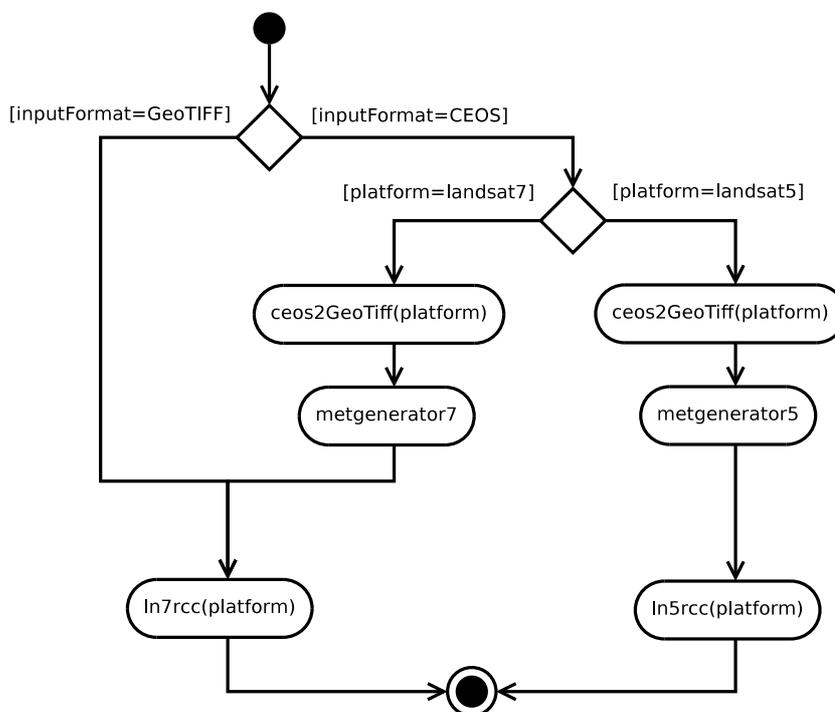


Figura 3.7: Diagramma del servizio di LIRCC.

per differenziare i due tipi di formato, GeoTIFF o CEOS. Nel caso il formato sia di tipo GeoTIFF la piattaforma può essere solo Landsat-7 e quindi le immagini vengono passate direttamente al programma *ln7rcc*. Nel caso invece il formato di input sia CEOS il percorso di processamento ha un'ulteriore ramificazione per distinguere le due possibili piattaforme, Landsat-5 o Landsat-7 e a seconda della piattaforma viene convertito l'archivio di immagini a GeoTIFF (unico formato accettato dai programmi che fanno la calibrazione) e poi successivamente passato al rispettivo programma di calibrazione, ovvero *ln5rcc* per i Landsat-5 e *ln7rcc* per i Landsat-7. Il dato elaborato viene archiviato come floating point in quanto rappresenta un campo continuo, a differenza dell'immagine originale che è composta da valori discreti di conteggi del sensore.

3.1.5 LISC

Il servizio LISC esegue una *Preliminary Spectral Categorisation* (di seguito indicata come Classificazione) di immagini precedentemente calibrate con il servizio LIRCC. Il servizio richiede quindi in input un archivio in formato GeoTIFF-floating point, e genera in uscita una singola immagine in formato GeoTIFF. Attraverso l'analisi dell'informazione spettrale delle singole bande e di parametri da esse derivate, il servizio LISC consente di generare mappe con diverse classi (fino a 72) di tipologia di suolo (diverse classi di vegetazione, suolo scoperto, area urbana, nubi, acqua, ghiaccio, ecc.). Il servizio permette di selezionare tra 3 diversi livelli di classificazione. Ogni livello di classificazione ha un numero di classi relative al tipo di terreno osservato che dipende dal sensore satellitare impiegato: maggiore sarà l'informazione spettrale (numero di bande) maggiore sarà il numero di classi potenzialmente ottenibili in output dall'elaborazione. I livelli di classificazione per i sensori montati su piattaforme Landsat sono i seguenti:

- **Limited (Small):** Classificazione con 15 classi.
- **Reduced (Intermediate):** Classificazione con 38 classi.
- **Full (Large):** Classificazione con 72 classi.

La terminologia indicata tra parentesi fa riferimento ai termini utilizzati all'interno del portale SSE. Per la realizzazione di questo servizio sono stati creati due programmi che svolgono diversi compiti durante le operazioni di classificazione dell'immagine. I programmi che compongono il servizio sono di seguito elencati:

- **preIsc:** Prepara le immagini delle varie bande per la classificazione scalando i valori della grandezza fisiche tra valori compresi tra 0 (valore minimo) e 255 (valore massimo). Oltre a questo riporta alla stessa risoluzione tutte le immagini.
- **lisc:** Esegue la classificazione vera e propria.

L'unica informazione necessaria in input al servizio è il livello di classificazione (*Type of Classification*). In figura 3.8 è rappresentato il diagramma di funzionamento di questo servizio.

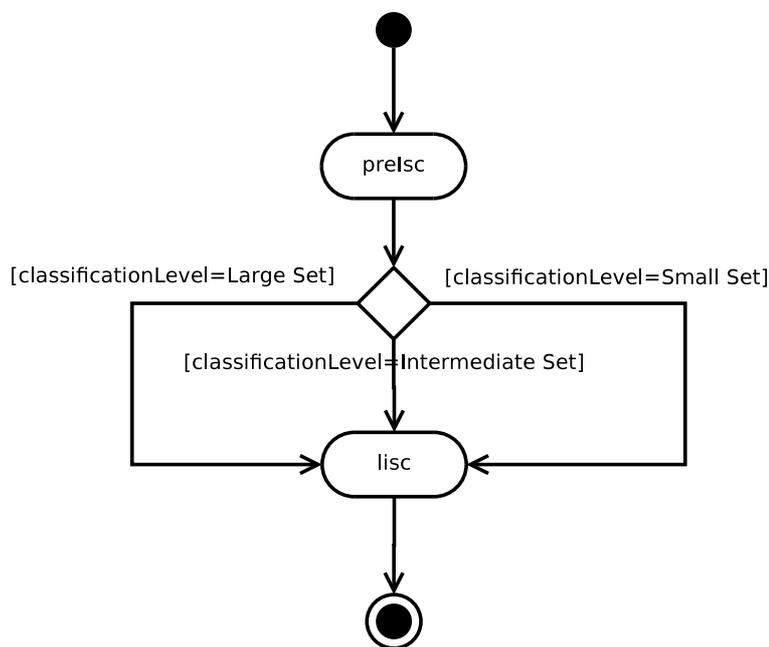


Figura 3.8: Diagramma del servizio di LISC.

Dal diagramma si vede come non vi sono strade di processamento differenti a seconda del formato o della piattaforma, questo perché il servizio accetta solo archivi di immagini prodotti dal servizio di calibrazione LIRCC. Vi è un primo step obbligato per il *preIsc* e successivamente una ramificazione per distinguere le tre tipologie di classificazione che comunque convergono sempre all'eseguibile *lisc*.

La figura 3.10 mostra il risultato in output al servizio di Classificazione applicato all'immagine sulla zona est della Scozia (vedi combinazione RGB di figura 3.9). Nel caso specifico è stata eseguita una classificazione con valore del parametro *Type of Classification* = Full. Dal confronto delle due immagini si può notare come l'algoritmo classifichi in modo automatico le diverse zone di vegetazione (diverse gradazioni di verde), le zone di suolo scoperto (diverse gradazioni di marrone), corpi idrici, nubi, ecc..



Figura 3.9: Archivio Landsat CEOS (LN5-cal.zip): combinazione RGB delle bande 3,2,1. Immagine acquisita dal sensore TM il 25 Marzo 2003.

3.2 Script di esecuzione dei servizi

I vari componenti che compongono ogni servizio sviluppato sono gestiti tramite l'uso di shell script in Bash. Ogni script è implementato seguendo uno schema standard composto essenzialmente di cinque fasi.

1. Controlli sugli argomenti passati come parametri allo script.
2. Scomattamento dell'archivio dato all'interno di una directory temporanea creata dentro la directory di lavoro.
3. Elaborazione o manipolazioni delle immagini contenute dentro l'archivio.
4. Creazione di un nuovo archivio di immagini con il risultato dell'elaborazione o manipolazione.



Figura 3.10: Archivio Landsat CEOS (LN5-cal-SpCI72.zip): Immagine processata dal servizio di Classificazione.

5. Pulizia della directory di lavoro con l'eliminazione di file temporanei e rimozione della directory temporanea.

I parametri che di default ogni script accetta sono i seguenti:

[workDir][zipFile][platform][format]

- **workDir:** Directory di lavoro in cui di deve trovare il file archivio e dove il servizio compie il suo lavoro.
- **zipFile:** Il nome del file archivio di tipo Zip in cui sono tenute le immagini.
- **platform:** La piattaforma dell'archivio, ovvero da in che missione Landsat sono state acquisite, può essere landsat5 o landsat7.

- **format:** Il formato delle immagini contenute nell'archivio, può essere CEOS o GeoTIFF.

In aggiunta a questi parametri di default devono essere specificati, a secondo del servizio, i parametri specifici per l'esecuzione. Nella seguente lista vengono riportati gli argomenti aggiuntivi per ogni servizio:

- **formatCheck:** Nessun parametro aggiuntivo.
- **formatConversion:** *outputFormat*, formato di immagine dell'archivio in output, può essere GeoTIFF, JPEG o JPEG2000.
- **Clipping:** *LLlon,LLlat URLon,URLat*, coordinate espresse in latitudine e longitudine dell'angolo in basso a sinistra e in alto a destra del box di taglio dell'immagine.
- **LIRCC:** Nessun parametro aggiuntivo.
- **LISC:** *Type Of Classification*, tipo di classificazione, può essere Full, Reduced o Limited.

