

Programma di tracciamento del pendolo di Foucault Manuale utente

Raffaelli Claudia, Scandiffio Francesco

Dipartimento di Ingegneria dell'Informazione, Università di
Firenze

24 settembre 2020

Indice

1	Introduzione	2
2	Esecuzione ed utilizzo	2
2.1	Modalità di tracciamento	3
2.1.1	Avvio del tracciamento	5
2.1.2	Sostituzione del Template	7
2.1.3	Grafico 2D real-time	7
2.2	Calibrazione manuale	8
2.2.1	Procedura di calibrazione	9
2.3	Grafico 2D da file	14
A	Compilare il programma	19
B	Appendice Matematica	20
B.1	Modifica della prospettiva	20
B.2	Metodo di tracciamento	22

1 Introduzione

Questo documento fornisce all'utente tutte le informazioni necessarie per utilizzare il programma di tracciamento di un pendolo di Foucault. L'applicativo distribuito permette l'estrazione ed il salvataggio su file **CSV** delle coordinate del centro del pendolo, offrendo al contempo la possibilità di visualizzare un grafico del moto in tempo reale così da poter effettuare un'immediata analisi qualitativa. La modalità di visualizzazione del grafico può essere utilizzata anche in versione non real-time fornendo al programma un file **CSV** di una precedente esecuzione.

Data l'impossibilità di effettuare una calibrazione della videocamera (causa lockdown per Covid19), il programma offre anche una modalità di *calibrazione* con cui è possibile correggere gli effetti introdotti dall'inclinazione della videocamera, come ad esempio l'effetto parallasse.

Infine, l'appendice contiene un tutorial su come poter compilare nuovamente il programma ed una trattazione matematica relativa alla correzione della prospettiva e al metodo di tracciamento.

2 Esecuzione ed utilizzo

Il programma per il tracciamento si presenta come un eseguibile che è possibile lanciare attraverso il Terminale. Per prima cosa è necessario raggiungere la cartella all'interno della quale è contenuto il file. Aprire dunque il Terminale e digitare al suo interno il seguente comando:

```
cd /home/pi/Desktop/release
```

Il percorso indicato è quello corrispondente alla cartella contenente il programma così come fornito. Da questo momento in poi indicheremo con il termine **HOME** tale percorso. Successivamente ad eventuali spostamenti del file stesso, **HOME** dovrà essere variato di conseguenza.

Una volta raggiunta la cartella contenente l'eseguibile, il programma può essere avviato in modalità di tracciamento eseguendo nel Terminale il comando:

```
./tracciamentoPendolo
```

Il comando di avvio appena citato ammette l'utilizzo di parametri opzionali, ognuno dei quali consente all'utente di avviare il programma in una determinata modalità. La Tabella 1 contiene la lista completa degli argomenti opzionali insieme ad una breve descrizione dell'effetto prodotto. Le singole modalità di esecuzione sono spiegate più nel dettaglio nel resto del manuale.

Argomento	Dettagli
<code>-c</code> oppure <code>-calibrate</code>	Usa questo comando per specificare l'esecuzione del programma in modalità di calibrazione.
<code>-g</code> oppure <code>-graph</code>	Avvia il programma in modalità offline, consentendo di visualizzare il grafico degli spostamenti registrati su un file CSV in una precedente esecuzione.
<code>-h</code> oppure <code>-help</code>	Mostra una breve spiegazione circa l'utilizzo degli argomenti a linea di comando

Tabella 1: Argomenti a linea di comando

A titolo esemplificativo circa l'utilizzo dei parametri opzionali, supponiamo di voler visualizzare la schermata di help. Sarà sufficiente scrivere nel Terminale:

```
./tracciamentoPendolo -h
```

2.1 Modalità di tracciamento

La modalità di esecuzione principale del programma è sicuramente quella relativa al tracciamento del pendolo. Per tracciare il pendolo viene utilizzato il metodo del **Template Matching**. Questa tecnica consiste nel confrontare ogni frame estratto dalla videocamera con un'immagine di esempio (**Template**) con lo scopo di individuare la zona dell'immagine in Input che più coincide al contenuto del Template. Risulta evidente come la scelta del Template costituisca un elemento importante ai fini della buona riuscita del tracciamento. Insieme al programma viene fornito il Template in Figura 1 che contiene il centro del pendolo insieme ad una porzione di sfera come contorno. Nel caso di variazioni del centro del pendolo, soprattutto cromatiche,



Figura 1: Template utilizzato nella funzione di tracciamento del pendolo.

tale immagine dovrà essere ovviamente sostituita in modo opportuno utilizzando la procedura illustrata nella sezione 2.1.2. Ulteriori informazioni sul funzionamento del Template Matching sono contenute in Appendice B.2.

Come è noto, lo scopo del programma è quello di fornire all'utente un documento che contenga le coordinate del centro del pendolo in funzione del tempo, così da poterne studiare il moto. Per questo motivo le coordinate del pendolo vengono salvate su un file **CSV**. All'avvio del programma viene richiesto all'utente se desidera che venga applicata la correzione della prospettiva. A seconda della risposta fornita, distinguiamo tra i seguenti casi:

- Se l'utente rifiuta, il programma genera **un solo** file **CSV** dal nome `anno_mese_giorno_ore_minuti_N`. Questo file contiene le coordinate estratte dall'immagine **senza** la correzione prospettica, motivo per cui è presente la lettera **N** nel nome.
- Se l'utente accetta, il programma genera **due** files **CSV**. Il primo contiene le coordinate originali, così come al punto precedente. Il secondo file invece ha nome `anno_mese_giorno_ore_minuti_Y` ed è quello in cui vengono memorizzate le coordinate con la correzione prospettica.
N.B. Prima di avviare la correzione prospettica è necessario seguire l'operazione di correzione manuale illustrata nel Paragrafo 2.2.

I **CSV** sono reperibili all'interno della cartella `PendulumCsv` posizionata in `Desktop/ProjectFolder`.

