UNIONE EUROPEA

REGIONE PUGLIA
AREA POLITICHE PER LO SVILUPPO IL LAVORO E L'INNOVAZIONE

AIUTI A SOSTEGNO DEI CLUSTER TECNOLOGICI REGIONALI PER L'INNOVAZIONE

PERSON

PERSON - PERVASIVE GAME FOR PERSONALIZED TREATMENT OF COGNITIVE AND FUNCTIONAL DEFICITS ASSOCIATED WITH CHRONIC AND NEURODEGENERATIVE DESEASES

Deliverable D10.5 – Definizione dell'interfaccia e del protocollo di comunicazione per la restituzione delle coordinate e realizzazione del setup prototipale

Codice Progetto	:	LQ8FBY0
Titolo Progetto	:	PERSON

Numero Deliverable	:	D10.5
Titolo Deliverable	:	Definizione dell'interfaccia e del protocollo di comunicazione per la restituzione delle coordinate e realizzazione del setup prototipale
Natura del Deliverable	:	Report
Mese di Rilascio da Capitolato	:	Luglio 2017
Livello di Disseminazione	:	Pubblico
Versione	:	1.1
Data Deliverable	:	10/11/2017

Abstract

Viene definita l'interfaccia realizzata in grado di gestire i dati provenienti dal SdVA (Sistema di Visione Artificiale), previa loro elaborazione attraverso i moduli preposti (GPU e FPGA), per renderli disponibili al Sensor Hub secondo modalità e protocolli definiti con il partner Santer Reply. L'interfaccia permette inoltre al SdVA di restituire le coordinate 3D ricostruite per mezzo del setup prototipale.

Keyword list

Sistema di Visione Artificiale, comunicazione, protocollo, formato dei messaggi, interfaccia, prestazioni

Tracking delle Versioni

Version	Changes	Author(s)
1.0	**First Draft**	**Maria Francesca de Ruvo, Marco Colaprico**
1.1	**Aggiunta sezione 8**	**Maria Francesca de Ruvo, Marco Colaprico**