3.3 Gestione degli errori

Nella creazione dei servizi sono stati standardizzati i vari codici di errore che gli script producono, in modo da poter ricondurre un certo codice di errore sempre alla stessa problematica durante l'elaborazione. Ogni script sviluppato nel caso di errore restituisce i seguenti codici:

- **1:** Numero di parametri di input insufficienti per l'esecuzione.
- **2:** Coordinate in input non valide.
- **3:** Errore nel parametro del formato di immagine.
- **5:** Directory di lavoro non valida.
- **6:** Zip file inesistente.

- **7:** Invalido formato del Zip file.
- **8:** Errore nel parametro della piattaforma.
- **10:** Numero dei LEA file all'interno di un CEOS incoerente.
- **11:** Errore del formato del Metadata file all'interno di un archivio di immagini GeoTIFF.
- **12:** Errore di elaborazione durante la creazione di un file JPEG.
- **13:** Errore di elaborazione durante la conversione di una immagine CEOS ad una GeoTIFF.
- **14:** Invalido formato di un immagine GeoTIFF.
- **18:** File immagine corrotto.

Nel caso di errore generico viene invece restituito il codice di errore **19**.

Capitolo 4

Servizi concatenati

4.1 Analisi delle possibili catene di processamento

Nelle varie fasi di processamento di un'immagine spesso vi è la necessità di usufruire di diversi servizi in modo sequenziale. Ad esempio, da un dataset di immagini può essere richiesto di ricavarne una sotto-scena, calibrarla radiometricamente e quindi eseguirne una classificazione. Da questa necessità di processamento sono state create due catene di elaborazione, in modo da rendere disponibili due servizi aggiuntivi che consentono di concatenare diverse elaborazioni senza dover fare per ogni step dell'elaborazione una richiesta al portale SSE. I servizi concatenati per come sono stati progettati (standardizzazione di input ed output) permettono l'esecuzione consecutiva dei singoli servizi di cui sono costituiti. Infatti, come prima soluzione è stata ipotizzata la semplice concatenazione dei vari script, dove l'archivio di uscita ad un determinato step della catena di processamento costituiva l'archivio di input del servizio successivo. Si è notato però che il tempo di decompressione dell'archivio di input e la successiva creazione dell'archivio di output occupava quasi la totalità del tempo di elaborazione. Si è scelto pertanto di sostituire l'approccio sequenziale dei singoli servizi con la creazione di catene di lavoro in cui l'archivio di input è scompattato solo all'inizio della catena e l'archivio di output è creato solo alla fine della catena. Sono state create due catene di lavoro i cui servizi sono denominati **Reconditioning** e **Categorization**.

4.2 Reconditioning

Il servizio di reconditioning crea una catena di lavoro usando i servizi **Clipping** e **Format Converion** già implementati. Le informazioni in input al servizio devono essere: quali servizi attivare nella catena (*services*), il formato (*inputFormat*), la piattaforma (*platform*), le coordinate espresse in latitudine e longitudine dell'angolo in basso a sinistra e dell'angolo in alto a destra della sotto-scena selezionata (*LLlon,LLlat URLon,URLat*), e il formato di uscita (*outputFormat*).

Per decidere quali servizi attivare nell'esecuzione della catena di lavoro, viene passato allo script di Reconditioning un primo parametro che deve essere formato da due caratteri. Il primo carattere identifica il servizio di Clipping e il secondo quello di Format Converion. I caratteri possono essere 0 o 1 e il servizio viene attivato se il carattere è 1.

In figura 4.1 è rappresentato il diagramma di funzionamento di questo servizio in cui a seconda degli input forniti vengono seguiti cammini differenti per il processamento dell'archivio.

Dal diagramma si vede che l'elaborazione passa attraverso una fase obbligatoria in cui vengono disabilitate varie funzionalità degli script dei servizi, esplicitate nel paragrafo 4.4, e successivamente a seconda di quali servizi sono stati abilitati nella catena vengono percorse via di processamento differenti.

4.3 Categorization

Il servizio di Categorization crea un catena di lavoro usando i servizi **Clipping**, **LIRCC** e **LISC**.

Le informazioni in input al servizio devono essere: quali servizi attivare nella catena (*services*), il formato (*inputFormat*), la piattaforma (*platform*), le coordinate espresse in latitudine e longitudine dell'angolo in basso a sinistra e dell'angolo in alto a destra della sotto-scena selezionata (*LLlon,LLlat URLon,URLat*), e il livello di classificazione (*Type Of Classification*). Per decidere quali servizi attivare nell'esecuzione della catena di lavoro, viene passato allo script di Categorization un primo parametro che deve

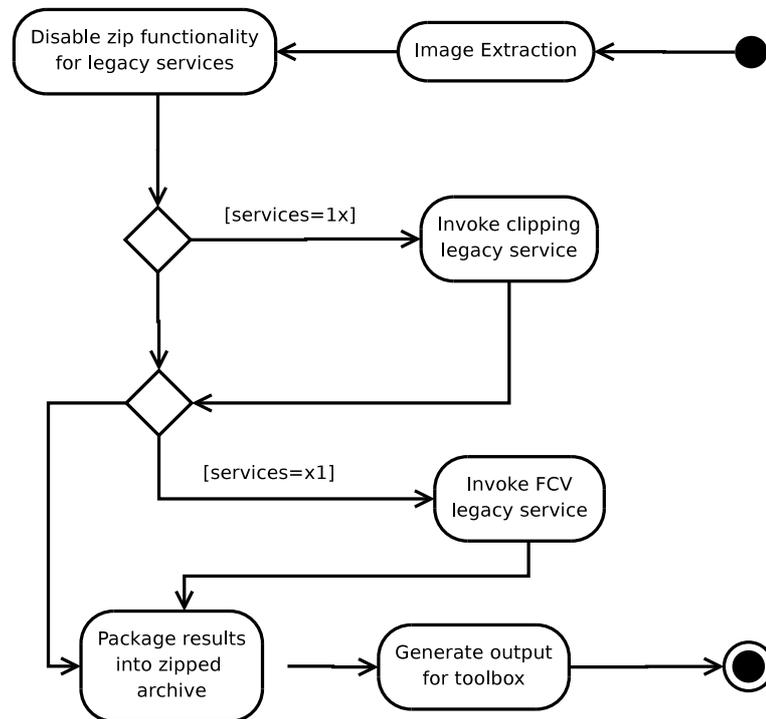


Figura 4.1: Diagramma del servizio di Reconditioning

essere formato da tre caratteri. Il primo carattere identifica il servizio di clipping, il secondo quello di LIRCC e il terzo quello di LISC. I caratteri possono essere 0 o 1 e il servizio viene attivato se il carattere è 1.

In figura 4.2 è rappresentato il diagramma di funzionamento di questo servizio in cui a seconda degli input forniti vengono seguiti cammini differenti per il processamento dell'archivio.

Un esempio di come la catena di lavoro sia gestita all'interno del servizio di Categorization è visibile in figura 4.3, in cui viene dato in input al servizio il dato grezzo e all'interno della catena viene eseguito il Clipping, la calibrazione radiometrica e infine viene dato in output l'immagine classificata.

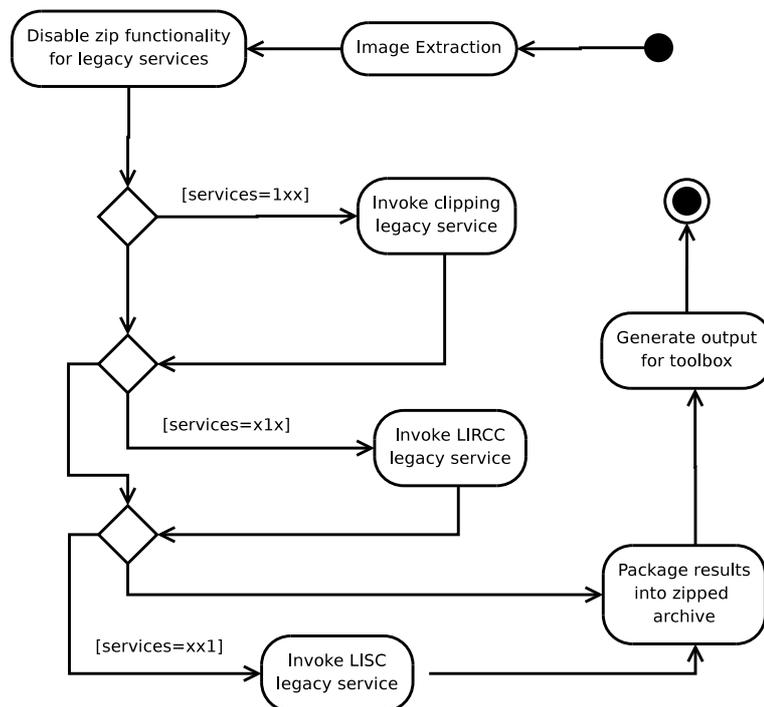


Figura 4.2: Diagramma del servizio di Categorization.

4.4 Modifiche software e script

Durante l'esecuzione degli script dei servizi, le fasi che occupano più tempo sono quella di scompattamento dell'archivio all'inizio del processamento e quella di creazione dell'archivio di output al termine dell'elaborazione. La creazione di una catena di servizi, dove per ogni servizio vengono richiamate queste due fasi farebbe inutilmente aumentare il tempo di processamento. Quindi si è deciso di creare uno script che crei un environment ad hoc per l'esecuzione dei servizi concatenati in modo da poter disabilitare le fasi di scompattamento e creazione dell'archivio finale durante gli step intermedi dell'esecuzione della catena e lasciare queste fasi solo all'inizio ed al termine del processamento. Lo script che richiama i vari servizi si chiama **fullchain.sh**. Questo script oltre a scompattare l'archivio di immagini all'inizio e creare l'archivio di output al termine dell'elaborazione, deve disabilitare o limitare il funzionamento di alcuni comandi di shell richiamati all'interno degli script in modo da disabilitare le fasi di scompattamento e creazione dell'archivio finale interne per ogni singolo servizio. I

comandi di shell disabilitati dallo script **fullchain.sh** sono i seguenti:

- **rm**: Cancellazione di file o directory disabilitato.
- **rmdir**: Cancellazione di directory disabilitato.
- **exit**: Vengono intercettati dei codici di uscita degli script in modo da verificare se vi sono stati problemi durante l'esecuzione.
- **unzip**: Scompattamento di un archivio Zip disabilitato.
- **zip**: Creazione dell'archivio di uscita disabilitato.
- **mkdir**: Creazione di directory disabilitato.
- **mv**: Spostamento di file o directory modificato in modo che se fallisce ritorna comunque un codice di successo.