Ogni **CSV** contiene tante righe (record) quante sono i frame processati e possiede la seguente struttura:

`Time;x;y`

- **Time**. Istante di tempo che indica il momento di estrazione della coordinata associata alla riga in questione. Tale istante di tempo è espresso

in secondi ed ha una precisione di 6 cifre decimali per un'elevata accuratezza. Questa misura è espressa in modo relativo, ovvero indica quanto tempo è passato a partire dalla prima rilevazione che viene fatta coincidere con l'istante $t = 0$.

- **x.** Componente **x** (in pixel interi) del centro del pendolo.
L'origine dell'asse si trova nella parte sinistra dell'immagine.
- **y.** Componente **y** (in pixel interi) del centro del pendolo.
L'origine dell'asse si trova nell'**angolo in alto a sinistra** dell'immagine.

Come è possibile notare, il separatore di campi utilizzato è il simbolo di **punto e virgola**.

N.B. Avere l'origine del sistema di riferimento dei pixel nell'angolo in alto a sinistra dell'immagine significa che ad esempio il punto $P(x, y)$ si trova a $20px$ di distanza dal bordo sinistro ed a $50px$ di distanza dal margine superiore del frame. Questo può sembrare contro intuitivo ma è necessario per avere consistenza dei dati tra la versione con prospettiva e quella senza.

2.1.1 Avvio del tracciamento

Per avviare il programma in modalità di estrazione delle coordinate è sufficiente aprire un terminale, usarlo per recarsi nella cartella **HOME** ed eseguire il comando di avvio senza alcun parametro aggiuntivo:

```
./tracciamentoPendolo
```

Appena avviato, il programma porrà all'utente la seguente domanda:

Si desidera applicare la correzione della prospettiva? [y/n]

La risposta deve essere scritta sul terminale e confermata premendo il tasto **Invio**. Come spiegato poco sopra, sono possibili due scenari:

- L'utente digita **n**. Viene creato un solo file senza correzione prospettica.
- L'utente digita **y**. Vengono creati due file, uno senza correzione ed uno con.

Ricordiamo che prima di avviare la modalità con correzione prospettica è necessario seguire la procedura illustrata nel Paragrafo 2.2.

Dopo aver risposto alla domanda si aprirà una finestra (Figura 2) in cui vengono mostrate le immagini così come sono estratte dalla videocamera, ovvero senza che alcuna correzione prospettica sia stata applicata. Per fornire

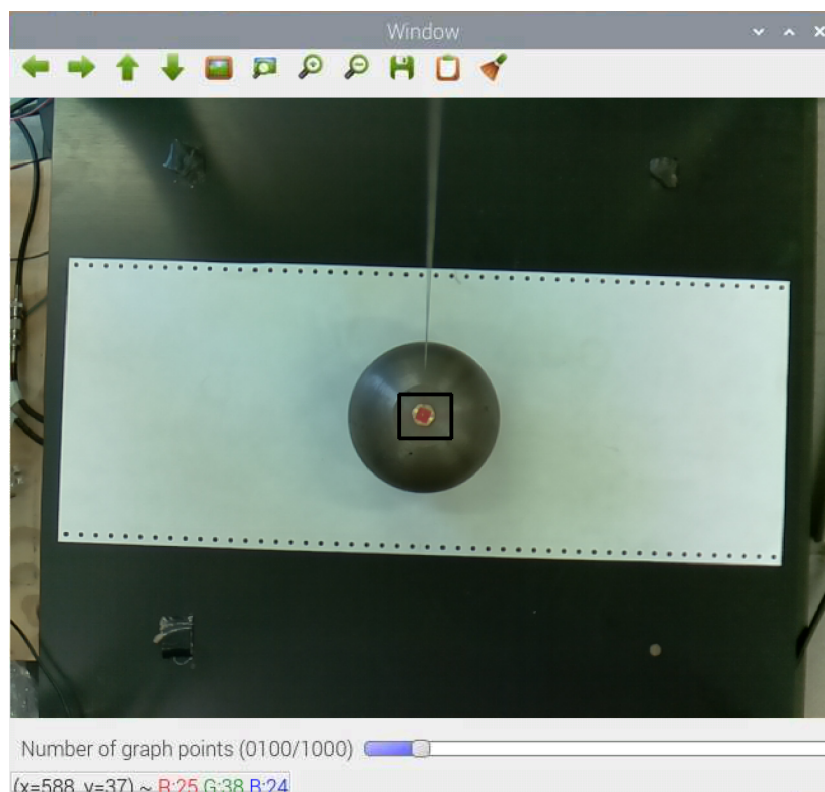


Figura 2: Finestra con il tracciamento del pendolo.

un feedback del corretto riconoscimento del centro del pendolo viene inoltre disegnato un rettangolo nero che circonda la regione dell'immagine che risulta più simile al Template.

Una volta avviato, il programma può essere lasciato in esecuzione per tutto il tempo che si desidera. Per interrompere l'esecuzione è sufficiente digitare la combinazione di tasti **Ctrl + c** all'interno dello stesso terminale in cui è stato eseguito il comando di avvio.

2.1.2 Sostituzione del Template

Può sempre verificarsi la necessità di sostituire l'immagine del Template. Il programma in questione non implementa alcun tipo di preprocessing o normalizzazione né sulle immagini da analizzare né sul Template. Questo vuol dire che l'utente ha il compito di fornire al sistema un template contenente un'immagine dell'oggetto da individuare il più simile possibile all'aspetto che esso possiede nelle immagini da analizzare. Elenchiamo dunque alcuni punti da tenere in considerazione durante la scelta del nuovo template.

- Il template deve essere codificato nello stesso **spazio colori** dell'immagine (es. un frame) su cui si vuole effettuare il tracciamento. Questo vuol dire che se nel momento del confronto una delle due immagini è ad esempio codificata in scala di grigi, allora anche l'altra deve esserlo.
- Le **dimensioni** dell'oggetto da ricercare (in pixel) devono essere simili nelle due immagini.
- L'**orientazione** dell'oggetto deve essere identica nelle due immagini.
- Le dimensioni del Template non possono essere più grandi di quelle dell'immagine da analizzare.