Indice

ABSTRACT **2**

KEYWORD LIST **2**

INDICE **3**

INDICE FIGURE **4**

INDICE GRAFICI **4**

GLOSSARIO, ACRONIMI E ABBREVIAZIONI **5**

1. OBIETTIVO DEL SDVA **6**

1.1 GESTURE 6

1.2 TRACKING DELLA POSIZIONE DELLA MANO 7

1.3 CALIBRAZIONE DEL SISTEMA 7

2. PROTOCOLLO DI COMUNICAZIONE **7**

2.1 SCELTA DEL MODELLO DI COMUNICAZIONE 8

2.2 SCELTA DEL PROTOCOLLO 8

2.3 SCELTA DEL BROKER 9

3. FORMATO DEI DATI **9**

4. DEFINIZIONE DELLE CODE E DEI MESSAGGI **10**

4.1 CODA DI GIOCO 10

4.1.1 Modalità di notifica dei messaggi *10*

4.1.1 Struttura dei messaggi dei dati di gioco *10*

4.2 CODE DI CONTROLLO 12

4.2.1 Messaggi di controllo *12*

4.2.2 Fasi di gioco in cui è coinvolto il SdVA *13*

5. IMPLEMENTAZIONE DELL'INTERFACCIA **13**

5.1 DEFINIZIONE DELLA UI 13

5.2 DEFINIZIONE DELLA LOGICA APPLICATIVA 18

6. RICONOSCIMENTO DELLE GESTURE **21**

6.1 Classificazione dei markers della mano *22*

6.2 Calcolo del numero di dita aperte *24*

6.3 Selezione del pixel puntato dal dito indice *26*

7. TEST E PRESTAZIONI **27**

8. INTEGRAZIONE CON IL SENSOR HUB E TEST DI FRUIZIONE IMMERSIVA DEL GAME **29**

Indice figure

Figura 1 - Interazione tra il SdVA e il Sensor Hub 6
Figura 2 - Schema logico del sistema PERSON 8
Figura 3 - Interazione tra i moduli del sistema PERSON 9
Figura 4 - Sistema a coda 10
Figura 5 - Sistema di riferimento in cui sono restituite le coordinate 3D del baricentro del dorso della mano 12
Figura 6 - Schermata di avvio SdVA 14
Figura 7 - Schermata di calibrazione SdVA 15
Figura 8 - Schermata di calibrazione effettuata 16
Figura 9 - Schermata di verifica della calibrazione 17
Figura 10 - SdVA in attesa di avvio della fase di gioco 17
Figura 11 - SdVA in fase di gioco 18
Figura 12 - Diagramma di stato del SdVA 19
Figura 13 Modello della mano adottato per il labelling dei markers 21
Figura 14 Flow chart concettuale dell'algoritmo di riconoscimento delle gesture 22
Figura 15 Flow chart concettuale dell'algoritmo di classificazione dei markers della mano 22
Figura 16 Geometria della mano. Il pollice è sempre contenuto nel semipiano che ha come origine la retta AB e che non contiene il punto C (semipiano evidenziato dal colore giallo). 23
Figura 17 Geometria della mano. Se si intersecano i segmenti che hanno come primo estremo il marker A e come secondo estremo la punta delle dita diverse dal pollice, l'intersezione più vicina al punto B, corrisponde a quella ottenuta dall'indice. 23
Figura 18 Classificazione dei markers della mano. In figura si può osservare come il SDVA ha classificato correttamente i markers. Il sensore del dorso della mano è stato riconosciuto correttamente, infatti sono stati colorati di rosso i markers A e C e di giallo il marker B, corrispondente alla nocca dell'indice. In celeste è stato evidenziata la punta del pollice, in blu la punta dell'indice e in verde la punta del medio, dell'anulare e del mignolo. 24
Figura 19 Gestualità: << 0 dita aperte >>. In figura (a) è semplice riconoscere che non ci sono dita aperte, in quanto non compaiono nell'immagine i markers relativi alle punta delle dita. In figura (b) è evidente che il SDVA, per riconoscere che il paziente ha tutte le dita chiuse, ha dovuto rilevare che il pollice, il cui marker è visibile nell'immagine acquisita, è chiuso. 24
Figura 20 Gestualità << 1 dito aperto >>. In figura (a) il numero di dita aperte coincide con il numero di dita presenti nell'immagine. Nelle figure (b), (c) e (d) è evidente che il SDVA, per riconoscere che il paziente ha un solo dito aperto, ha dovuto rilevare quali, tra le dita presenti nell'immagine acquisita, sono chiuse. Nell'immagine (b) il dito medio e l'anulare, anche se visibili, sono chiusi. Nelle immagini (c) e (d), il pollice è chiuso. 25
Figura 21 Gestualità << 2 dita aperte>>. In figura (a) il SDVA riconosce che entrambe le dita presenti nell'immagine sono aperte, mentre nella figura (b) verifica che tra le tre dita visibili, il pollice è chiuso e quindi il numero di dita aperte rimane 2 (numero in rosso in alto a sinistra). 25
Figura 22 Gestualità << 3 dita aperte >>. Il SDVA rileva che tutte e tre le dita presenti in figura (a) sono aperte, mentre, tra le quattro dita presenti in figura (b), il dito medio è chiuso. 26
Figura 23 Gestualità << 4 dita aperte >>. Il SDVA, in figura (a), rileva che tutte è quattro le dita presenti nell'immagine acquisita sono aperte, mentre in figura (b) evidenzia che tra le cinque dita visibili, il pollice è chiuso e quindi il numero di dita aperte rimane 4. 26
Figura 24 Flow chart concettuale dell'algoritmo di selezione del pixel puntato dal dito indice 27
Figura 25 Gestualità << Puntamento >> Il pixel puntato dall'utente è evidenziato con il simbolo di un mirino rosso. 27
Figura 26 - Interfaccia dello stub del Sensor Hub 28
Figura 27 - Interfaccia dello stub del Game 28
Figura 28 – Installazione del dimostratore in un contesto immersivo 30
Figura 29 – Fruizione del serious game. Nel riquadro, il particolare dei marker passivi come inquadrati dal SdVA. 30

