Sistemi multimediali: Video/Image Compositing

Autore: Luca Banzato
Corso: Sistemi Multimediali
Prof. Giorgio Leonardi
Luglio 2017

1. Descrizione del problema

Il compito di questo progetto è di realizzare una libreria che contiene al suo interno l'implementazione di alcuni metodi per la gestione del canale alfa su immagini RGB standard e di alcune applicazioni interessanti legate al mondo cinematografico quali la sostituzione di determinate parti di immagini contrassegnate da uno specifico colore con una immagine di sfondo differente.

Le immagini di campione sono state prelevate dalla rete. Si è scelto di considerare anche immagini con formati differenti l'una dall'altra e aventi un numero differente di canali.

2. Lista degli acronimi e abbreviazioni

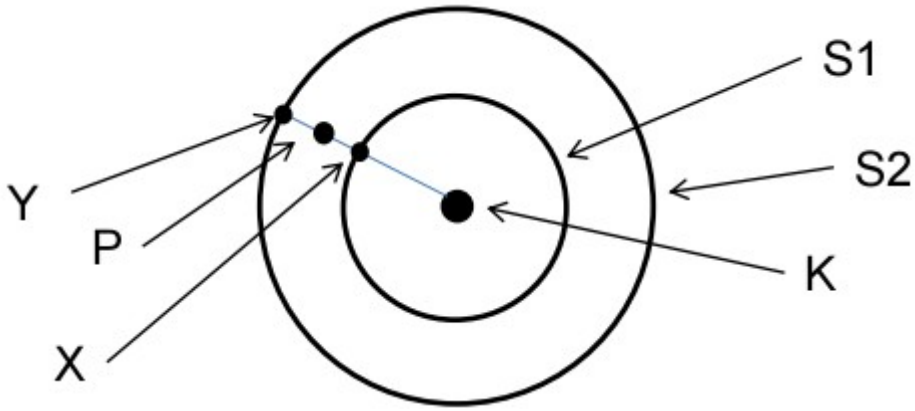
A seguire sono elencate le differenti abbreviazioni utilizzate in queste note tecniche:

- RGB: color space basato sulla variazione di intensità dei colori Rosso, Verde, Blu
- HDR: High Dynamic Range

3. Assunzioni teoretiche e formulazioni

La libreria supporta al momento solo operazioni con esattamente due immagini della stessa dimensione. Nel caso in cui le immagini abbiano una risoluzione differente, verrà segnalato un errore mediante un'eccezione Java.
E' invece consentito elaborare con immagini di formato differente (sono supportati tutti i tipi di immagine che Java può gestire) e con formato Grayscale o RGB, salvo in alcuni casi in cui sarà necessario utilizzare solo quest'ultimo.
Nel progetto vengono utilizzate le seguenti idee e algoritmi:

- Alpha Blending: utilizzato per comporre immagini fra loro con effetto trasparenza utilizzando tutti i pixel a disposizione. Date due immagini F= (fr , fg , fb , αf) e B= (br , bg , bb , αb) definite nel color space RGBA, è possibile definire ciascun pixel dell'immagine risultante C come C[i] = (αf F[i] + (1- αf) αb B[i]) / α', con α'= αf + (1- αf) αb
- Chroma Key: è l'operazione di sostituzione dei pixel di un'immagine F, aventi una determinata tonalità di colore, con quelli di una immagine B. E' possibile in generale definire ciascun pixel dell'immagine risultante C come C[i] =α F[i] + (1- α) B[i], con α= 0 per i pixel corrispondenti alla tonalità di colore verde o blu e α>0 per gli altri pixel.
- Algoritmo di Vlahos: un metodo più specifico per il Chroma Keying. Si è osservato sperimentalmente che i risultati migliori sono raggiunti quando il valore α è calcolato basandosi sull'intensità della componente blu o verde del pixel rispetto alle altre due.
  Si applica la formula di sostituzione generale per il Chroma Key, con α = 1 − (Fb − MAX(Fr , Fg)).
- Algoritmo Primatte: alternativa a Vlahos per il calcolo del valore α. Tutti i pixel vengono rappresentati in uno spazio tridimensionale, i cui piani rappresentano ciascuno una componente del color space RGB. Vengono definiti cinque elementi: una immagine F, una immagine B, un colore chiave K, una sfera chiave S1 e una sfera di tolleranza S2, entrambe con centro K e raggio definito dall'utente.
  Rappresentando ciascun pixel come un punto nello spazio tridimensionale, tutti i pixel di F situati all'interno di S1 vengono impostati come invisibili (α=0), al di fuori di S2 visibili (α=1) e quelli compresi tra S1 e S2 semi-trasparenti (α=|XP|/|XY|). In modo duale è possibile definire il comportamento dei pixel di B.