Disabilitando e/o limitando tutti i comandi sopra elencati, alla fine di ogni step della catena lo script `fullchian.sh` deve preoccuparsi di ricreare gli input corretti per il servizio successivo, come se si fosse appena scompattato l'archivio di immagini. Per poter fare questo, lo script `fullchian.sh`, nell'esecuzione di un servizio concatenato tiene traccia dei nuovi file creati durante il singolo step, in modo da poter riconoscere quali siano i file prodotti dall'elaborazione del servizio richiamato. Una volta creata la lista dei file nuovi e confrontata con una lista di file creata prima di richiamare il servizio, lo script è in grado di cancellare i file vecchi e mantenere quelli nuovi. In questo modo allo step successivo il servizio concatenato si troverà solamente i file prodotti dallo step precedente. Nell'appendice A.2 è riportato il codice completo dello script `fullchian.sh`.

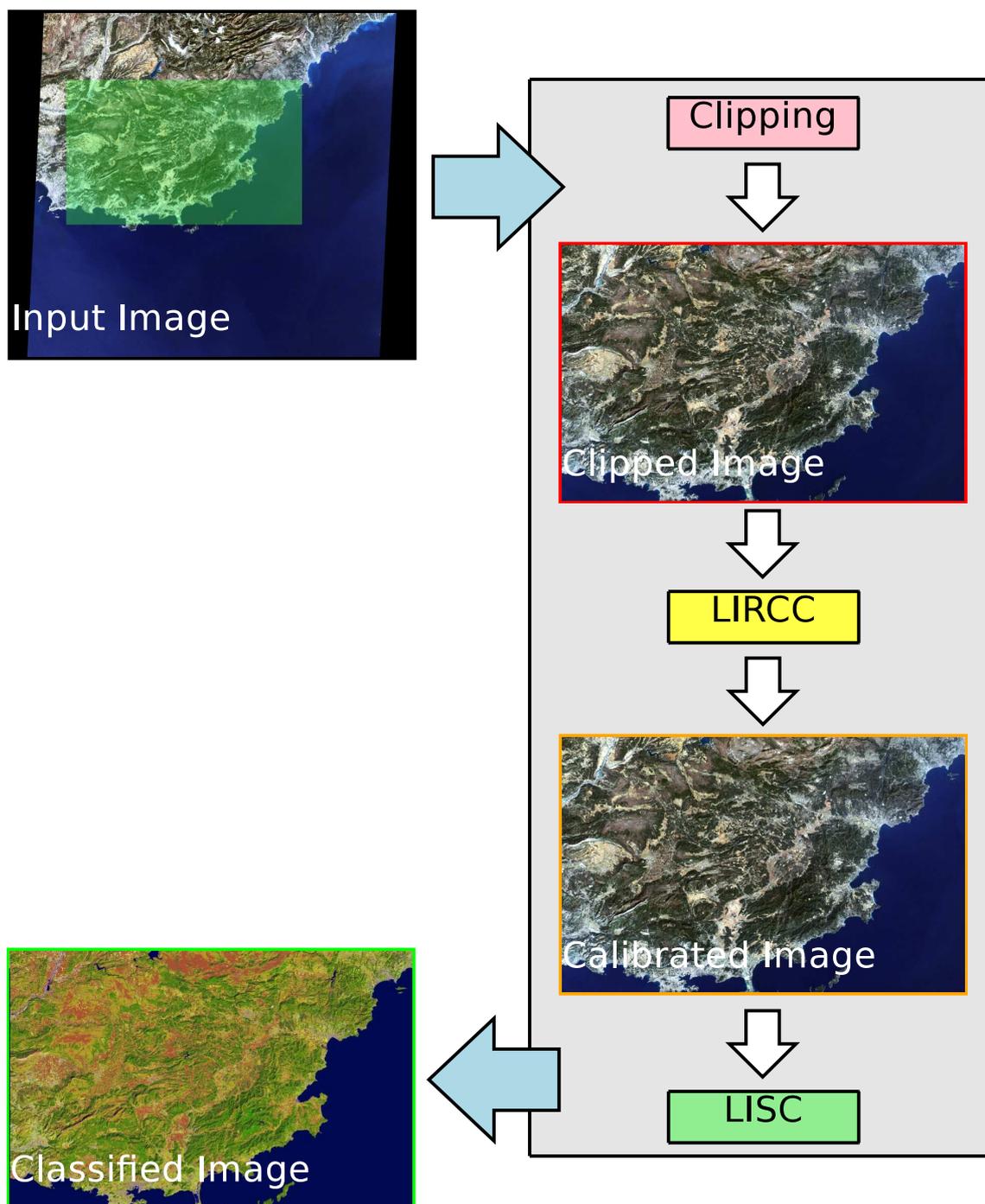


Figura 4.3: Schema della catena di processamento completa del servizio di Categorization.

Capitolo 5

Test di verifica sui servizi

5.1 Overview dei test eseguiti

Terminata la fase di sviluppo del software sono stati eseguiti test per verificarne il corretto funzionamento prima della consegna e dell'integrazione su portale SSE in modalità operativa. In accordo con gli standard European Cooperation For Space Standardization (ECSS) e con il documento di Verification and Validation Test Plan (VVTP), [14], le procedure di verifica e validazione del software implementato prevedono diversi livelli di test:

- Unit Test (UT), eseguiti direttamente sulle macchine di sviluppo per verificare il corretto funzionamento dei pacchetti software implementati (eseguibili e script Bash);
- Integration Test (IT), eseguiti per verificare la corretta integrazione dei servizi sul portale SSE;
- Acceptance Test (AT), eseguiti per dimostrare la conformità ed il rispetto di tutti i requisiti funzionali dei servizi.

In seguito alla consegna del software è stata pianificata una giornata di test presso i laboratori dell'Agenzia durante la quale sono stati effettuati gli AT in presenza dei responsabili ESA del progetto KEI. In Tabella 5.1 sono riportati tutti i test eseguiti.

Ciascun test è univocamente identificato da un codice (prima colonna) composto da tre campi (separati dal carattere '_'): il nome del servizio, la sigla AT, e un numero progressivo. La seconda colonna della tabella dà una breve descrizione del test effettuato specificando gli input passati al servizio. La terza colonna riporta il tempo necessario per il completamento del test, mentre nella quarta colonna è indicato l'esito (Passato/Fallito).

| Test ID | Descrizione | Durata (min) | Passato/Fallito |
|-----------|---|--------------|-----------------|
| CLP_AT_01 | Clipping archivio Landsat 5 CEOS | 10 | Passato |
| CLP_AT_02 | Clipping archivio Landsat 7 CEOS | 10 | Passato |
| CLP_AT_03 | Clipping archivio Landsat 7 Level 1G | 10 | Passato |
| CLP_AT_04 | Clipping archivio Landsat 7 GeoCover | 10 | Passato |
| FCV_AT_05 | Format Conversion da Landsat 5 CEOS a GeoTIFF | 5 | Passato |
| FCV_AT_06 | Format Conversion da Landsat 5 CEOS a Jpeg2000 | 12 | Passato |
| FCV_AT_07 | Format Conversion da Landsat 5 CEOS a Jpeg | 5 | Passato |
| FCV_AT_08 | Format Conversion da Landsat 7 CEOS a GeoTIFF | 5 | Passato |
| FCV_AT_09 | Format Conversion da Landsat 7 CEOS a Jpeg2000 | 12 | Passato |
| FCV_AT_10 | Format Conversion da Landsat 7 CEOS a Jpeg | 5 | Passato |
| FCV_AT_11 | Format Conversion da Landsat 7 Level1G a Jpeg2000 | 12 | Passato |
| FCV_AT_12 | Format Conversion da Landsat 7 Level1G a Jpeg | 5 | Passato |
| FCV_AT_13 | Format Conversion da Landsat 7 GeoCover a Jpeg2000 | 12 | Passato |
| FCV_AT_14 | Format Conversion da Landsat 7 GeoCover a Jpeg file | 5 | Passato |

Tabella 5.1: Lista degli Acceptance Test eseguiti sui servizi KEI.

| Test ID | Descrizione | Durata (min) | Passato/Fallito |
|-------------|---|--------------|-----------------|
| FCK_AT_15.1 | Format Check archivio Landsat 5 Correct CEOS | 3 | Passato |
| FCK_AT_15.2 | Format Check archivio Landsat 5 Corrupted CEOS | 3 | Passato |
| FCK_AT_16.1 | Format Check archivio Landsat 7 Correct CEOS | 3 | Passato |
| FCK_AT_16.2 | Format Check archivio Landsat 7 Corrupted CEOS | 3 | Passato |
| FCK_AT_17 | Format Check archivio Landsat 7 Level 1G | 3 | Passato |
| FCK_AT_18 | Format Check archivio Landsat 7 GeoCover | 3 | Passato |
| RCC_AT_19 | LIRCC archivio Landsat 5 CEOS | 8 | Passato |
| RCC_AT_20 | LIRCC archivio Landsat 7 CEOS | 8 | Passato |
| RCC_AT_21 | LIRCC archivio Landsat 7 Level 1G | 10 | Passato |
| RCC_AT_22 | LIRCC archivio Landsat 7 GeoCover | 10 | Passato |
| ISC_AT_23.1 | LISC archivio Landsat 5 CEOS - Large set | 10 | Passato |
| ISC_AT_23.2 | LISC archivio Landsat 5 CEOS - Intermediate set | 10 | Passato |
| ISC_AT_23.3 | LISC archivio Landsat 5 CEOS - Small set | 10 | Passato |
| ISC_AT_24.1 | LISC archivio Landsat 7 CEOS - Large set | 10 | Passato |
| ISC_AT_24.2 | LISC archivio Landsat 7 CEOS - Intermediate set | 10 | Passato |
| ISC_AT_24.3 | LISC archivio Landsat 7 CEOS - Small set | 10 | Passato |
| ISC_AT_25.1 | LISC archivio Landsat 7 Level 1G - Large set | 12 | Passato |