Un metodo rapido ed efficace per rispettare i punti appena illustrati consiste nel ricavare il template ritagliando l'oggetto da individuare direttamente dall'immagine che si vuole analizzare. Ad esempio, il template fornito col programma è stato ottenuto ritagliando un frame raffigurante il pendolo in situazione di riposo.

Per modificare il Template basta recarsi nella cartella del programma e sostituire il file `template.png`. Il programma attende un file con esattamente questo nome e formato, dunque è fondamentale che anche il nuovo file sia in formato `png`. Una modifica del nome e formato utilizzati dal programma è possibile solo cambiando il codice e ricompilando i files. Per ulteriori informazioni circa la compilazione vedere l'appendice A.

2.1.3 Grafico 2D real-time

In alternativa alle immagini della videocamera è possibile visualizzare un grafico real-time delle coordinate estratte. Sottolineiamo che i punti disegnati su questo grafico sono quelli **senza** la correzione della prospettiva. In questo

modo i punti salvati risultano perfettamente in linea con il feedback fornito dal templateMatching sui frames estratti dalla videocamera.

Per attivare la visualizzazione del **grafico realtime** possiamo osservare la Figura 3. Iniziamo facendo click sul pennello nella barra in alto della scher-



Figura 3: Strumenti per visualizzare e controllare il grafico real-time.

mata (simbolo 1). Si aprirà una piccola finestra contenente la checkbox **Show graph** (simbolo 2) con cui mostrare il grafico. Essendo un grafico real-time è possibile mostrare solo un numero limitato di punti. Il numero di punti da mostrare è regolato tramite la trackbar posta in basso (simbolo 3): i valori disponibili sono nel range $[0, 1000]$, dove il numero di default è 100. Man mano che il tempo avanza i nuovi punti vengono aggiunti mentre i meno recenti sono rimossi, donando un effetto dissolvenza alle linee tracciate. Per ritornare a visualizzare le immagini della videocamera è sufficiente premere nuovamente la checkbox.

2.2 Calibrazione manuale

Prima di poter utilizzare la correzione prospettica nella modalità di tracciamento è necessario effettuare la correzione manuale dell'inclinazione della fotocamera. Una fotocamera inclinata produce infatti l'effetto parallasse, ovvero gli oggetti sono percepiti in una posizione diversa rispetto a quella che si noterebbe osservando la scena con lo sguardo perpendicolare al piano su cui giace l'oggetto. Il modo migliore per correggere questa distorsione è quello di calibrare la fotocamera utilizzando uno dei numerosi algoritmi appositamente ideati tra quelli disponibili in rete¹. Calibrare la videocamera

¹Ad esempio il tutorial fornito da OpenCV disponibile al seguente link:
https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html

è un processo lungo che deve essere necessariamente svolto in presenza ma ha il pregio di correggere tutte le distorsioni presenti nell'immagine. Data l'impossibilità di effettuare l'operazione di Calibrazione fisica, il programma offre un metodo di calibrazione manuale che consente di correggere rapidamente l'effetto parallasse, anche da remoto. Per poter funzionare è necessario che l'utente selezioni manualmente 4 punti che il programma utilizzerà per modificare la prospettiva delle immagini. [Appendice A]

N.B. È molto importante evidenziare che la procedura descritta in questa sezione deve essere eseguita ogni qual volta si verifichi una variazione dell'asse focale: questo include traslazioni, rotazioni e inclinazioni della fotocamera, anche accidentali.

2.2.1 Procedura di calibrazione

Per avviare la modalità di calibrazione, eseguire il programma da terminale aggiungendo l'argomento **-calibrate** oppure **-c**

```
./tracciamentoPendolo -c
```

oppure

```
./tracciamentoPendolo -calibrate
```

L'interfaccia grafica del sistema di calibrazione (Figura 4) è composta da due finestre: la **Camera View** mostra le immagini estratte in tempo reale dalla videocamera mentre la finestra **Warped View** mostra il risultato della trasformazione prospettica. In fase di avvio il programma controlla che il file **calibration.txt** sia presente nella cartella **HOME**: questo è il file su cui vengono salvate le coordinate dei punti selezionati, così da poter essere lette dal programma durante l'esecuzione in modalità di tracciamento. Se il file è presente e contiene delle coordinate valide queste vengono caricate e utilizzate per applicare la trasformazione, altrimenti un nuovo file viene creato. Inoltre, se nella medesima cartella è presente il file **template.png**, allora la modalità di calibrazione applica il Template Matching alle immagini mostrate nella finestra **Camera View**. In questo caso la finalità del matching è solo quella di fornire un ulteriore feedback visivo all'utente: non viene effettuata nessuna estrazione di coordinate. L'immagine 4 mostra l'interfaccia grafica appena descritta. Lo scopo dell'utente è quello di posizionare 4 punti sull'immagine

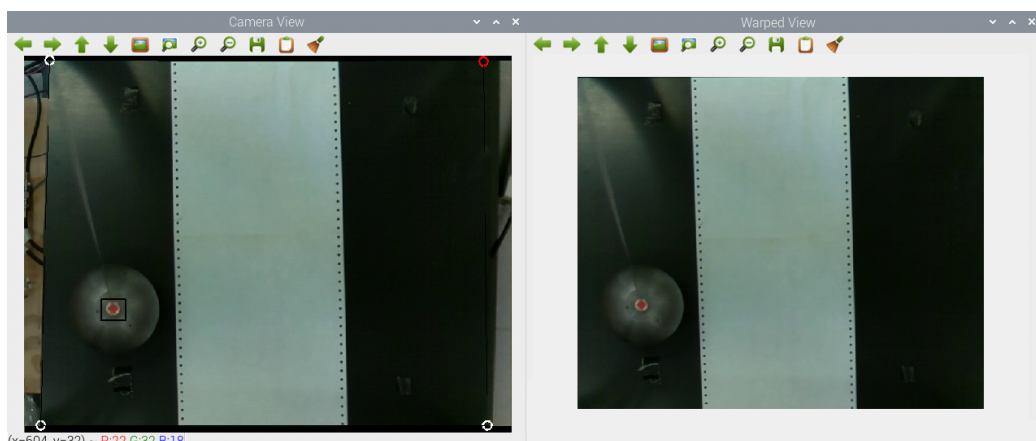


Figura 4: Interfaccia grafica del sistema di calibrazione.