3.1 - Illustrazione caso pixel semitrasparenti

4. Descrizione della libreria

4.1 – Alpha Blending

La classe AlphaBlending è stata realizzata seguendo uno schema molto semplice, prevedendo la presenza di un solo costruttore che prende in input due oggetti BufferedImage e un terzo valore double rappresentante il valore di opacità della prima immagine passata in input al costruttore. Il valore dell'immagine di background è calcolabile eseguendo il reciproco (1- α).
Il metodo blend esegue l'operazione riportata nel seguente pseudocodice:

per ogni canale dell'immagine
    per ogni riga del raster
        per ogni colonna del raster
            Precalcolo il reciproco di αf, definito con 1-αf
            Raster[riga][colonna] = (αf F + (1- αf) αb B) / α'

Con:

- α'= αf + (1- αf) αb,
- αf = valore di opacità α della prima immagine
- αb = valore di opacità α della seconda immagine
- F e B = Valore del pixel in posizione riga-colonna dei raster delle due immagini

4.2 – Chroma Keying (Vlahos)

Il costruttore della classe prende in input due oggetti BufferedImage, rappresentanti l'immagine di foreground e background, e un terzo parametro di tipo intero che funge da switch per decidere il tipo di colore chiave (verde o blu) che dovrà essere sostituito con all'interno della prima immagine. E' richiesto ovviamente che l'immagine di foreground sia una immagine in colorspace RGB e non Grayscale e nel caso in cui non venga rispettato questo vincolo, una eccezione è sollevata.
Per facilitare la specifica di questo intero da parte dell'utente, senza che sia a conoscenza di quale valore sia associato al verde o al blu, sono forniti due valori statici KEYCOLOR_BLUE e KEYCOLOR_GREEN.

Di seguito lo pseudocodice seguito per l'implementazione del metodo replace, che esegue l'operazione di chroma keying:

per ogni riga del raster
    per ogni colonna del raster
        precalcolo α
        per ogni canale dell'immagine
            Raster[riga][colonna] = α F + (1- α) B

Con:

- Fr, Fg e Fb rappresentanti il valore del pixel nel canale Rosso, Verde o Blu
- α = 1 − ((Fb − MAX(Fr , Fg))/255) nel caso in cui il colore chiave sia il blu o
  α = 1 − ((Fg − MAX(Fr , Fb))/255) nel caso in cui sia il verde
- F e B = Valore del pixel in posizione riga-colonna dei raster delle due immagini

Per evitare di scrivere due casi separati per il calcolo di alfa in base al colore chiave, è stato impostato KEYCOLOR_BLUE a 2 e KEYCOLOR_GREEN a 1.
In questo modo quando è necessario specificare il valore del pixel del canale del colore chiave, si inserisce direttamente il valore (0 Rosso, 1 Verde, 2 Blu), mentre per l'altro è specificato dalla formula 3 − valore canale colore chiave (Se ad esempio si è scelto il blu, si ottiene il verde (3-2 = 1) e viceversa.
Per questioni di efficienza è stato deciso di visitare subito i tre canali RGB di ciascun pixel anziché eseguire una visita completa per ciascuno di essi. Questo per evitare che il valore alfa sia calcolato inutilmente tutte le volte che si esplora un livello, essendo uguale in tutti per il pixel nella stessa posizione.