Tabella 5.1: Lista degli Acceptance Test eseguiti sui servizi KEI.

| Test ID | Descrizione | Durata (min) | Passato/Fallito |
|-------------|--|--------------|-----------------|
| ISC_AT_25.2 | LISC archivio Landsat 7 Level 1G - Intermediate set | 12 | Passato |
| ISC_AT_25.3 | LISC archivio Landsat 7 Level 1G - Small set | 12 | Passato |
| ISC_AT_26.1 | LISC archivio Landsat 7 GeoCover - Large set | 12 | Passato |
| ISC_AT_26.2 | LISC archivio Landsat 7 GeoCover - Intermediate set | 12 | Passato |
| ISC_AT_26.3 | LISC archivio Landsat 7 GeoCover - Small set | 12 | Passato |
| IREC_AT_27 | Reconditioning: accesso al catalogo, selezione immagine e Clipping | 10 | Passato |
| IREC_AT_28 | Reconditioning : accesso al catalogo, selezione immagine e Format Conversion a GeoTIFF | 12 | Passato |
| IREC_AT_29 | Reconditioning: accesso al catalogo, selezione immagine e Format Conversion a JPEG2000 | 15 | Passato |
| IREC_AT_30 | Reconditioning: accesso al catalogo, selezione immagine e Format Conversion a JPEG | 11 | Passato |
| IREC_AT_31 | Reconditioning: accesso al catalogo, selezione immagine e Format Conversion a JPEG2000 | 15 | Passato |
| IREC_AT_32 | Reconditioning: accesso al catalogo, selezione immagine e Format Conversion a JPEG | 11 | Passato |
| ISPC_AT_33 | Categorization: accesso al catalogo, selezione immagine e Clipping | 10 | Passato |
| ISPC_AT_34 | Categorization: accesso al catalogo, selezione immagine e LIRCC | 10 | Passato |

Tabella 5.1: Lista degli Acceptance Test eseguiti sui servizi KEI.

| Test ID | Descrizione | Durata (min) | Passato/Fallito |
|--------------|--|--------------|-----------------|
| ISPC_AT_35.1 | Categorization: accesso al catalogo, selezione immagine, LIRCC e LISC - Large set | 15 | Passato |
| ISPC_AT_35.2 | Categorization: accesso al catalogo, selezione immagine, LIRCC e LISC - Intermediate set | 14 | Passato |
| ISPC_AT_35.3 | Categorization: accesso al catalogo, selezione immagine, LIRCC e LISC - Small set | 13 | Passato |
| ISPC_AT_36 | Categorization: accesso al catalogo, selezione immagine, Clipping e LIRCC | 14 | Passato |
| ISPC_AT_37.1 | Categorization: accesso al catalogo, selezione immagine, Clipping, LIRCC e LISC - Large set | 17 | Passato |
| ISPC_AT_37.2 | Categorization: accesso al catalogo, selezione immagine, Clipping, LIRCC e LISC - Intermediate set | 16 | Passato |
| ISPC_AT_37.3 | Categorization: accesso al catalogo, selezione immagine, Clipping, LIRCC e LISC - Small set | 15 | Passato |

Tabella 5.1: Lista degli Acceptance Test eseguiti sui servizi KEI.

L'esito positivo di ogni test riportato in Tabella 5.1 richiede la verifica di tutti gli step necessari per accedere al portale SSE ed eseguire il servizio. Nelle Tabelle 5.2 - 5.6 sono riportate la sequenza di operazioni eseguite durante gli AT dei servizi: nella prima colonna è indicata una descrizione dell'operazione eseguita dall'utente; nella seconda colonna eventuali commenti o osservazioni rilevati durante l'esecuzione; nella terza colonna l'esito dello step. Come si può notare dal confronto delle singole tabelle, il primo step di verifica di tutti i servizi coincide (log al portale SSE e selezione del servizio in esame). Poi, in funzione del numero di parametri e/o opzioni selezionabili dall'utente il test presenta un numero di step differente da servizio a servizio.

| Procedura/comando | Commenti/Note | Passato/Fallito |
|---|---------------|-----------------|
| Log al portale SSE e collegamento al servizio di Clipping | | Passato |
| Inserimento del campo indirizzo ftp dell'archivio di input | | Passato |
| Selezione AOI con lo strumento di selezione o inserendo i valori [LLlat,LLlon URLat, URLon] | | Passato |
| Selezione piattaforma satellitare tramite drop-down list | | Passato |
| Selezione formato di input tramite drop-down list | | Passato |
| Esecuzione elaborazione cliccando sul bottone 'Proceed' | | Passato |
| Conferma dell'elaborazione cliccando sul bottone 'Confirm' | | Passato |
| Attesa termine esecuzione elaborazione nella pagina 'Order ID' accessibile dalla 'Order List Page' | | Passato |
| Download del risultato cliccando sul link ftp | | Passato |
| Decompressione dei file contenuti nell'archivio | | Passato |
| Verifica delle immagini denominate 'archivio-nn-ss.tif' e del metadata file 'archiveName-ss.met' | | Passato |
| Apertura delle immagini con GIMP e verifica della coerenza della sotto-scena con l'immagine originale | | Passato |
| Apertura del metadata file e verifica della conformità al metadata file GeoCover | | Passato |

Tabella 5.2: Acceptance Test per il servizio di Clipping: la sequenza di operazioni di verifica del corretto funzionamento del servizio è applicata agli archivi Landsat-5 CEOS, Landsat-7 CEOS, Landsat Level 1G e Landsat GeoCover.

5.2 Problemi durante lo sviluppo e i test

I test riportati in questo capitolo hanno messo in luce vari problemi legati sia ad errori concettuali di programmazione sia all'integrazione sul portale web. I problemi più rilevanti si sono verificati durante lo sviluppo del servizio di Clipping. Nelle prime versioni di questo servizio, infatti, non si erano implementati correttamente gli algoritmi matematici per la trasformazione delle coordinate geografiche in coordinate espresse in pixel all'interno dell'immagine. Una successiva revisione del metodo di conversione delle coordinate ha permesso, nelle versioni successive del servizio, di correggere questo errore legato principalmente ad un uso errato del valore dell'angolo di rotazione

| Procedura/comando | Commenti/Note | Passato/Fallito |
|---|---------------|-----------------|
| Log al portale SSE e collegamento al servizio di Format Conversion | | Passato |
| Inserimento del campo indirizzo ftp dell'archivio di input | | Passato |
| Selezione piattaforma satellitare tramite drop-down list | | Passato |
| Selezione formato di input tramite drop-down list | | Passato |
| Selezione formato di output tramite drop-down list | | Passato |
| Esecuzione elaborazione cliccando sul bottone 'Proceed' | | Passato |
| Conferma dell'elaborazione cliccando sul bottone 'Confirm' | | Passato |
| Attesa termine esecuzione elaborazione nella pagina 'order ID' accessibile dalla 'Order List Page' | | Passato |
| Download del risultato cliccando sul link ftp | | Passato |
| Decompressione dei file contenuti nell'archivio | | Passato |
| Verifica delle immagini denominate 'archivio-nn.tif' e del metadata file 'archiveName.met' | | Passed |
| Apertura delle immagini con GIMP e verifica della coerenza della sotto-scena con l'immagine originale | | Passato |
| Apertura del metadata file e verifica della conformità al metadata file GeoCover | | Passato |

Tabella 5.3: Acceptance Test per il servizio di Format Conversion: la sequenza di operazioni di verifica del corretto funzionamento del servizio è applicata agli archivi Landsat-5 CEOS, Landsat-7 CEOS, Landsat Level 1G e Landsat GeoCover. Il numero totale di AT condotti sul servizio di Format Conversion è di 10 in accordo con le possibili combinazioni tra formati di input e output.

dell'immagine all'interno della matrice di conversione. Il problema è stato identificato quando sono stati elaborati archivi di immagini ruotate. Durante lo sviluppo iniziale del servizio si era fatto uso di immagini con angolo di rotazione pari a zero, che non consentivano quindi una verifica completa della procedura di estrazione di sotto-scene implementata. In fase di test è invece stato utilizzato un database di riferimento più ampio, in cui la presenza di archivi di immagini ruotate ha evidenziato i problemi relativi alla conversione delle coordinate.

Problemi minori dal punto di vista dell'implementazione si sono riscontrati nella creazione e gestione di strutture dati dinamiche per gestire il contenuto dei metadata

| Procedura/comando | Commenti/Note | Passato/Fallito |
|--|---------------|-----------------|
| Log al portale SSE e collegamento al servizio di Format Check | | Passato |
| Inserimento del campo indirizzo ftp dell'archivio di input | | Passato |
| Selezione piattaforma satellitare tramite drop-down list | | Passato |
| Selezione formato di input tramite drop-down list | | Passato |
| Esecuzione elaborazione cliccando sul bottone 'Proceed' | | Passato |
| Conferma dell'elaborazione cliccando sul bottone 'Confirm' | | Passato |
| Attesa termine esecuzione elaborazione nella pagina 'order ID' accessibile dalla 'Order List Page' | | Passato |
| Verifica del messaggio 'Valid/Non Valid' | | Passato |
| Verifica tra risultato e archivio in input | | Passato |

Tabella 5.4: Acceptance Test per il servizio di Format Check: la sequenza di operazioni di verifica del corretto funzionamento del servizio è applicata agli archivi Landsat-5 CEOS, Landsat-7 CEOS, Landsat Level 1G e Landsat GeoCover.

file degli archivi satellitari Landsat Level 1G e Landsat GeoCover. Infatti una prima analisi sui metafile aveva portato a considerarli completamente testuali, mentre in seconda battuta si è verificata la presenza in coda al file di una parte binaria. In questo caso, per risolvere il problema è bastato ignorare, durante l'acquisizione del file, la parte binaria in cui non sono contenute informazioni necessarie all'esecuzione dei servizi.