Indice grafici

Grafico 1 – Velocità di trasmissione dei pacchetti di gioco 29
Grafico 2 - Prestazioni del SdVA 29

Glossario, acronimi e abbreviazioni

Item	Descrizione
SdVA	Sistema di Visione Artificiale
3D	Tridimensionale
2D	Bidimensionale
ms	Millisecondi (1ms = 10^{-3} secondi)
AMQP	Advanced Message Queuing Protocol
IP	Internet Protocol
MQTT	Message Queue Telemetry Transport
STOMP	Simple/Streaming Text Oriented Messaging Protocol
TCP	Transmission Control Protocol
UI	User Interface

1. Obiettivo del SdVA

L'obiettivo principale del SdVA è essenzialmente quello di realizzare un'interfaccia fra il player e il game, che possa tradurre e interpretare la gestualità del paziente stesso come interazioni fra esso e le scene di gioco.

A tal fine, a partire dai flussi video provenienti dalle 2 telecamere stereo, il SdVA ricostruisce la gestualità del paziente, ed invia le informazioni ottenute (dati di gioco) al Sensor Hub con una frequenza di 50 Hz.

In questo documento vengono definite:

- le gesture che l'applicazione sarà in grado di interpretare
- la struttura e il formato dei dati di gioco che verranno inviati al Sensor Hub
- la modalità e i protocolli da utilizzare

La Figura 1 mostra lo schema logico di interazione tra l'applicazione di VA, il Sensor Hub e il Game.