4.3 – Chroma Keying (Primatte)

Seguendo lo scheletro impostato per la creazione della classe che implementa il ChromaKeying basato sull'algoritmo di Vlahos, al costruttore sono aggiunti due parametri indicanti i raggi delle sfere da utilizzare per Primatte.
Come punto centrale della sfera è impostato il colore chiave specificato dai medesimi interi statici KEYCOLOR_BLUE e KEYCOLOR_GREEN, a cui corrispondono rispettivamente i punti tridimensionali [0, 0, 255] e [0, 255, 0] e che possono essere modificati in base alle proprie esigenze.
Analogo è il metodo replace, con però un differente approccio per la definizione di alfa, specifico di questo algoritmo.

per ogni riga del raster
    per ogni colonna del raster
        precalcolo α con la funzione alphaCircles
        per ogni canale dell'immagine
            Raster[riga][colonna] = α F + (1- α) B

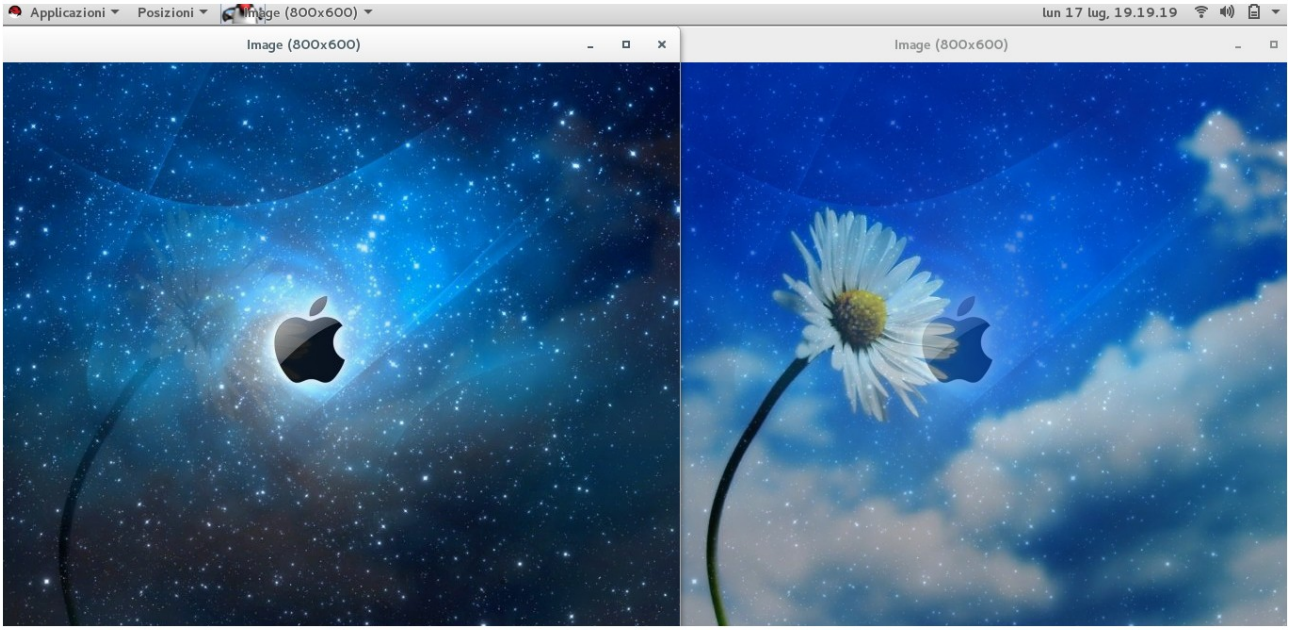
La funzione alphaCircles prende in input due punti tridimensionali corrispondenti al centro della sfera (il colore chiave) e a un punto definito dai valori dei canali RGB del pixel che si sta correntemente verificando.

Utilizzati insieme al raggio delle sfere, consentono di determinare, mediante l'utilizzo della distanza Euclidea su tre dimensioni, se il secondo punto è contenuto o meno in una delle due sfere con centro il primo punto e aventi raggi specificati dall'utente nel costruttore. In base al risultato dato da questa verifica e dalle specifiche di Primatte, viene ritornato un apposito valore alfa.