| Procedura/comando | Commenti/Note | Passato/Fallito |
|--|---------------|-----------------|
| Log al portale SSE e collegamento al servizio di LIRCC | | Passato |
| Inserimento del campo indirizzo ftp dell'archivio di input | | Passato |
| Selezione piattaforma satellitare tramite drop-down list | | Passato |
| Selezione formato di input tramite drop-down list | | Passato |
| Esecuzione elaborazione cliccando sul bottone 'Proceed' | | Passato |
| Conferma dell'elaborazione cliccando sul bottone 'Confirm' | | Passato |
| Attesa termine esecuzione elaborazione nella pagina 'order ID' accessibile dalla 'Order List Page' | | Passato |
| Download del risultato cliccando sul link ftp | | Passato |
| Decompressione dei file contenuti nell'archivio | | Passato |
| Verifica delle immagini denominate 'archivio-nn.tif' e del metadata file 'archiveName.met' | | Passato |
| Apertura delle immagini con ENVI e verifica delle immagini processate | | Passato |
| Apertura del file 'archivio-cal.out' e verifica del contenuto (lista dei nomi dei file processati) | | Passato |

Tabella 5.5: Acceptance Test per il servizio di LIRCC: la sequenza di operazioni di verifica del corretto funzionamento del servizio è applicata agli archivi Landsat-5 CEOS, Landsat-7 CEOS, Landsat Level 1G e Landsat GeoCover.

| Procedura/comando | Commenti/Note | Passato/Fallito |
|--|---------------|-----------------|
| Log al portale SSE e collegamento al servizio di LISC | | Passato |
| Inserimento del campo indirizzo ftp dell'archivio di input | | Passato |
| Selezione piattaforma satellitare tramite drop-down list | | Passato |
| Selezione del livello di classificazione tramite drop-down list | | Passato |
| Esecuzione elaborazione cliccando sul bottone 'Proceed' | | Passato |
| Conferma dell'elaborazione cliccando sul bottone 'Confirm' | | Passato |
| Attesa termine esecuzione elaborazione nella pagina 'order ID' accessibile dalla 'Order List Page' | | Passato |
| Download del risultato cliccando sul link ftp | | Passato |
| Decompressione dell'archivio 'archivio-sc.zip' | | Passato |
| Verifica della mappa di classificazione in formato GeoTIFF | | Passato |

Tabella 5.6: Acceptance Test per il servizio di LISC: la sequenza di operazioni di verifica del corretto funzionamento del servizio è applicata all'archivio di output del servizio LIRCC.

Capitolo 6

Conclusioni

Questo lavoro di tesi presenta alcuni aspetti che vanno evidenziati. L'ambito dell'Osservazione della Terra da Satellite è un ambito multi-disciplinare dove le competenze informatiche rivestono sicuramente un ruolo fondamentale. Infatti i problemi riscontrati nella gestione di differenti formati di archiviazione dei dati hanno richiesto un intenso sforzo nell'apprendere l'utilizzo delle librerie GDAL, tiff e libgeotiff. Inoltre, la manipolazione dei dati satellitari non può prescindere da una approfondita conoscenza dei meccanismi di georeferenziazione che ho introdotto nel software sviluppato. Infine per portare a termine il lavoro svolto durante il tirocinio è risultato indispensabile una formazione preliminare sulle tipologie di satelliti, sul funzionamento dei vari sensori e più in generale sui concetti fisici del **Remote Sensing**.

Oltre alle competenze specifiche maturate, lo svolgimento della tesi presso l'azienda MEEO mi ha permesso di essere inserito in un team di sviluppo e quindi di poter acquisire ed approfondire le procedure e le metodiche utilizzate nel lavoro di gruppo.

L'esperienza in MEEO è stata formativa dal punto di vista dell'applicazione di standard qualitativi nell'ambito della Ricerca e Sviluppo: il progetto KEI, all'interno del quale rientrano le attività descritte, come tutti i progetti ESA, deve essere condotto secondo i rigidi standard ESA (ECSS) , che impongono la redazione di una ricca e dettagliata documentazione, l'applicazione di tecniche di tracking degli errori e approfondite attività di test e di valutazione del software implementato.

Il software prodotto, dopo aver superato i test di verifica e validazione aziendali, è

stato consegnato all'ESA per ulteriori test di verifica a cui ha fatto seguito l'integrazione su portale SSE dove è attualmente operativo. Le applicazioni sviluppate sono specifiche per determinati archivi satellitari LANDSAT e SPOT, tuttavia il progetto KEI prevede l'estensione dei software sviluppati ad altri sensori satellitari quali, ad esempio, Moderate Resolution Imaging Spectroradiometer (MODIS) e Advanced Very High Resolution Radiometer (AVHRR). L'applicazione del software sviluppato per questi sensori richiederà una completa rivisitazione dei moduli nella parte di lettura dei dati.

La tecnologia per l'Osservazione della Terra da Satellite è in continua evoluzione e la quantità di dati prodotta aumenta esponenzialmente con il passare del tempo. Per contro c'è una grande carenza di figure professionali con le competenze adeguate per manipolare ed estrarre informazioni utili da questi dati, quindi l'esperienza di questa tesi ha avuto un ruolo altamente qualificante dal punto di vista professionale.

Appendice A

Codici sorgente

A.1 Programma per il clipping: clipgtiff

A.1.1 clipgtiff.c

```
1 #include <stdio.h>
2 #include <gdal.h>
3 #include <string.h>
4 #include <ogr_srs_api.h>
5 #include "LatLong-UTMconversion.h"
6
7 int findMin(int a, int b, int c, int d);
8 int findMax(int a, int b, int c, int d);
9
10 int main(int argc, char **argv){
11     int          i, pbNorth, zone;
12     int          pxSizeX, pxSizeY;
13     int          BSizeX, BSizeY;
14     char         *pszProjection, *newProjection;
15     char         *name_in, *name_out;
16     double       adfGeoTransform[6];
17     double       matrix_inv[4];
```

```

18
19     GDALDatasetH      hDataset , hDstDS;
20     GDALDriverH      hDriver , hDDst;
21     GByte             *abyRaster;
22     OGRSpatialReferenceH  hSRS;
23     GDALRasterBandH  hBandSrc , hBandDst;
24
25     // 0 = Lat in G, 1 = Lon in G,
26     // 2 = UTMNorthing 3 = UTMEasting
27     double LL[4], UR[4], UL[4], LR[4];
28     double imgLL[2], imgUR[2];
29     // 0 = x, 1 = y
30     int pxLL[2], pxUR[2], pxUL[2], pxLR[2];
31     int BoxLL[2], BoxUR[2];
32     GDALAllRegister();
33     CPLSetErrorHandler(NULL);
34     if( argc != 7){
35         printf("Usage: \_clipgtiff \_[INPUT\_FILE] \_[OUTPUT\_FILE] \_[Low \_
           Left \_Lat] \_[Low \_Left \_Lon] \_[Up \_Right \_Lat] \_[Up \_Right \_Lon] \_
           n");
36         return(1);
37     }
38     name_in  = argv[1];
39     name_out = argv[2];
40     LL[0] = atof(argv[3]);
41     LL[1] = atof(argv[4]);
42     UR[0] = atof(argv[5]);
43     UR[1] = atof(argv[6]);
44     hDataset = GDALOpen( name_in , GA_ReadOnly );
45     if( hDataset == NULL ){
46         printf("Unable \_open \_file \_%s \_n", name_in);
47         return(1);
48     }

```

```
49     #ifdef DEBUG
50     printf( "%s -> %s ( LL: %.2f/%.2f UR: %.2f/%.2f )\n",
51           name_in, name_out, LL[0], LL[1], UR[0], UR[1] );
52     #endif
53     GDALGetGeoTransform( hDataset, adfGeoTransform );
54     adfGeoTransform[5] = fabs( adfGeoTransform[5] );
55     matrix_inv[0] = adfGeoTransform[5] / ( adfGeoTransform[1]*
56     adfGeoTransform[5] - adfGeoTransform[2]*adfGeoTransform
57     [4] );
58     matrix_inv[1] = -adfGeoTransform[2] / ( adfGeoTransform[1]*
59     adfGeoTransform[5] - adfGeoTransform[2]*adfGeoTransform
60     [4] );
61     matrix_inv[2] = -adfGeoTransform[4] / ( adfGeoTransform[1]*
62     adfGeoTransform[5] - adfGeoTransform[2]*adfGeoTransform
63     [4] );
64     matrix_inv[3] = adfGeoTransform[1] / ( adfGeoTransform[1]*
65     adfGeoTransform[5] - adfGeoTransform[2]*adfGeoTransform
66     [4] );
67     #ifdef DEBUG
68     printf( "Rotation Matrix: \n\t%f\t%f\n\t%f\t%f\n",
69           adfGeoTransform[1], adfGeoTransform[2], adfGeoTransform
70           [4], adfGeoTransform[5] );
71     printf( "Inverse Matrix: \n\t%f\t%f\n\t%f\t%f\n", matrix_inv
72           [0], matrix_inv[1], matrix_inv[2], matrix_inv[3] );
73     #endif
74     pszProjection = (char *) GDALGetProjectionRef( hDataset );
75     hSRS = OSRNewSpatialReference( NULL );
76     newProjection = (char *) malloc( sizeof(char) * strlen(
77     pszProjection ) );
78     if( newProjection == NULL ) {
79         // Insufficient memory for allocation handling
80         printf( "Allocation of raster failed!" );
81         exit( 1 );
82     }
```

```

69     }
70     strcpy(newProjection , pszProjection);
71     OSRImportFromWkt( hSRS, &pszProjection );
72     zone = OSRGetUTMZone( hSRS, NULL);
73     #ifdef DEBUG
74     printf("Zone: %d\n", zone);
75     #endif
76     UL[0] = UR[0];
77     UL[1] = LL[1];
78     LR[0] = LL[0];
79     LR[1] = UR[1];
80     LLtoUTM(LL[0], LL[1], &LL[2], &LL[3], zone);
81     LLtoUTM(UR[0], UR[1], &UR[2], &UR[3], zone);
82     LLtoUTM(UL[0], UL[1], &UL[2], &UL[3], zone);
83     LLtoUTM(LR[0], LR[1], &LR[2], &LR[3], zone);
84     pxSizeX = GDALGetRasterXSize(hDataset);
85     pxSizeY = GDALGetRasterYSize(hDataset);
86     imgLL[1] = adfGeoTransform[0] + (( adfGeoTransform[1]
87         * 0) + ( adfGeoTransform[4] * (pxSizeY-1)));
88     imgLL[0] = adfGeoTransform[3] - abs((( adfGeoTransform
89         [2] * 0) + ( adfGeoTransform[5] * (pxSizeY-1))));
90     imgUR[1] = adfGeoTransform[0] + (( adfGeoTransform[1]
91         * (pxSizeX-1)) + ( adfGeoTransform[2] * 0));
92     imgUR[0] = adfGeoTransform[3] - abs((( adfGeoTransform
93         [4] * (pxSizeX-1)) + ( adfGeoTransform[5] * 0));
94     #ifdef DEBUG
95     printf("Image coord:\n");
96     printf("\tLL: %f/%f\tUR: %f/%f\n", imgLL[0], imgLL[1],
97         imgUR[0],imgUR[1]);
98     printf("Sub_S_coord:\n");
99     printf("\tLL: %f/%f\tUR: %f/%f\n", LL[2], LL[3], UR
100         [2], UR[3]);
101     #endif