estratta dalla videocamera in modo da individuare un quadrilatero. Questo quadrilatero deve contenere la regione su cui si vuole applicare la trasformazione prospettica. È **molto importante** assicurarsi che l'oggetto tracciato si trovi sempre all'interno della regione selezionata. Qualora l'oggetto dovesse essere individuato fuori da tale regione, le coordinate *corrette* associate a tale punto assumerebbero un valore negativo.

Prima di descrivere più nel dettaglio la procedura di calibrazione, elenchiamo i pulsanti presenti nella barra posizionata in alto nella finestra.

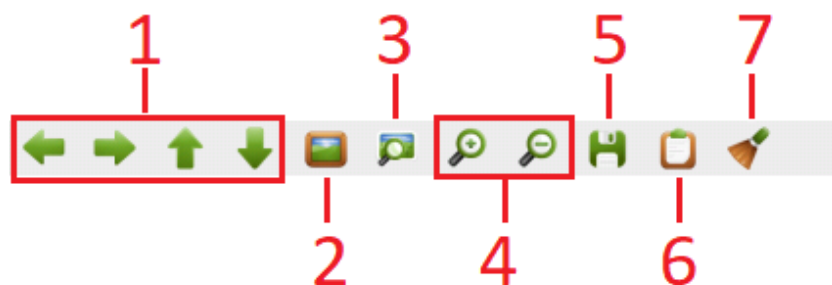


Figura 5: Barra dei pulsanti relativi alla gestione della finestra.

Facendo riferimento alla numerazione in Figura 5 abbiamo le seguenti:

1. Freccie direzionali con cui muoversi all'interno dell'immagine. Funzionano solo se l'immagine è zoommata. È possibile ottenere lo stesso effetto utilizzando le frecce presenti sulla tastiera.

2. Ripristina lo zoom dell'immagine al valore di default.
3. Applica lo zoom massimo, permettendo di osservare i valori RGB dei singoli pixel.
4. Aumentare o ridurre lo zoom. In alternativa è possibile utilizzare la rotellina del mouse.
5. Salva l'immagine mostrata dalla finestra.
6. Copia il contenuto della finestra negli appunti. Serve per incollare l'immagine in un documento senza doverla salvare prima su file.
7. Apre la barra con i pulsanti relativi alla procedura di calibrazione.

Per poter accedere agli strumenti da utilizzare per la calibrazione, premere l'icona numero 7. Alla pressione del pulsante si aprirà la barra in Fig. 6

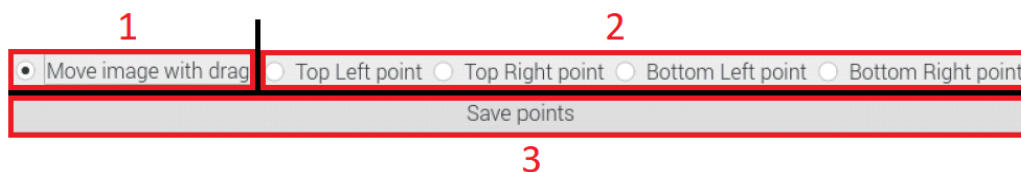


Figura 6: Barra dei pulsanti per la selezione dei punti.

La barra in Fig. 6 contiene i seguenti pulsanti:

1. **Move image with drag**: permette di muoversi all'interno dell'immagine facendo click su un punto qualunque dell'immagine stessa e trascinando il mouse, funzione particolarmente utile per spostarsi rapidamente quando l'immagine ha un alto livello di zoom. Questa è la modalità di default.
2. Ognuna delle radiobox nel rettangolo 2 permette di posizionare un vertice del quadrilatero. In ordine da sinistra a destra abbiamo:
 - Seleziona il vertice in alto a sinistra
 - Seleziona il vertice in alto a destra
 - Seleziona il vertice in basso a sinistra
 - Seleziona il vertice in basso a destra

3. Salva sul file `calibration.txt` le posizioni dei punti.

N.B.: il salvataggio delle coordinate non è automatico alla chiusura del programma.

Come anticipato, l'utente ha il compito di posizionare 4 punti in modo da individuare un quadrilatero sull'immagine estratta dalla videocamera. Solo sulla regione contenuta da questo quadrilatero verrà applicata la trasformazione della prospettiva: è importante che il quadrilatero sia posizionato in modo da contenere sempre l'oggetto da tracciare. Prendiamo come esempio un frame raffigurante il pendolo estratto dalla videocamera ed ipotizziamo sia la prima volta che la procedura di calibrazione viene eseguita. All'avvio del programma verrà generato un nuovo file `calibration.txt` vuoto: per questo motivo la finestra **Camera View** è priva di punti mentre la **Warped View** risulta totalmente nera (Fig 7).