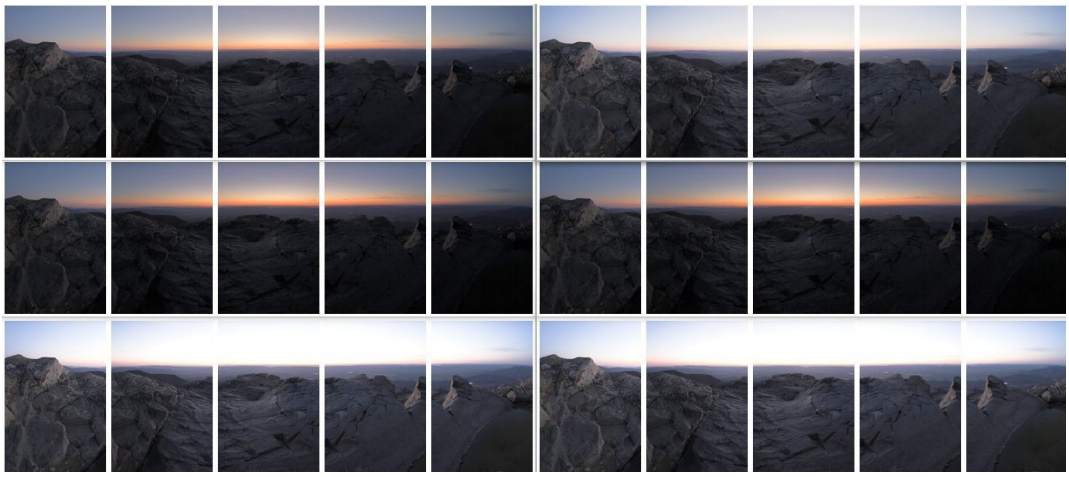
5. Descrizione dei test e risultati

Sono stati eseguiti i seguenti test:

- AlphaBlending: si è provato ad eseguire il blending tradizionale con due singole immagini ed è stato ottenuto per tutti i test svolti il risultato ottenuto. Le immagini considerate possiedono dimensioni identiche, come stabilito durante la fase di progettazione.



5.1 - Coppia di immagini con applicazione del valore di alfa uguale a 0.7 ad entrambe



5.2 - Utilizzo dell'Alpha Blending per l'High Dynamic Range (Foto di base: DSLRBuzz)

Fra i vari esperimenti condotti viene presentato in questa relazione quello riguardante due immagini utilizzate per l'alpha blending. A sinistra è stato utilizzato un valore di alfa uguale a 0.7 per la prima delle due immagini utilizzate come base, mentre a destra il valore di alfa viene applicato alla seconda immagine di base. Viene dato per assunto che la somma del valore di alfa delle due immagini sua uguale a 1. Un utilizzo più interessante dell'alpha blending è la combinazione di immagini identiche fra loro, ma con un'esposizione differente, per simulare la tecnica di High Dynamic Range che ha raggiunto un notevole successo negli ultimi anni.
L'obiettivo di questa tecnica è di ottenere immagini con luminosità e contrasto bilanciati partendo da due o più immagini sottoesposte e sovraesposte.
Nell'immagine 5.2 vengono presentati i risultati (in prima fila) dell'applicazione di differenti livelli di alpha alle immagini di base (sottoesposta in seconda fila, sovraesposta in terza fila). Per l'immagine risultante a sinistra sono stati utilizzati valori di alpha pari a 0.6 e 0.4, mentre per quella a destra valori pari a 0.2 e 0.8.

- CromaKeying (Vlahos): la prima versione del CromaKeying risulta essere molto semplice da implementare, ma alcuni dettagli dell'output possono non essere perfetti per via delle formule utilizzate in Vlahos.
  Caricando i files di esempio Vlahos1.jpg e Vlahos2.jpg presenti nella directory delle immagini campione, si ottiene l'immagine 5.3; nonostante si ottenga un buon risultato a livello globale, persistono alcune tracce del green screen nelle zone di contorno del personaggio inserito in primo piano.
  In determinate situazioni si è ottenuta un'output del tutto inaspettato, come quello presentato nell'immagine 5.4; il tutto è dovuto al fatto che il verde utilizzato per il green screen nella prima immagine non è nel color space RGB quello rappresentato dal valore (0, 255, 0). Per questo motivo sarebbe opportuno sostituire il colore del green screen, mediante un editor di immagini quale GIMP, con il verde effettivo al fine di ottenere un buon risultato con Vlahos.

