Università degli studi di Modena e Reggio Emilia

Dipartimento di Ingegneria “Enzo Ferrari”

Corso di Laurea Magistrale in Ingegneria Informatica



Bird's Eye View Transformation

pROGETTO DI CORSO

Fabrizio Di Blasi | Informatica industriale ed IoT | a.a 2018/2019

**Sommario**

[Introduzione 3](#__RefHeading___Toc1208_2846315957)

[a. Finalità del progetto 3](#__RefHeading___Toc1210_2846315957)

[b. Stato dell’arte 3](#__RefHeading___Toc211_3904458324)

[Descrizione 5](#__RefHeading___Toc213_3904458324)

[a. Schema a Blocchi 5](#__RefHeading___Toc1216_2846315957)

# Introduzione

# Finalità del progetto

In questa relazione viene descritto il procedimento per effettuare la trasformazione “Bird's Eye View”. Tale trasformazione consiste nel ruotare, secondo dei parametri impostati dall’utente, ogni singolo fotogramma di un video già preesistente, oppure catturato dalla webcam (o videocamera).

In questo modo si ha la possibilità di interpolare e proiettare il video catturato secondo un’angolatura differente da quella realmente ripresa.

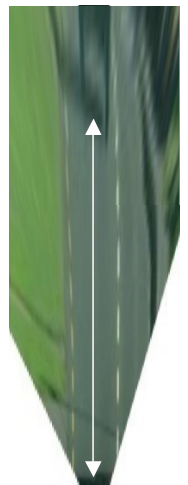
L’utente ha la possibilità di effettuare la trasposizione dei pixel tramite:

* Funzioni già esistenti di OpenCV
* Funzioni accelerate in CUDA

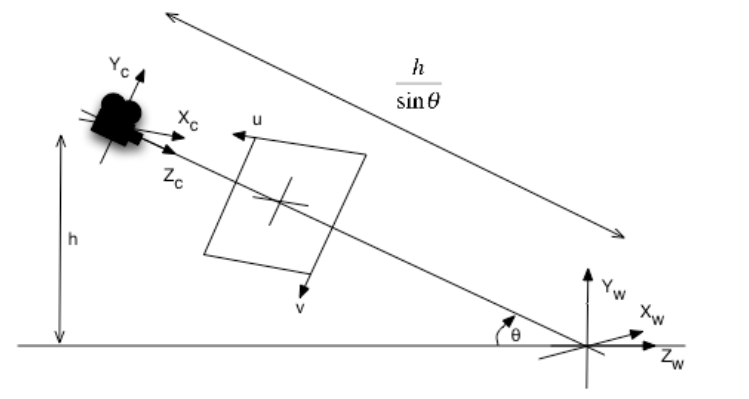
Tale trasposizione deve essere effettuata nel più beve tempo possibile in modo da non creare troppo ritardo, ed un calo degli FPS. Dopo una introduzione pratica del lavoro, verranno esposti i risultati sperimentali.

# Stato dell’arte

La “Bird's Eye View” fa parte della Inverse Perspective Mapping (IPM) che consiste in una tecnica matematica in cui le coordinate di un sistema vengono trasformate da una prospettiva all’altra. La IPM può essere usata in ambito automotive per ottenere una visione top-down, ad esempio :



Dalle due figure si può notare lo scopo principale, ovverosia, la proiezione di un fotogramma di una dashcam secondo una visuale top-down. Per creare tale visualizzazione, si ha il bisogno di trovare il corretto mapping di un punto della superficie (x,y,z) con il piano (u,v), noto l’angolo θ.

La riproiezione può essere calcolata tramite la seguente formula:

(u,v,1)T = K T R(x,y,z,1)T

dove:

R : matrice di rotazione

T : matrice di traslazione

K : matrice contenente i parametri della fotocamera

R = T = K =

Tramite le seguenti matrici, è possibile calcolare il mapping top down di ciascun singolo pixel, infatti, data *f*, distanza focale della fotocamera, ed *s,* parametro di distorsione della camera, l’equazione :

(u,v,1)T = K T R ( x, y, z, 1)T

l’equazione può essere riscritta come :

essendo interessati al piano della strada (Yw = 0), la precedente formula diventa:

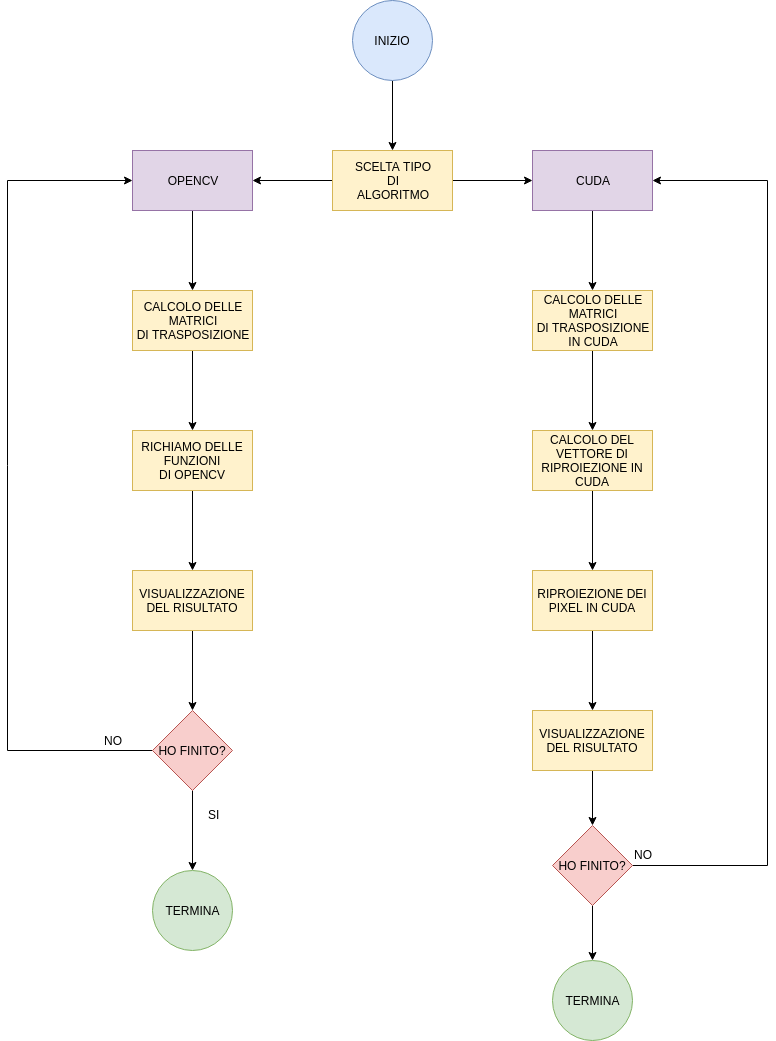
Grazie a quest’ultima equazione è possibile ottenere il mapping top down desiderato.

Visto il numero elevato di prodotti matriciali, si è scelto di parallelizzare il più possibile il calcolo matriciale tramite l’uso di kernel CUDA.

Inoltre questo procedimento di traslazione deve essere effettuato su ogni singolo canale dell’immagine RGB

# Descrizione

# Schema a Blocchi

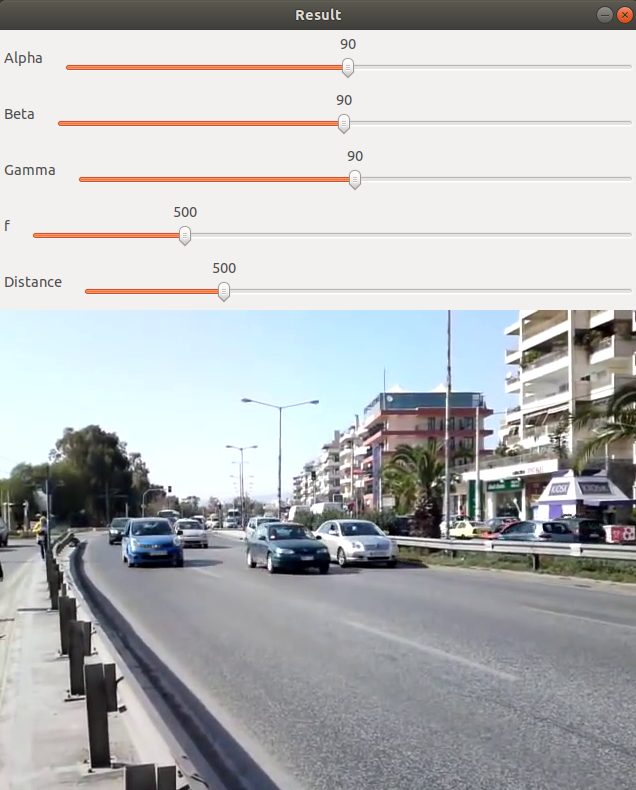


Dallo schema a blocchi presentato è possibile capire le due differenti modalità di esecuzione, selezionabili da linea di comando, del programma e le varie fasi svolte da esso.

Il funzionamento logico dei due rami è pressoché lo stesso, ma, usando il procedimento in CUDA, si ha la problematica di calcolare manualmente le posizioni dei nuovi pixel, per poi effettuare la trasposizione su tutti i 3 canali RGB. Per la memorizzazione del fotogramma sul device sono stati utilizzati degli oggetti definiti da OpenCV, che vanno direttamente ad allocare della memoria nella GPU, manipolabile tramite un kernel CUDA.

# Guida per L’utente e Codice Sorgente

***Interfaccia:***

La schermata iniziale è la seguente, l’utente ha a disposizione 5 regolazioni, le prime 3 sono per definire l’angolo con il quale ruotare il fotogramma, mentre gli ultimi due servono ad impostare i parametri della fotocamera.

Tale interfaccia è identica per entrambi i “rami” dell’algoritmo

il programma viene eseguito da linea di comando e si hanno 4 opzioni differenti, in base se si vuole l'algoritmo con solamente chiamate di OpenCV oppure se accelerato in cuda, e se si vuole analizzare lo stream della webcam oppure un video già preesistente.

Se si vuole usare l’accelerazione in CUDA :

$ ./app y

$ /app y <video path>

Se si vogliono usare solamente direttive di OpenCV:

$ ./app n

$ ./app n <video path>

***Main:***