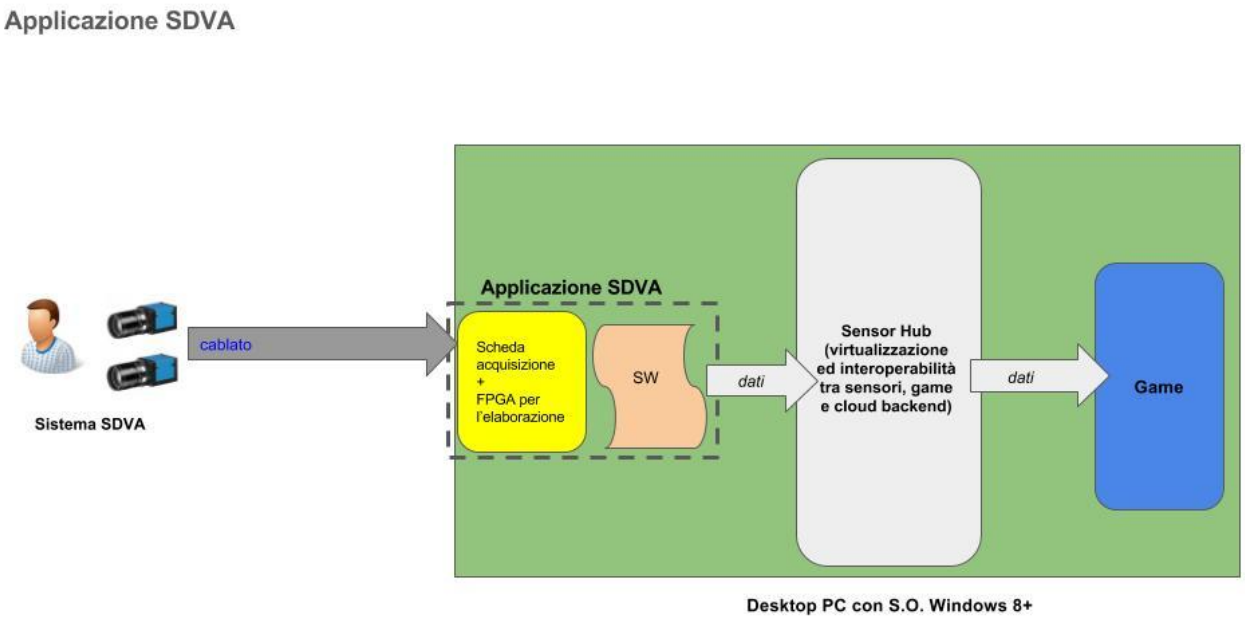


Figura 1 - Interazione tra il SdVA e il Sensor Hub

1.1 Gesture

Si è partiti dall'obiettivo di individuare le coordinate del pixel puntato dal paziente nell'interazione con il serious game, si è poi valutata anche la possibilità di interpretare delle gesture che riproducano ad esempio la presa di un oggetto virtuale ("grab"), la conferma di un puntamento (cioè quello che si è soliti fare con il "click" del mouse) e la possibilità di riconoscere automaticamente il numero di dita che l'utente ha aperto.

Si è deciso di associare il significato di prendere un oggetto virtuale ("grab"), al gesto intuitivo di chiusura del pugno.

Mentre per l'operazione di "click", sono state valutate le seguenti due ipotesi:

- collegare il "click", all' "immobilità" del puntamento, cioè al fatto che il puntatore non si muova e resti localizzato in un'area circoscritta per un periodo di tempo predefinito (l'ampiezza della area circoscritta e la durata del tempo, che definiscono il concetto di immobilità, possono diventare dei

parametri definiti in funzione del paziente, se si verificherà la necessità di adattare il prototipo PERSON ai singoli pazienti)
- associare al “click” un gesto convenzionale, come per esempio alzare il pollice o chiudere la mano

La seconda ipotesi è stata scartata in quanto nell’esecuzione della gestualità associata al “click”, il paziente potrebbe modificare la direzione del dito che sta puntando e si potrebbe generare così un equivoco sull’oggetto puntato. Inoltre associare un ulteriore gestualità al paziente potrebbe risultare troppo “complesso” per alcuni pazienti.

Poiché, sia il gesto relativo alla presa di un’oggetto virtuale, che quello che si è deciso di associare al significato del “click”, possono essere interpretati a partire dal numero di dita che l’utente ha aperto, il SdVA si limiterà a fornire al Sensor Hub ogni 20 ms:

- le coordinate del pixel puntato
- il numero di dita che il paziente ha aperto

Sarà poi compito del Game capire se:

- il paziente che ha chiuso il pugno sta indicando il numero zero oppure sta afferrando un oggetto virtuale
- il paziente che ha solo un dito aperto, sta indicando il numero uno, o sta compiendo l’azione di puntare
- il paziente che punta in un’area circoscritta del monitor per un periodo di tempo predefinito, sta compiendo l’azione di click

Si è scelto di delegare tale compito al Game, in quanto il Game ha delle informazioni relative alla corrente scena di gioco, che il SdVA non ha, e grazie a queste ulteriori informazioni può interpretare più correttamente i dati forniti dall’applicazione di VA.

1.2 Tracking della posizione della mano

Oltre all’interpretazione delle gesture, si è valutata anche la possibilità che l’oggetto virtuale, una volta afferrato mediante apposito gesto (qualora la gesture del “grab” risulti implementabile in modo affidabile), possa essere traslato in base alla posizione 3D della mano del paziente.