5.3 - Esempio di Chroma Keying con l'algoritmo di Vlahos

5.4 - Vlahos può presentare anomalie con green o blue screen errati

- CromaKeying (Primatte): questa tecnica produce risultati migliori a discapito della complessità dell'algoritmo utilizzato. Prendendo in considerazione le immagini utilizzate in precedenza con l'algoritmo di Vlahos, si può notare come la qualità dell'immagine sia nettamente migliore, con la totale assenza di contorni verdi nell'immagine 5.5.
  Anche il caso peggiore ottenuto con Vlahos (immagine 5.4) subisce un drastico miglioramento (immagine 5.6), con la totale assenza di residui del green screen sullo sfondo; tuttavia continuano a persistere alcune zone segnate dal colore verde sul soggetto in primo piano. E' possibile migliorare il risultato modificando i raggi delle sfere utilizzate in Primatte, ma la qualità risultante deve essere valutata volta per volta, in quanto potrebbero essere rimosse o alterate sezioni da preservare (immagine 5.7).

5.5 - Risutato della combinazione di Vlahos1.jpg e Vlahos2.jpg con Primatte

5.6 - Vlahos2.jpg e Vlahos3.jpg con Primatte (r1=170, r2=200)

5.7 - Vlahos2.jpg e Vlahos3.jpg con Primatte (r1=200, r2=235)

6. Conclusioni

Il progetto si occupa di fornire allo sviluppatore una libreria con le tre implementazioni (Blending, Vlahos, Primatte). Nell'approccio seguito si è provato a realizzare un'implementazione con parti di codice ottimizzati, in quanto il processamento delle immagini di piccole dimensioni può essere accettabile in termini di complessità temporale, mentre risulta molto più oneroso per immagini con una dimensione più importante come la risoluzione 4K.

Per l'alpha blending è stato osservato come ottenere risultati interessanti che non siano solo riguardanti la banale sovrapposizione di due immagini, ma che permettano anche di correggere una immagine troppo sovraesposta/sottoesposta sfruttando più copie di questa.

Vlahos e Primatte sono risultati molto interessanti, sia per i concetti alla base di ciascun algoritmo, sia per le performance ottenute in termini di tempo e qualità dei risultati.
Se si vogliono ottenere immagini con la parte del green/blue screen sostituito con poche iterazioni da parte dell'utente, Vlahos si rileva più che sufficiente. L'aspetto negativo è la possibile presenza di tracce del green screen che delimitano i contorni della figura posta in primo piano e la quasi completa insensibilità alla sostituzione di tonalità di verde e blu leggermente differenti rispetto ai valori RGB (0, 255, 0) e (0, 0, 255).
Primatte si è invece dimostrato molto valido per il compito per cui è stato progettato ed opera bene sia nella rilevazione delle tonalità di verde e blu da sostituire, sia nella corretta rimozione delle tracce prossime ai contorni presenti con l'applicazione dell'algoritmo di Vlahos. L'aspetto negativo si osserva nella necessità da parte dell'utente di alterare secondo le proprie esigenze alcuni parametri utilizzati dall'algoritmo e nel costo computazionale leggermente superiore rispetto a Vlahos.

Il progetto è stato complessivamente molto utile per comprendere gli argomenti studiati durante il corso, oltre a realizzare qualcosa di concreto e imparare alcuni trucchi di ottimizzazione che sicuramente torneranno utili in un futuro prossimo.

7. Sviluppi futuri

La libreria al momento è funzionante e produce buoni risultati. Tuttavia sono stati omessi alcuni punti e sarebbe interessante potere implementarli in futuro:

1. Al momento si prevede l'utilizzo di due immagini per operazione; sarebbe interessante potere introdurre la possibilità di aggiungere un numero maggiore di immagini per ottenere risultati migliori, in particolare per l'operazione di blending.
2. Si può operare con immagini aventi la stessa dimensione; questo vincolo andrebbe rimosso e sarebbe richiesto a priori lo studio delle varie casistiche che possono presentarsi (ad esempio l'immagine da sostituire al green/blue screen ha dimensioni inferiori).
3. Nel progetto sono presenti tre programmi dimostrativi basati su interfaccia a carattere e i parametri possono essere modificati tramite un file di configurazione. Sarebbe opportuno introdurre un'interfaccia grafica per semplificare notevolmente l'utilizzo di questi applicativi, specie se l'utilizzatore non è un esperto del settore informatico.