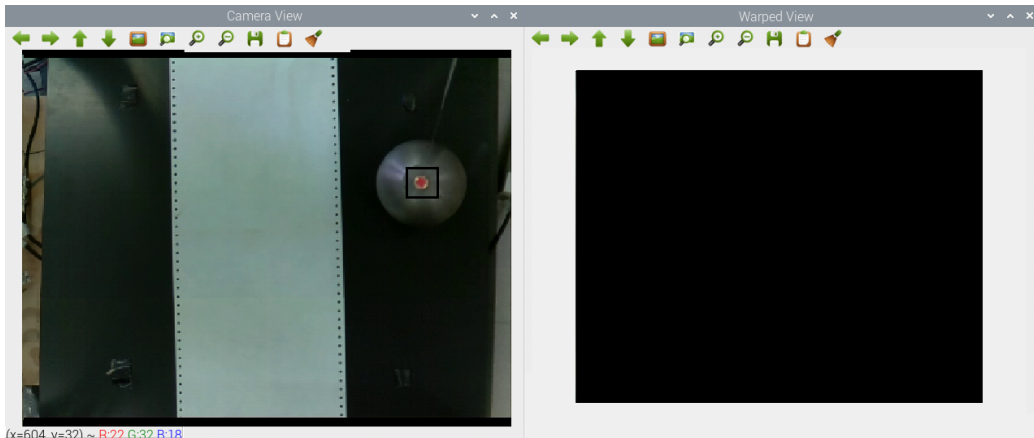


Figura 7: GUI della procedura di calibrazione nel momento del primo avvio.

Per prima cosa apriamo la barra dei pulsanti (Fig. 6) e selezioniamo il **Top Left Point**, ovvero il vertice in alto a sinistra del quadrilatero che dobbiamo posizionare. In questo caso il pendolo ha una base quadrata che ci facilita molto nella scelta dei punti: basta infatti piazzare i punti nei vertici di questo quadrato. Dal momento che anche pochi pixel di differenza nella posizione di uno di questi punti può modificare significativamente il risultato ottenuto, zoommiamo maggiormente e posizioniamo il punto come in Fig. 8. Per

posizionare un punto **attivo** è sufficiente fare click sul punto del frame desiderato.



Figura 8: Posizionamento del Top Left Corner sul frame mostrato dalla Camera View

Come appena detto, i punti possono essere spostati solo quando sono **attivi**: un punto diventa attivo quando la radiobox corrispondente viene premuta nella barra dei pulsanti in Fig. 6. Inoltre un punto attivo viene identificato sul frame tramite un cerchio rosso. I punti non attualmente attivi sono invece identificati tramite un cerchio bianco.

Andiamo adesso a selezionare il **Top Right point** e ripetiamo la procedura ma con l'angolo in alto a destra. Osserveremo che man mano che i punti vengono posizionati l'effetto della trasformazione diventa visibile e mostrato dalla finestra **Warped View** (la prima volta che si applica la prospettiva il risultato potrebbe essere visibile solo dopo aver posizionato tutti e 4 i punti). Una volta posizionati tutti i punti nel modo desiderato è sufficiente premere il pulsante **Save points** per poterli salvare sul file `calibration.txt`. In Fig. 9 è mostrato il risultato della selezione dei punti effettuata durante lo sviluppo del programma.

A questo punto la fase di calibrazione è terminata ed il programma può applicare la correzione prospettica anche durante la modalità di tracciamento (Paragrafo 2.1). Alla prossima riapertura della modalità di calibrazione, il programma mostrerà i punti validi salvati sul file `calibration.txt`.

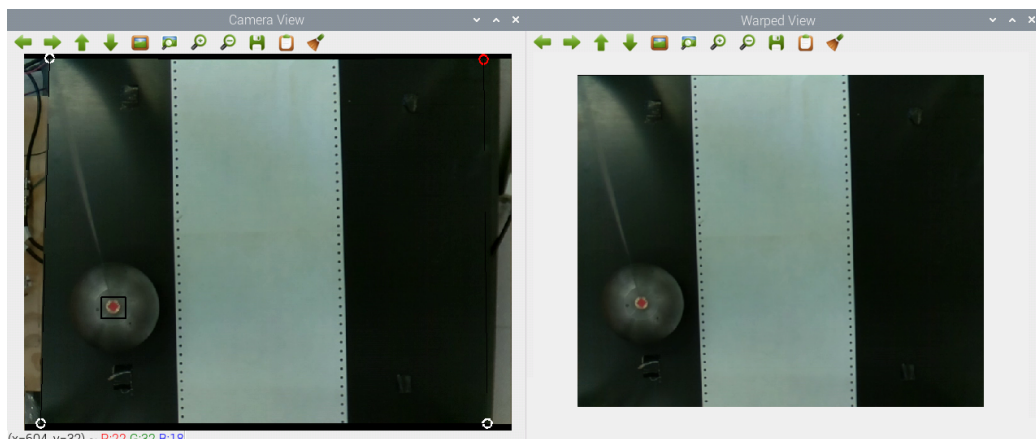


Figura 9: Risultato della procedura di calibrazione. A sinistra l'immagine originale, a destra il frame con la correzione prospettica.

2.3 Grafico 2D da file

In questa sezione descriviamo una modalità di rappresentazione grafica delle coordinate, alternativa a quella real-time precedentemente discussa, che utilizza i file **CSV** generati dal programma. Per attivarla bisogna eseguire il programma specificando il parametro **-g** oppure **-graph**

```
./tracciamentoPendolo -g
```

oppure

```
./tracciamentoPendolo -graph
```

A questo punto il programma richiederà l'inserimento, sempre nel terminale, del percorso **relativo** fino al file **CSV** in cui sono salvate le coordinate che si desidera rappresentare graficamente. Il percorso relativo è quello che parte dalla posizione del file eseguibile ed arriva nella cartella contenente il **CSV**. Supponiamo ad esempio che una precedente esecuzione del programma abbia prodotto un file **CSV** secondo le specifiche elencate nella Sezione 2.1; per semplificare l'esempio ipotizziamo che il file di interesse sia stato rinominato in **myFile.csv**. Se l'eseguibile è contenuto in **Desktop/ProjectFolder** e il **myFile.csv** si trova in **Desktop/ProjectFolder/PendulumCsv**, allora il percorso da fornire è il seguente:

```
../PendulumCsv/myFile.csv
```

La pressione del tasto invio aprirà una finestra su cui verrà tracciato un grafico 2D a partire dalle coordinate contenute nel file **CSV** indicato. Già a seguito di pochi istanti di esecuzione il grafico di prova visualizzato assomiglierà a quello riportato in Figura 10.