Affinché il game possa ricavare in tempo reale le informazioni relative allo spostamento della mano, il SdVA invierà al Sensor Hub, ogni 20 ms, anche:

- le coordinate 3D del baricentro del dorso della mano
- il versore normale al piano del dorso della mano

1.3 Calibrazione del sistema

Per poter espletare le proprie funzionalità, è necessaria una prima di fase di calibrazione in cui attraverso un pattern mostrato sul display del monitor, questo è individuato dalle telecamere e la sua posizione è referenziata in 3D. Questa operazione è indispensabile per poter ricavare correttamente il pixel del monitor indicato, le coordinate 3D del baricentro del dorso della mano, nonché il versore normale al piano del dorso della mano. Se durante l’utilizzo del sistema la posizione delle telecamere o del monitor dovesse subire variazioni, sarà necessario ripetere la calibrazione.

2. Protocollo di comunicazione

Basandosi sullo scenario globale del sistema PERSON, si è prima scelto il paradigma di comunicazione più idoneo, poi il protocollo ed infine il message broker.

2.1 Scelta del modello di comunicazione

Il sistema Person è un sistema distribuito su tre livelli. Il primo livello è costituito dalla sensoristica che comprende il guanto (sensore/attuatore aptico), il setting di visione artificiale e il casco EEG; il secondo livello è costituito dall'unità client che ospita su un desktop PC la parte applicativa, di gioco e il Sensor Hub; il terzo livello è il cloud backend che si occupa principalmente di memorizzare e analizzare i dati provenienti dal tracciamento delle attività di sistema.

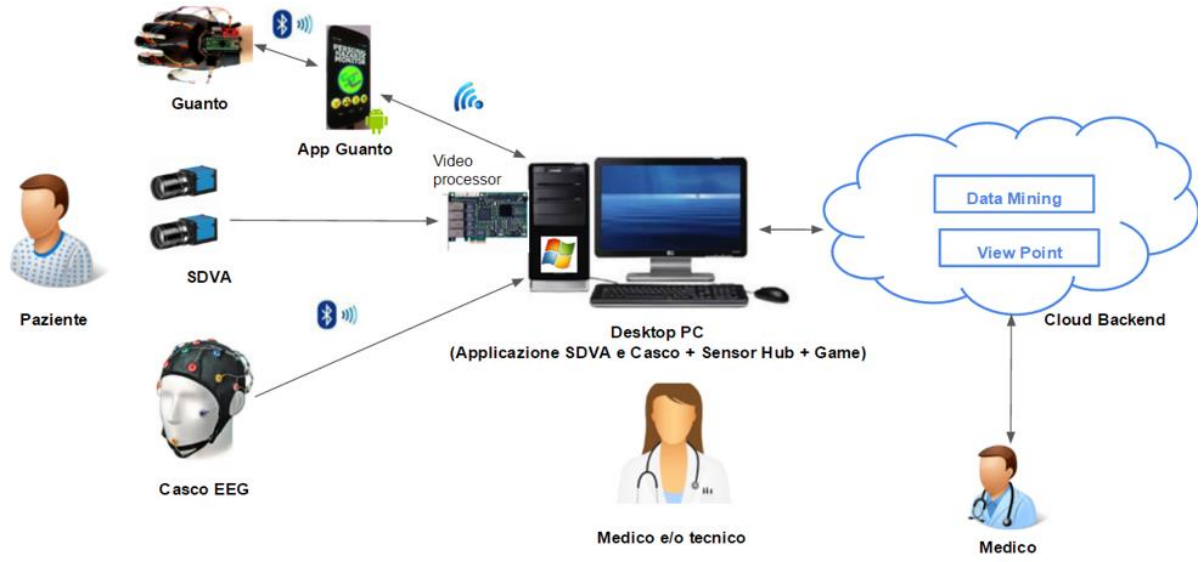


Figura 2 - Schema logico del sistema PERSON

Per garantire la portabilità e la flessibilità del sistema, per la comunicazione tra i diversi moduli, si è scelto di utilizzare invece che il tradizionale modello client/server, dove il client comunica direttamente con un endpoint, il paradigma publish/subscriber, che disaccoppia il client che sta inviando un messaggio particolare (publisher), da un altro client, che riceve il messaggio (subscriber). Ciò significa che il publisher e il subscriber non conoscono l'esistenza l'uno dell'altro. Nel modello publisher/subscriber c'è un terzo componente, chiamato broker, che è conosciuto sia dal publisher che dal subscriber, che filtra tutti i messaggi in arrivo e li distribuisce di conseguenza.