```

```
96 // Calc pixel coord
97 pxLR[0] = (int)(( matrix_inv[2] * (LL[2]-adfGeoTransform
    [3]) ) + ( matrix_inv[0] * (UR[3]-adfGeoTransform[0])))
    ;
98 pxLR[1] = (int)(( matrix_inv[3] * (LL[2]-adfGeoTransform
    [3]) ) + ( matrix_inv[1] * (UR[3]-adfGeoTransform[0])))
    * -1;
99 pxLL[0] = (int)(( matrix_inv[2] * (LL[2]-adfGeoTransform
    [3]) ) + ( matrix_inv[0] * (LL[3]-adfGeoTransform[0])))
    ;
100 pxLL[1] = (int)(( matrix_inv[3] * (LL[2]-adfGeoTransform
    [3]) ) + ( matrix_inv[1] * (LL[3]-adfGeoTransform[0])))
    * -1;
101 pxUR[0] = (int)(( matrix_inv[2] * (UR[2]-adfGeoTransform
    [3]) ) + ( matrix_inv[0] * (UR[3]-adfGeoTransform[0])))
    ;
102 pxUR[1] = (int)(( matrix_inv[3] * (UR[2]-adfGeoTransform
    [3]) ) + ( matrix_inv[1] * (UR[3]-adfGeoTransform[0])))
    * -1;
103 pxUL[0] = (int)(( matrix_inv[2] * (UR[2]-adfGeoTransform
    [3]) ) + ( matrix_inv[0] * (LL[3]-adfGeoTransform[0])))
    ;
104 pxUL[1] = (int)(( matrix_inv[3] * (UR[2]-adfGeoTransform
    [3]) ) + ( matrix_inv[1] * (LL[3]-adfGeoTransform[0])))
    * -1;
105
106 hDriver = GDALGetDatasetDriver( hDataset );
107 hDDst = GDALGetDriverByName( "GTiff" );
108
109 #ifdef DEBUG
110 printf("Pixel:\n");
111 printf("\t%d/%d\t%d/%d\n", pxUL[0], pxUL[1], pxUR[0], pxUR
    [1]);
```

```

112     printf( "\t%d/%d\t%d/%d\n", pxLL[0], pxLL[1], pxLR[0], pxLR
113           [1]);
114     #endif
115
116     BoxLL[0] = findMin( pxUL[0], pxLL[0], pxUR[0], pxLR[0]);
117     BoxLL[1] = findMin( pxUL[1], pxLL[1], pxUR[1], pxLR[1]);
118     BoxUR[0] = findMax( pxUL[0], pxLL[0], pxUR[0], pxLR[0]);
119     BoxUR[1] = findMax( pxUL[1], pxLL[1], pxUR[1], pxLR[1]);
120     if( ( BoxLL[0] > pxSizeX ) || ( BoxLL[1] > pxSizeY ) ){
121         printf( "Invalid coords of subscene 5\n");
122         exit(2);
123     }
124     if( ( BoxUR[0] < 0 ) || ( BoxUR[1] < 0 ) ){
125         printf( "Invalid coords of subscene 6\n");
126         exit(2);
127     }
128     if( BoxLL[0] < 0 ) BoxLL[0] = 0;
129     if( BoxLL[1] < 0 ) BoxLL[1] = 0;
130     if( BoxUR[0] > (pxSizeX-1) ) BoxUR[0] = pxSizeX - 1;
131     if( BoxUR[1] > (pxSizeY-1) ) BoxUR[1] = pxSizeY - 1;
132     BSizeX = BoxUR[0] - BoxLL[0];
133     BSizeY = BoxUR[1] - BoxLL[1];
134     if ( ( BSizeX % 2 ) == 0 ) BSizeX++;
135     if ( ( BSizeY % 2 ) == 0 ) BSizeY++;
136
137     #ifdef DEBUG
138     printf( "LL: %.2f Lat/ %.2f Lon\ t-> %.2fE/ %.2fN\n", LL[0],
139           LL[1], LL[3], LL[2]);
140     printf( "UR: %.2f Lat/ %.2f Lon\ t-> %.2fE/ %.2fN\n", UR[0],
141           UR[1], UR[3], UR[2]);
142     printf( "Pixel:\n");
143     printf( "\t%d/%d\t%d/%d\n", pxUL[0], pxUL[1], pxUR[0], pxUR
144           [1]);

```

```

141     printf("\t%d/%d\t%d/%d\n", pxLL[0], pxLL[1], pxLR[0],pxLR
        [1]);
142     printf("Box coords:\n");
143     printf("\tLL: %dx%d\tUR: %dx%d\n",BoxLL[0] ,BoxLL[1],BoxUR
        [0] ,BoxUR[1]);
144 #endif
145
146     if( ( BSizeX > pxSizeX ) && (BSizeY > pxSizeY) ){
147         printf("Invalid coords of subscene 10\n");
148         exit(2);
149     }
150     if( ( BSizeX == 1 ) || (BSizeY == 1 ) ){
151         printf("Invalid coords of subscene 11\n");
152         exit(2);
153     }
154     abyRaster = (GByte *) malloc(sizeof(GByte)* BSizeX *
        BSizeY );
155     if(abyRaster == NULL){
156         // Insufficient memory for allocation handling
157         printf("Allocation of raster failed!");
158         exit(1);
159     }
160 #ifdef DEBUG
161     printf("Src image size: %dx%d\n",pxSizeX , pxSizeY);
162     printf("\t\tadfGeoTransform:\n");
163     for(i = 0 ; i < 6 ; i++){
164         printf("\t[%d]: %f\n", i ,adfGeoTransform [ i ]);
165     }
166 #endif
167     adfGeoTransform [0] =      adfGeoTransform [0] +
168         (( adfGeoTransform [1] * (BoxLL[0]-0.5)) + (
            adfGeoTransform [2] * (BoxLL[1]+0.5)));
169     adfGeoTransform [3] = adfGeoTransform [3] -

```

```

170         abs((( adfGeoTransform [4] * (BoxLL[0]-0.5)) +
171             ( adfGeoTransform [5] * (BoxLL[1]+0.5)))));
172 adfGeoTransform [0] += adfGeoTransform [1] * 0.5;
173 adfGeoTransform [3] += adfGeoTransform [1] * 0.5;
174
175 #ifdef DEBUG
176 printf("Box size : %dx%d\n", BSizeX, BSizeY);
177 printf("adfGeoTransform:\n");
178 for(i = 0 ; i < 6 ; i++){
179     printf("\t[%d]: %f\n", i, adfGeoTransform [ i ] );
180 }
181 #endif
182 hDstDS = GDALCreate( hDriver, name_out, BSizeX, BSizeY, 1,
183     GDT_Byte, NULL );
184 if( hDstDS == NULL ){
185     printf("Unable to create GeoTIFF file %s\n",
186         name_out);
187     exit(1);
188 }
189 GDALSetGeoTransform( hDstDS, adfGeoTransform );
190 GDALSetProjection( hDstDS, newProjection );
191 hBandSrc = GDALGetRasterBand( hDataset, 1 );
192 hBandDst = GDALGetRasterBand( hDstDS, 1 );
193 #ifdef DEBUG
194 printf("Load src image ->");
195 #endif
196 GDALRasterIO( hBandSrc, GF_Read, BoxLL[0], BoxLL[1],
197     BSizeX, BSizeY, abyRaster, BSizeX, BSizeY, GDT_Byte,
198     0, 0 );
199 #ifdef DEBUG
200 printf("Write dst image ->");
201 #endif
202 GDALRasterIO( hBandDst, GF_Write, 0, 0, BSizeX, BSizeY,

```

```
        abyRaster, BSizeX, BSizeY, GDT_Byte, 0, 0);
198 #ifdef DEBUG
199     printf("Done\n");
200 #endif
201     GDALClose( hDstDS );
202 }
203
204 int findMin(int a, int b, int c, int d){
205     int i, r;
206     int tmp[4];
207     tmp[0] = a; tmp[1] = b; tmp[2] = c; tmp[3] = d;
208     r = tmp[0];
209     for (i = 0 ; i < 4 ; i++){
210         if ( tmp[i] < r )
211             r = tmp[i];
212     }
213     return(r);
214
215 }
216
217 int findMax(int a, int b, int c, int d){
218     int i, r;
219     int tmp[4];
220     tmp[0] = a; tmp[1] = b; tmp[2] = c; tmp[3] = d;
221     r = tmp[0];
222     for (i = 0 ; i < 4 ; i++){
223         if ( tmp[i] > r )
224             r = tmp[i];
225     }
226     return(r);
227 }
```