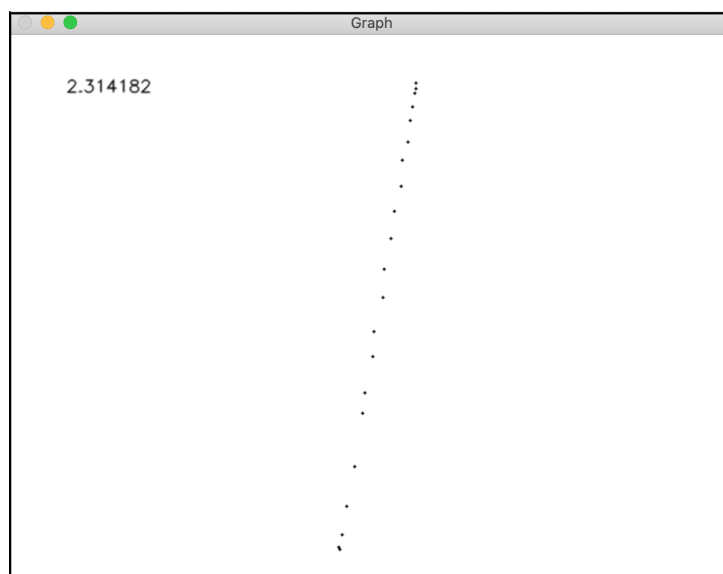


Figura 10: Esempio di esecuzione della funzione di tracciamento del grafico 2D in modalità offline.

Il numero in alto a sinistra nella finestra mostra il tempo di estrazione dell'ultima coordinata disegnata: si tratta cioè dell'istante di tempo associato alla coordinata contenuto nel documento **CSV** fornito in input. Di default il grafico mostra un numero di 30 punti, salvo ad avvio del programma in cui il numero di coordinate mostrate sale gradualmente fino al raggiungimento dei 30 punti. Arrivato a regime questo numero viene mantenuto, cancellando via via i punti più vecchi. Il numero di punti contemporaneamente visibili sul grafico può essere modificato attraverso la seguente procedura:

1. Fare click sulla finestra *graph* per assicurarsi che questa sia attiva. Premere dunque il tasto **c** o in alternativa il tasto **5** per modificare il numero di **coordinate**.
2. Sul terminale sarà comparsa una scritta in cui viene richiesto di digitare la quantità di punti da mostrare.

3. Dopo aver digitato il nuovo valore sul Terminale, confermare premendo invio. Il terminale stamperà a video un messaggio di conferma del nuovo valore selezionato. Osservando la finestra *graph* sarà possibile vedere il risultato della modifica applicata. Le situazioni che si possono verificare sono due:

- Viene inserito un numero di valori **superiore** a quello precedentemente utilizzato. L'ammontare delle coordinate allora salirà progressivamente con la lettura del documento, fino ad arrivare al valore richiesto.
- Viene inserito un numero di valori **inferiore** a quello precedentemente utilizzato. I punti in eccedenza e cronologicamente più vecchi vengono istantaneamente rimossi.

È stato inoltre deciso di imporre un limite inferiore di punti visualizzabili, posizionato a 5. Inserendo invece un numero molto elevato di punti, potenzialmente anche il numero totale dei punti registrati nel documento, è possibile visualizzare un grafico in grado di disegnare l'intero moto del pendolo (Figura 11).

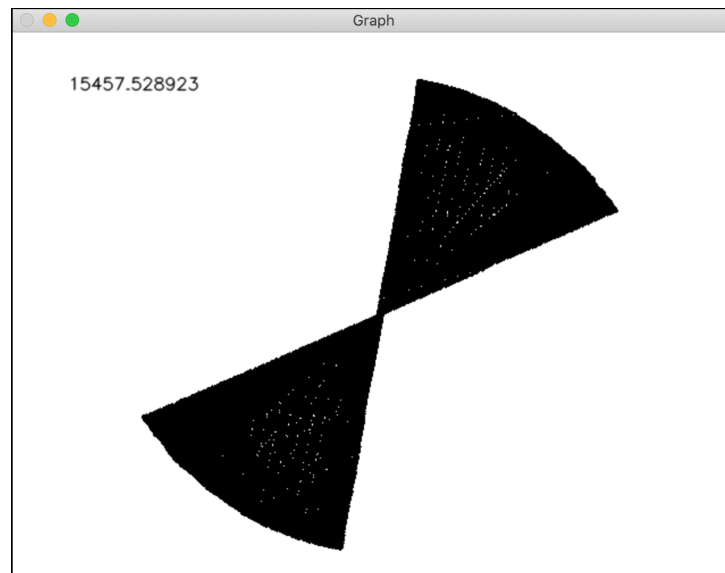


Figura 11: Esempio di grafico 2D prodotto utilizzando un elevato ammontare di punti, equivalenti a circa 4 ore di esecuzione.

N.B. È importante sottolineare che prima di poter modificare il numero di coordinate, così come per eseguire una qualunque delle operazioni descritte nel resto della sezione corrente, bisogna assicurarsi che la finestra del grafico sia attiva. Solo in tal caso, infatti, il programma può intercettare la pressione dei tasti. Eventuali richieste avanzate successivamente dal Terminale, invece, richiederanno di essere scritte direttamente nel Terminale, come per esempio l’inserimento del numero di punti.

In Tabella 2 è riportata la lista completa delle funzioni che possono essere invocate dall’utente durante l’esecuzione della modalità di visualizzazione di grafico offline.