Il disaccoppiamento tra publisher e subscriber si può distinguere nelle seguenti tre dimensioni:

- **Disaccoppiamento spaziale**: pur interagendo, non è necessario che il publisher ed il subscriber si conoscano. Questo perché l'interazione è indiretta, cioè si verifica tramite intervento del broker
- **Disaccoppiamento temporale**: poiché l'interazione tra publisher e subscriver è indiretta, non è necessario che entrambe le entità siano attive. Quando il publisher notifica un evento, tale evento verrà riferito al subscriber interessato, non appena questo risulterà attivo
- **Disaccoppiamento della sincronizzazione**: il publisher, dopo aver notificato un evento, potrà proseguire con altre attività senza preoccuparsi che i subscriber interessati abbiano ricevuto o meno tale notifica. Di questo infatti si interesserà il broker

2.2 Scelta del protocollo

I tre più comuni e popolari protocolli di messaggistica con paradigma publisher/subscriber, basati su TCP/IP sono:

- MQTT (Message Queue Telemetry Transport)

- STOMP (Simple/Streaming Text Oriented Messaging Protocol)
- AMQP (Advanced Message Queuing Protocol)

MQTT è stato specificamente disegnato per dispositivi con limiti di risorse, bassa banda e reti con alta latenza come linee ad accesso remoto e links satellitari. Date le sue caratteristiche è ideale per sistemi embedded, e applicazioni di tipi mobile.

Il protocollo STOMP è stato invece disegnato per garantire la semplicità e un'alta interoperabilità, per esempio è possibile connettersi ad un broker STOMP utilizzando un semplice client telnet.

Mentre, le due principali ragioni per utilizzare il protocollo AMQP solo l'affidabilità e l'interoperabilità. Come il nome suggerisce, fornisce una vasta gamma di funzionalità legate alla messaggistica, tra cui una messa in coda affidabile, un instradamento dei messaggi basato sui topic, una grande flessibilità del routing e la sicurezza.

Dato che il sistema PERSON è ospitato su un desktop PC e che quindi non ha delle particolari esigenze di banda e delle risorse limitate, ma richiede invece l'affidabilità del sistema di messaggistica, si è optato per la scelta del protocollo AMQP.

2.3 Scelta del broker

Come message broker si è scelto di utilizzare RabbitMQ, in quanto: è open source, implementa il **protocollo AMQP 0-9-1**, e le librerie client per interfacciarsi con il broker sono disponibili nei principali linguaggi di programmazione (tra cui java e c++).

In Figura 3 è possibile osservare lo schema di interazione tra i moduli del sistema PERSON.