A.1.2 LatLong-UTMconversion.c

```
1 #include <math.h>
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include "LatLong-UTMconversion.h"
5
6
7 void LLtoUTM(double Lat, double Long, double *UTMNorthing,
8             double *UTMEasting, int ZoneNumber){
9     double a = 6378137;
10    double eccSquared = 0.00669438;
11    double deg2rad;
12    double k0 = 0.9996;
13    double LongOrigin;
14    double eccPrimeSquared;
15    double N, T, C, A, M;
16    double LongTemp;
17    double LatRad;
18    double LongRad;
19    double LongOriginRad;
20    deg2rad = M_PI / 180;
21    LongTemp = (Long+180)-(int)((Long+180)/360)*360-180;
22    LatRad = Lat*deg2rad;
23    LongRad = LongTemp*deg2rad;
24    if( Lat >= 56.0 && Lat < 64.0 && LongTemp >= 3.0 &&
25        LongTemp < 12.0 )
26        ZoneNumber = 32;
27
28    if( Lat >= 72.0 && Lat < 84.0 ){
29        if( LongTemp >= 0.0 && LongTemp < 9.0 )
30            ZoneNumber = 31;
31        else if( LongTemp >= 9.0 && LongTemp < 21.0 )
32            ZoneNumber = 33;
33        else if( LongTemp >= 21.0 && LongTemp < 33.0 )
```

```

ZoneNumber = 35;
30     else if( LongTemp >= 33.0 && LongTemp < 42.0 )
        ZoneNumber = 37;
31 }
32 LongOrigin = (ZoneNumber - 1)*6 - 180 + 3; //+3 puts
        origin in middle of zone
33 LongOriginRad = LongOrigin * deg2rad;
34 eccPrimeSquared = (eccSquared)/(1 - eccSquared);
35 N = a/sqrt(1 - eccSquared * sin(LatRad) * sin(LatRad));
36 T = tan(LatRad) * tan(LatRad);
37 C = eccPrimeSquared * cos(LatRad) * cos(LatRad);
38 A = cos(LatRad) * (LongRad - LongOriginRad);
39 M = a * ((1 - eccSquared / 4 - 3 * eccSquared * eccSquared / 64 - 5 *
        eccSquared * eccSquared * eccSquared / 256) * LatRad
40 - (3 * eccSquared / 8 + 3 * eccSquared * eccSquared / 32 + 45 *
        eccSquared * eccSquared * eccSquared / 1024) * sin(2 * LatRad)
41 + (15 * eccSquared * eccSquared / 256 + 45 * eccSquared * eccSquared
        * eccSquared / 1024) * sin(4 * LatRad)
42 - (35 * eccSquared * eccSquared * eccSquared / 3072) * sin(6 * LatRad)
        );
43 *UTMEasting = (double)(k0 * N * (A + (1 - T + C) * A * A * A / 6 + (5 - 18 * T + T
        * T + 72 * C - 58 * eccPrimeSquared) * A * A * A * A / 120) + 500000.0);
44 *UTMNorthing = (double)(k0 * (M + N * tan(LatRad) * (A * A / 2 + (5 - T + 9 *
        C + 4 * C * C) * A * A * A * A / 24
45 + (61 - 58 * T + T * T + 600 * C - 330 * eccPrimeSquared) * A * A * A * A *
        A * A / 720)));
46 if(Lat < 0)
47     *UTMNorthing += 10000000.0; //10000000 meter offset
        for southern hemisphere
48 }
49
50 void UTMtoLL(double UTMNorthing, double UTMEasting, int
        ZoneNumber, int NorthernHemisphere, double *Lat, double *

```

```

Long ){
51     double a = 6378137;
52     double eccSquared = 0.00669438;
53     double k0 = 0.9996;
54     double eccPrimeSquared ;
55     double e1 = (1-sqrt(1-eccSquared))/(1+sqrt(1-
        eccSquared));
56     double N1, T1, C1, R1, D, M;
57     double LongOrigin;
58     double mu, phi1 , phi1Rad;
59     double x, y;
60     char* ZoneLetter;
61     double rad2deg;
62     rad2deg = 180.0 / M_PI;
63     x = UTM Easting - 500000.0;
64     y = UTM Northing;
65     NorthernHemisphere -= 6;
66     if ( NorthernHemisphere != 0){ ;
67         //point is in southern hemisphere
68         y -= 1000000.0;//remove 10,000,000 meter
           offset used for southern hemisphere
69     }
70     LongOrigin = (ZoneNumber - 1)*6 - 180 + 3; //+3 puts
           origin in middle of zone
71     eccPrimeSquared = (eccSquared)/(1-eccSquared);
72     M = y / k0;
73     mu = M/( a*(1-eccSquared/4-3*eccSquared*eccSquared
           /64-5*eccSquared*eccSquared*eccSquared/256));
74     phi1Rad = mu + (3*e1/2-27*e1*e1*e1/32)*sin(2*mu)
75                 + (21*e1*e1/16-55*e1*e1*e1*e1
           /32)*sin(4*mu)
76                 +(151*e1*e1*e1/96)*sin(6*mu);
77     phi1 = phi1Rad*rad2deg;

```

```

78     N1 = a / sqrt(1 - eccSquared * sin(phi1Rad) * sin(phi1Rad));
79     T1 = tan(phi1Rad) * tan(phi1Rad);
80     C1 = eccPrimeSquared * cos(phi1Rad) * cos(phi1Rad);
81     R1 = a * (1 - eccSquared) / pow(1 - eccSquared * sin(phi1Rad) *
      sin(phi1Rad), 1.5);
82     D = x / (N1 * k0);
83     *Lat = phi1Rad - (N1 * tan(phi1Rad) / R1) * (D * D / 2 - (5 + 3 * T1
      + 10 * C1 - 4 * C1 * C1 - 9 * eccPrimeSquared) * D * D * D * D
      / 24 + (61 + 90 * T1 + 298 * C1 + 45 * T1 * T1 - 252 * eccPrimeSquared
      - 3 * C1 * C1) * D * D * D * D * D * D / 720);
84     *Lat = *Lat * rad2deg;
85     *Long = (D - (1 + 2 * T1 + C1) * D * D * D / 6 + (5 - 2 * C1 + 28 * T1 - 3 * C1 * C1
      + 8 * eccPrimeSquared + 24 * T1 * T1)
86                                     * D * D * D * D * D / 120) / cos(
      phi1Rad);
87     *Long = LongOrigin + *Long * rad2deg;
88 }

```

A.1.3 LatLong-UTMconversion.h

```

1  #include <string.h>
2
3  #ifndef LATLONGCONV
4  #define LATLONGCONV
5
6  void LLtoUTM(double Lat, double Long, double *UTMNorthing,
      double *UTMEasting, int ZoneNumber);
7  void UTMtoLL(double UTMNorthing, double UTMEasting, int
      ZoneNumber, int NorthernHemisphere, double *Lat, double *
      Long );
8
9  #endif

```

A.2 Script per la concatenazione dei servizi: fullchain

A.2.1 fullchain.sh

```
1 #!/bin/bash
2
3 TMPDIR="$1"
4 ZIP="$2"
5 SENSOR="$3"
6 INFORMAT="$4"
7 LOCAL_WD="$(echo "$0" | rev | cut -f 2 -d "/" | rev)"
8
9 [ -z "$5" ] && exit
10 argv=( $* )
11 if [ "GeoTIFF" != "$INFORMAT" ] && [ "CEOS" != "$INFORMAT" ]
12     ; then
13     echo "Invalid input format: $INFORMAT"
14     exit 3
15 fi
16 if [ ! -d "$TMPDIR" ] ; then
17     echo "$TMPDIR is not a directory"
18     exit 5
19 fi
20 if [ ! -f "$TMPDIR/$ZIP" ] ; then
21     echo "$ZIP not exists"
22     exit 6
23 fi
24 if [ "$SENSOR" != "landsat7" ] && [ "$SENSOR" != "landsat5" ]
25     ; then
26     echo "Invalid Platform"
27     exit 8
28 fi
```

```

28 if [ "$SENSOR" = "landsat5" ] && [ "GeoTIFF" = "$INFORMAT" ]
    ; then
29     echo "Invalid Platform for GeoTIFF format"
30     exit 8
31 fi
32 SUBDIR="dir$$"
33 echo "Create subdir $$SUBDIR for chain ..."
34 mkdir "$TMPDIR/$SUBDIR"
35 cd "$TMPDIR/$SUBDIR"
36 echo "Unzipping section for chain ..."
37 unzip -o "../$ZIP"
38 if [ "$?" != "0" ] ; then
39     echo "Invalid zip format"
40     rm -rf "$TMPDIR/$SUBDIR"
41     exit 7
42 fi
43 cd - &> /dev/null
44 cd "$LOCAL_WD"
45 ROOT_DIR="$(pwd | rev | cut -f3 -d"/" | rev)"
46 echo "Root of services: $ROOT_DIR"
47 ##### Configurazion PATH scripts #####
48 formatCheck="$ROOT_DIR/formatCheck/scripts/formatCheck.sh"
49 formatConversion="$ROOT_DIR/formatConversion/scripts/
    formatConversion.sh"
50 clipping="$ROOT_DIR/clipping/scripts/clipping.sh"
51 lircc="$ROOT_DIR/lircc/scripts/lircc.sh"
52 lisc="$ROOT_DIR/lisc/scripts/lisc.sh"
53 #####
54 for i in $formatCheck $formatConversion $clipping $lircc $lisc
    ; do
55     if [ ! -f "$i" ] ; then
56         echo "$i invalid path!"
57         exit 30