Comando	Dettagli
p oppure 1	Per mettere in pausa l’esecuzione. Occorrerà premere r per riprenderla.
r oppure 2	Riavvia l’esecuzione dopo una pausa. Premere p per fermare nuovamente.
v oppure 3	Per modificare la velocità da un valore da 0 (più lento), a 4 (più veloce). Per default tale valore è 1.
s oppure 4	Salva il frame mostrato correntemente nella cartella principale da cui è avviato il programma.
c oppure 5	Modifica il numero di punti di coordinate mostrate nel grafico contemporaneamente. Per default il valore è di 30 punti.
h oppure ?	Mostra un messaggio di aiuto contenente una tabella simile a quella qui fornita.

Tabella 2: Comandi per il controllo del grafico offline.

Procedendo con ordine, premere il tasto **p** o equivalentemente **1**, avendo prima cliccato sulla finestra *graph*, provoca la messa in **pausa** dell’esecuzione. Il tempo viene dunque arrestato, ma è comunque possibile effettuare tutte le operazioni realizzabili con il programma in moto. Premere consecutivamente

l'opzione di pausa provoca una risposta dal Terminale. Questo suggerisce che forse l'azione desiderata era quella di **re-start** del moto.

Per riavviare il disegno delle coordinate premere il tasto **r** o 2. Anche in questo caso premere consecutivamente il comando **r** non produrrà effetti.

Per modificare la **velocità** con cui vengono visualizzati i nuovi punti (e dunque cancellati quelli vecchi) premere il tasto **v** o equivalentemente 3. A questo punto verrà richiesto di inserire nel terminale il nuovo valore della velocità. Il programma supporta 5 velocità dove 0 è la più lenta e 4 è la più veloce. La velocità di default è impostata ad 1. Il terminale stamperà a video un messaggio di conferma del nuovo valore selezionato.

Per **salvare** su file un'istantanea del contenuto della finestra *graph* premere **s** oppure 4. Il grafico verrà salvato nella cartella in cui si trova l'eseguibile con un nome della forma **graph-"tempo".png** e in cui **"tempo"** indica il tempo riportato sul grafico in alto a sinistra. Il nome del file salvato viene comunque comunicato anche dal Terminale. In alternativa è possibile utilizzare l'icona *salva* presente nella barra dei pulsanti della finestra.

Premendo **h** o **?** un breve riassunto dei comandi viene stampato sul terminale.

A Compilare il programma

Per rendere effettive eventuali modifiche del codice risulta necessario compilare il programma, operazione da compiere tramite terminale.

Per prima cosa spostiamoci nella cartella contenente il codice; in questo esempio supponiamo sia quella originale raggiungibile con il comando seguente:

```
cd Desktop/ProjectFolder/release/source
```

Per compilare eseguiamo il comando:

```
g++ calibration.cpp main.cpp offlineGraph.cpp -o  
tracciamentoPendolo -I/usr/local/include/opencv4/  
-L/usr/local/lib  
-I/home/pi/Desktop/ProjectFolder/headers -lopencv_core  
-lopencv_imgproc -lopencv_highgui -lopencv_ml  
-lopencv_video -lopencv_features2d -lopencv_calib3d  
-lopencv_objdetect -lopencv_flann -lopencv_videoio  
-lopencv_imgcodecs -lpthread
```

1. Il nome del comando è **g++** e serve per invocare il compilatore per C++ presente sul raspberry.
2. Di seguito alla chiamata al compilatore è riportata una lista di files **.cpp**: si tratta dei file sorgente del C++. In questo esempio utilizziamo i file sorgente del programma.
3. Il parametro **-o tracciamentoPendolo** serve per generare un file di output eseguibile dal nome **tracciamentoPendolo**.
4. Il resto del comando serve per indicare al compilatore dove reperire le varie librerie e headers utilizzati nel programma.
 - Il parametro **-I** serve per specificare al compilatore ulteriori percorsi di progetto da compilare. Nella cartella **headers** sono contenuti i file di intestazione del programma, ovvero i file che contengono le dichiarazioni delle funzioni utilizzate.

- Il parametro `-L` serve per indicare al compilatore un percorso cartella nel quale cercare i file delle librerie esterne utilizzate. In particolare la cartella `/usr/local/lib` contiene i files della libreria OpenCV: specifichiamo i nomi di questi file a seguire semplicemente scrivendo `-lnameFile`.

Ulteriori informazioni sul compilatore `g++` sono reperibili sul sito ufficiale <https://gcc.gnu.org/>

B Appendice Matematica

B.1 Modifica della prospettiva

Per modificare la prospettiva viene applicata un'Omografia definita dai punti selezionati manualmente dall'utente nell'immagine originale e gli angoli dell'immagine di output. Tali punti permettono di definire la *Matrice di Trasformazione* associata all'omografia, ovvero la matrice che descrive in che modo mappare i punti da un'immagine all'altra. (Lemma 1)

Lemma 1 *Data una trasformazione $T : R^n \rightarrow R^m$ ed un punto $x \in R^n$, la matrice di trasformazione $M_{n,m}$ associata a T è la matrice tale che*

$$T(x) = Mx$$

La matrice viene calcolata attraverso l'equazione (1) a partire da 4 coppie di punti del tipo $\langle P_{in}, P_{out} \rangle$. Geometricamente, i punti corrispondono ai vertici di 2 quadrilateri, uno posizionato sull'immagine di Input ed uno su quella di Output. Le coppie definiscono le relazioni tra i vertici delle due figure, cioè indicano in che modo mappare ogni punto P_{in} nel frame di output.

$$\begin{bmatrix} t_i x'_i \\ t_i y'_i \\ t_i \end{bmatrix} = M \cdot \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (1)$$

dove $i = 0 \dots 3$ and $P_{in} = (x_i, y_i), P_{out} = (x'_i, y'_i)$

I criteri con cui selezionare i punti sono i seguenti. I 4 punti sull'immagine in input devono essere posizionati dall'utente in modo da delimitare una

regione su cui applicare la trasformazione della prospettiva. Il quadrilatero dell'Output è sempre un rettangolo che coincide con l'immagine di output stessa. I punti P_{out} sono dunque applicati ai vertici del frame in uscita. Questo vuol dir che la loro posizione dipende dalle dimensioni del frame. Per evitare che la trasformazione deformi il rapporto delle forme degli oggetti, le dimensioni dell'output devono corrispondere al più piccolo rettangolo in grado di circoscrivere il quadrilatero in input.