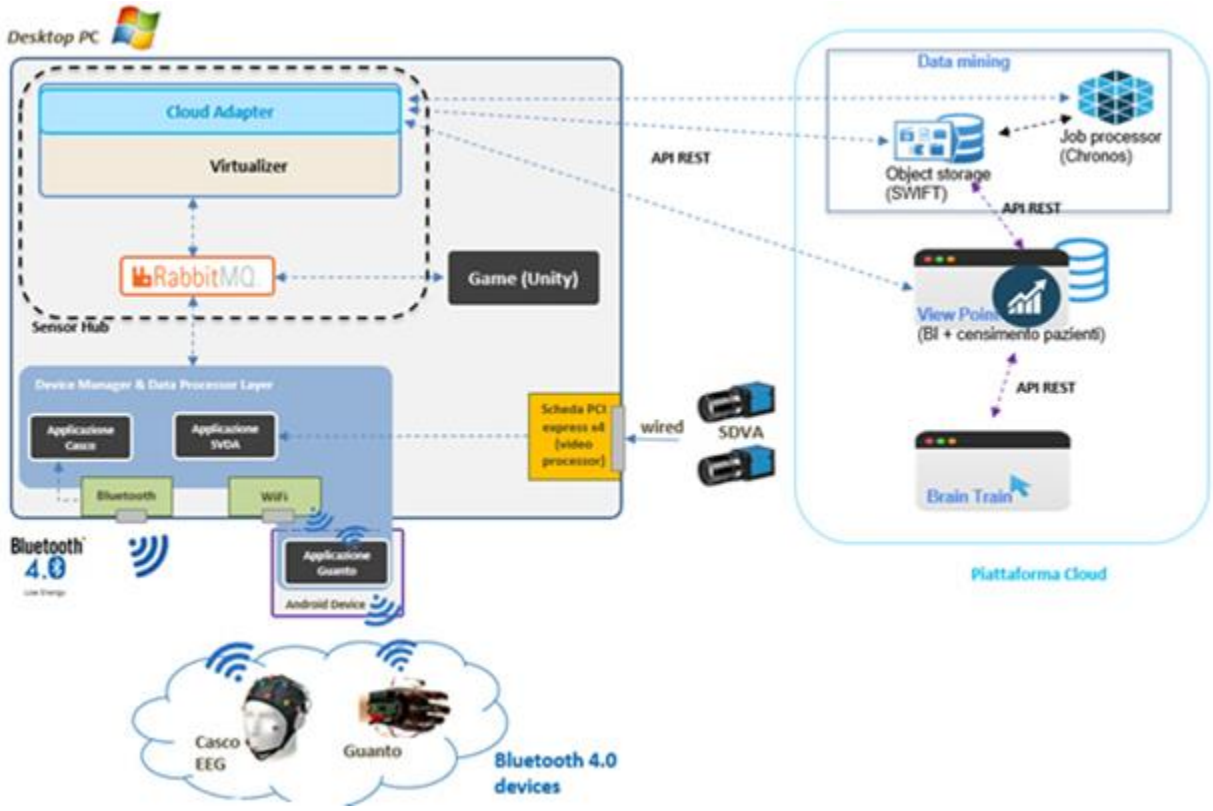


Figura 3 - Interazione tra i moduli del sistema PERSON

3. Formato dei dati

Si è scelto di utilizzare come formato per lo scambio di dati, il formato **JSON** (JavaScript Object Notation). Si tratta di un formato di testo completamente indipendente dal linguaggio di programmazione. Questa caratteristica associata alla sua semplicità hanno reso JSON un linguaggio ideale per lo scambio di dati.

4. Definizione delle code e dei messaggi

La Figura 4 mostra lo schema delle code per la comunicazione dei moduli che costituiscono il sistema PERSON.