```

```
58     fi
59 done
60 HOW_TO_EXIT="ok"
61 goWorkDir () {
62     echo "$1" | rev | cut -f 2- -d "/" | rev
63 }
64
65 rm () {
66     echo "-->Delete_disable ,_chained_service"
67     return 0
68 }
69 rmdir () {
70     echo "-->Delete_disable ,_chained_service"
71     return 0
72 }
73 exit () {
74     if [ ! "$1" = "0" ] ; then
75         echo "Chain_broke ,_process_exit_with_$1_code"
76         HOW_TO_EXIT="bad"
77         break
78     fi
79     echo "-->Exit_0_disable ,_chained_service"
80 }
81 unzip () {
82     echo "-->Unzip_disable ,_chained_service"
83     return 0
84 }
85 zip () {
86     echo "-->Zip_disable ,_chained_service"
87     return 0
88 }
89 mkdir () {
90     echo "-->Make_dir_disable ,_chained_service"
```

```

91     return 0
92 }
93 mv() {
94     echo "-->_Move_modify ,_chained_service"
95     /bin/mv $*
96     return 0
97
98 }
99 #####
100
101 echo "-----_Start_Chain_"
102 cnt_arg=4
103 clean="false"
104 END_ZIP_NAME="$(_echo "$ZIP" |_rev |_cut -f2-d"." |_rev)"
105
106 while [ ! -z "${argv[$cnt_arg]}" ] ; do
107     echo -e "#####_Start_${argv[
108         $cnt_arg ]}#####"
109     old_list_file="$(_find $TMPDIR -type f |_grep -vi ".zip" |
110         _grep -vi "output.log" |_tr " " "#" )"
111     case "${argv[$cnt_arg]}" in
112         Check)
113             cd $( goWorkDir $formatCheck )
114             . $formatCheck "$TMPDIR" "$ZIP" "$SENSOR" "$
115                 $INFORMAT"
116             clean="false"
117             ;;
118         Conversion)
119             cd $( goWorkDir $formatConversion )
120             cnt_arg=$(( cnt_arg + 1 ))
121             OUTFORMAT="${argv[$cnt_arg]}"
122             . $formatConversion "$TMPDIR" "$ZIP" "$SENSOR" "$
123                 $INFORMAT" "$OUTFORMAT"

```

```

120     [ "$INFORMAT" = "CEOS" ] && [ ! "$OUTFORMAT" = "
121         GeoTIFF" ] && /bin/rm -f $TMPDIR/*.tif
122     INFORMAT="$OUTFORMAT"
123     clean="true"
124     END_ZIP_NAME="$END_ZIP_NAME-$OUTFORMAT"
125 ;;
126 Clipping )
127     cnt_arg=$(( cnt_arg + 1 ))
128     chain_LL="{ argv[ $cnt_arg ] }"
129     cnt_arg=$(( cnt_arg + 1 ))
130     chain_UR="{ argv[ $cnt_arg ] }"
131     cd $( goWorkDir $clipping )
132     . $clipping "$TMPDIR" "$ZIP" "$SENSOR" "$INFORMAT"
133         "$chain_LL" "$chain_UR"
134     clean="true"
135     END_ZIP_NAME="$END_ZIP_NAME-ss"
136     INFORMAT="GeoTIFF"
137     cd "$TMPDIR/$SUBDIR/"
138     /bin/rm -f $( ls | grep -iv "_ss" )
139 ;;
140 Lircc )
141     cnt_arg=$(( cnt_arg + 1 ))
142     clipped="{ argv[ $cnt_arg ] }"
143     [ ! "$clipped" = "clipped" ] && cnt_arg=$((
144         cnt_arg - 1 ))
145     cd $( goWorkDir $lircc )
146     . $lircc "$TMPDIR" "$ZIP" "$SENSOR" "$
147         INFORMAT" "$clipped"
148     cd "$TMPDIR/$SUBDIR/"
149     /bin/rm -f $( ls | grep -iv "_cal" )
150     clean="true"
151     END_ZIP_NAME="$END_ZIP_NAME-cal"

```

```

149         INFORMAT="GeoTIFF"
150     ;;
151     Lisc)
152         cd $( goWorkDir $lisc )
153         cnt_arg=$(( cnt_arg + 1 )
154         type="{argv[ $cnt_arg ]}"
155         . $lisc "$TMPDIR" "$ZIP" "$type"
156         echo "Clean tmp files ..."
157         /bin/rm -f $TMPDIR/$SUBDIR/rcc*.tif
158         clean="true"
159         END_ZIP_NAME="$END_ZIP_NAME-sc"
160         INFORMAT="GeoTIFF"
161
162     ;;
163     *)
164         echo "Unknow option ${argv[ $cnt_arg ]}"
165         clean="false"
166         exit 19
167     ;;
168     esac
169     if [ "$clean" = "true" ] ; then
170         new_list_file="$( find $TMPDIR -type f | grep -vi "
171             output.log" )"
172         echo "Clean directory and prepare for new elaborate ..."
173
174         for i in $new_list_file ; do
175             file_name="$( echo "$i" | rev | awk -F/ '{ print $1
176                 }' | rev )"
177             if [ ! -z "$( echo "$old_list_file" | rev | grep
178                 "^$( echo "$file_name" | rev )" | rev )" ] ;
179                 then
180                 /bin/rm -fr "$i" 2> /dev/null
181             else

```

```

177         [ ! -d "$TMPDIR/$SUBDIR" ] && /bin/mkdir "
           $TMPDIR/$SUBDIR"
178         if [ -f "$i" ] ; then
179             if [ -z "$(_echo "$i" | _grep -i ".zip")"
                ] ; then
180                 /bin/mv -f "$i" "$TMPDIR/$SUBDIR" 2> /
                    dev/null
181             fi
182         fi
183     fi
184 done
185     find $TMPDIR -type d -exec /bin/rmdir {} \; &> /dev/
        null
186 fi
187 clean="false"
188 cnt_arg=$(( cnt_arg + 1 )
189 done
190 echo "End_chain."
191 if [ "$SHOW_TO_EXIT" = "ok" ] ; then
192     if [ ! "$END_ZIP_NAME" = "$(_echo "$ZIP" | _rev | _cut -f 2-
        _d "." | _rev)" ] ; then
193         echo "Make_final_zip_file ..."
194         END_ZIP_NAME="$END_ZIP_NAME.zip"
195         /bin/rm -f $TMPDIR/$SUBDIR/result
196         cd "$TMPDIR/$SUBDIR"
197         /usr/bin/zip ../$END_ZIP_NAME $( ls * | grep -vi "
            output.log" )
198         echo -n "$END_ZIP_NAME" > $TMPDIR/result
199     fi
200     echo "Cleaning ..."
201     /bin/rm -fr "$TMPDIR/$SUBDIR"
202 fi

```

Appendice B

Acronimi

AOI Area Of Interest

API Application Programming Interface

AT Acceptance Test

AVHRR Advanced Very High Resolution Radiometer

BT Brightness Temperature

CEOS Committee on Earth Observation Satellites

CNES Centre National d'Etudes Spatiales

DIMAP DIgital IMAge maP

ESA European Space Agency

DN Digital Number

ETM Enhanced Thematic Mapper

ETM+ Enhanced Thematic Mapper Plus

ECSS European Cooperation For Space Standardization

GDAL Geospatial Data Abstraction Library

GLCF Global Land Cover Facility

HRG High Resolution Geometric

HRVIR High Resolution in the Visible and Infra-Red

IDL Integrated Numerical Analysis & Data Visualization Solution

IFD Image File Directory

IT Integration Test

KEI KIM Extensions and Installation

KIM Knowledge based Information Mining

LIRCC Landsat Image Radiometric Conversion and Correction

LISC Landsat Image Spectral Categorization

LL Low Left

LR Low Right

MEEO Meteorological and Environmental Earth Observation

MODIS Moderate Resolution Imaging Spectroradiometer

MWIR Mid-wavelength InfraRed

NIR Near InfraRed

SSE Service Support Environment

SWIR Short-wavelength InfraRed

TIFF Tagged Image File Format

TIR Thermal InfraRed

TM Thematic Mapper

TOA Top Of Atmosphere

UL Upper Left

UR Upper Right

USGS United States Geological Survey

UT Unit Test

UTM Universal Transverse Mercator

XML eXtensible Markup Language

VIS Visible

VVTP Verification and Validation Test Plan

WRS-2 World Reference System

Bibliografia

- [1] Dimap Reference Documentation. Website.
<http://www.spotimage.fr/dimap/spec/documentation/refdoc.htm>
- [2] The SPOT satellites. Website.
http://http://www.spot.com/html/SICORP/_401_444_445_.php
- [3] LIBGEOTIFF. Website, Marzo 2001.
<http://remotesensing.org/geotiff/api/index.html>
- [4] Landsat TM/ETM+ CEOS/ESA Products. Rap. tecn., ESA, Dicembre 2005.
http://earth.esa.int/includes/resources/dsp_DocDetailsPopUp.cfm?fobjectid=4437
- [5] SPOT satellite technical data - General features and instruments. Website, 2005.
http://www.spotimage.fr/html/_167_224_229_.php
- [6] Using The TIFF Library. Website, Dicembre 2005.
<http://www.remotesensing.org/libtiff/libtiff.html>
- [7] Enhanced Thematic Mapper (ETM+). Website, Agosto 2006.
<http://edc.usgs.gov/products/satellite/landsat7.html>
- [8] Landsat 5 IGS Documentation. Website, Giugno 2006.
http://landsat.usgs.gov/resources/l_igs_docs.php
- [9] Landsat 7 IGS Documentation. Website, Luglio 2006.
http://landsat.usgs.gov/resources/l7_igs_docs.php

- [10] Thematic Mapper (TM). Website, Agosto 2006.
<http://edc.usgs.gov/products/satellite/tm.html>.
- [11] GDAL - Geospatial Data Abstraction Library. Website, Aprile 2007.
<http://www.gdal.org/index.html>.
- [12] List of agencies in CEOS. Website, Marzo 2007.
<http://www.ceos.org/pages/agencies.html>.
- [13] Naming Convention Guide. Website, 2007.
<http://glcf.umiacs.umd.edu/data/guide/namingconvention/index.shtml>.
- [14] S. Mantovani e S. Natali. KEI-MEEO Software Verification, Validation and Unit/Integration Test Plans. Rap. tecn., MEE0, Giugno 2007.