Nella modalità di calibrazione la matrice viene calcolata utilizzando la funzione `getPerspectiveTransform` e viene poi fornita come parametro durante la chiamata a `warpPerspective` che applica effettivamente la trasformazione prospettica all'immagine. Durante l'esecuzione del programma di tracciamento è invece sufficiente correggere solo le coordinate relative all'unico punto di interesse nella scena, il centro del pendolo. Per farlo utilizziamo la Formula 2, dove (x, y) sono le coordinate del pendolo fornite dal Template Matching.

$$\begin{aligned} x_{new} &= \frac{M_{1,1}x + M_{1,2}y + M_{1,3}}{M_{3,1}x + M_{3,2}y + M_{3,3}} \\ y_{new} &= \frac{M_{2,1}x + M_{2,2}y + M_{2,3}}{M_{3,1}x + M_{3,2}y + M_{3,3}} \end{aligned} \quad (2)$$

Come possiamo notare da (2), la funzione `getPerspectiveTransform` restituisce una matrice di dimensioni 3x3: questo significa, sempre per il lemma 1, che la trasformazione applicata è del tipo $R^3 \rightarrow R^3$. Dal momento che i punti estratti provengono da un'immagine, essi possiedono solamente le componenti x ed y . La (2) è quindi definita assumendo che i punti estratti abbiano in realtà anche come terza componente il valore $z = 1$. A questo punto risulta chiara anche la presenza del denominatore, il cui compito è quello di garantire che anche il risultato della trasformazione sia un punto in R^3 . Possiamo infatti rapidamente verificare quanto espresso in Equazione 3.

$$z_{new} = \frac{M_{3,1}x + M_{3,2}y + M_{3,3}}{M_{3,1}x + M_{3,2}y + M_{3,3}} = 1 \quad (3)$$

Come considerazione finale osserviamo che nelle formule precedenti la scrittura $M_{3,3}$ corrispondono alla versione semplificata di

$$M_{3,3} \cdot z \quad \text{dove} \quad z = 1$$

B.2 Metodo di tracciamento

Per tracciare la posizione del pendolo è stata utilizzata la tecnica del Template Matching. Come anticipato nella Sezione 2.1, essa consiste nel ricercare all'interno di un'immagine di Input la regione più simile al contenuto di un'altra immagine detta Template. Il concetto di similarità tra immagini è molto intuitivo per un essere umano ma necessita di una formalizzazione matematica per poter essere implementato in un software: dobbiamo dunque definire una **metrica di similarità**. Per individuare la metrica più adatta abbiamo confrontato i vari metodi di calcolo del coefficiente di similarità messi a disposizione da OpenCV. Il testo condotto consiste nel contare il numero di eventuali mismatch, ovvero di volte in cui l'algoritmo non ha individuato correttamente il centro del pendolo. Dai risultati riassunti in Tabella 3 osserviamo che l'andamento migliore si ottiene calcolando la **Zero-normalized cross correlation (ZNCC)**, riassunta in (4).

$$R(x, y) = \frac{\sum_{i,j} ((T(i, j) - \mu_T) \cdot (I(x + i, y + j) - \mu_I))}{\sqrt{\sum_{i,j} (T(i, j) - \mu_T)^2 \cdot \sum_{i,j} (I(x + i, y + j) - \mu_I)^2}} \quad (4)$$

$$\mu_T = \frac{\sum_{k,l} T(k, l)}{K \cdot L} \quad \mu_I = \frac{\sum_{m,n} I(m, n)}{M \cdot N}$$

Vediamo dunque da dove deriva la formula 4.

In teoria dei segnali, il coefficiente di Cross Correlazione è spesso utilizzato come parametro di similarità. Dal momento che le immagini sono dei segnali bidimensionali, possiamo applicare lo stesso concetto anche nel campo dell'immagine processing. Consideriamo un'immagine di Input $I(x, y)$ di dimensioni $M \times N$ ed un Template $T(x, y)$ di dimensioni $K \times L$ con $K \leq M$ e $L \leq N$. La correlazione tra le due immagini nel punto (x, y) è data da

$$C(x, y) = \sum_{i=0}^{K-1} \sum_{j=0}^{L-1} T(i, j) I(x + i, y + j) \quad (5)$$

Tuttavia la Cross Correlazione in Formula 5 è molto sensibile ai cambiamenti di intensità nei valori dei pixel. Ad esempio osserviamo che se l'intensità della sorgente raddoppia, anche il valore di C lo fa. Per evitare che questo accada è molto comune utilizzare la versione normalizzata (NCC), espressa in Formula

6.

$$C(x, y) = \frac{\sum_{i,j} T(i, j) I(x + i, y + j)}{\sqrt{\sum_{i,j} T(i, j)^2 \cdot \sum_{i,j} I(x + i, y + j)^2}} \quad (6)$$

A sua volta la NCC possiede un'altra fragilità, ovvero non è invariante rispetto ai cambiamenti di luminosità della scena. Per introdurre la proprietà di invarianza si utilizza dunque la ZNCC in Formula 4: notiamo che essa è molto simile alla NCC, l'unica differenza è che ad ogni pixel viene sottratto il valore medio dei pixel che fanno parte della regione in cui si verifica sovrapposizione.

Tabella 3: Confronto tra diverse metriche di similarità applicate al pendolo in movimento.

Formula	Match when still	Accuracy (%)
SQDIFF	yes	54.5
SQDIFF NORMED	yes	52.2
CCORR	no	10.1
CCORR NORMED	yes	97
CCOEFF	no	13.3
CCOEFF NORMED (ZNCC)	yes	100