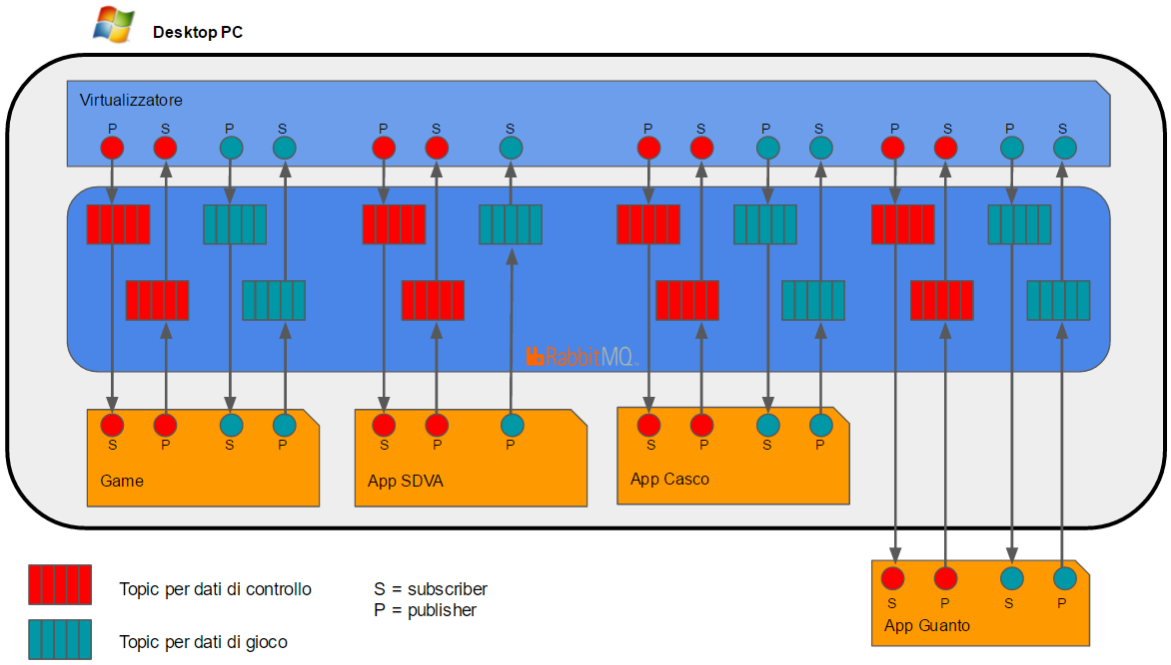


Figura 4 - Sistema a coda

Come è possibile osservare, si è scelto di utilizzare un unico Message Exchange per tutte le comunicazioni e differenziare in due tipi, le code tra Sensor Hub e gli altri moduli:

1. code di controllo
2. code di gioco

Le prime vengono utilizzate per coordinare le fasi di gioco, mentre le seconde per scambiare i dati.

Le code che il modulo di VA utilizza per interfacciarsi con il Sensor Hub sono tre:

1. una coda per i messaggi di gioco dal SdVA verso il Sensor Hub
2. una coda per i messaggi di controllo dal Sensor Hub verso il SdVA
3. una coda per i messaggi di controllo dal SdVA verso il Sensor Hub

4.1 Coda di gioco

Come preannunciato nel paragrafo precedente la coda di gioco relativa al SdVA viene utilizzata per inviare al Sendor Hub i dati di gioco, ottenuti dall'analisi delle gesture.

4.1.1 Modalità di notifica dei messaggi

I dati vengono inviati dal SdVA al SensorHub con una frequenza di 50 Hz, e con una notifica di tipo push, in modo tale da garantire una comunicazione in tempo reale tra i due moduli.

4.1.1 Struttura dei messaggi dei dati di gioco

Di seguito viene mostrato il pacchetto di dati trasmesso:

{

“timeStamp”:string,

“pixel”:[float,float],

“monitorSize”:[float,float],

“openFingers”:int,

“point3D”:[float,float,float],

“normalPlane”:[float,float,float]

}

dove:

- **timeStamp**: ora e data del momento in cui il pacchetto dati viene inviato. Il timestamp è espresso come millisecondi trascorsi a partire da UNIX epoch (1 gennaio 1970).
- **pixel**: è la posizione in pixel di ciò che sta puntando il paziente. La prima componente del vettore corrisponde alla coordinata x e la seconda alla coordinata y. I valori assunti dalla coordinata x variano nel range [-1, W], dove W indica la larghezza dello schermo in pixel. L’intervallo dei possibili valori della coordinata y è invece [-1, H], dove H corrisponde all’altezza dello schermo in pixel. Se entrambe le componenti del vettore sono positive, il modulo Game può considerare valido questo campo, se invece entrambe le componenti del vettore assumono valore -1, significa che il SdVA non è riuscito ad interpretare la gesture del paziente (perché per esempio la mano è fuori dall’area di lavoro, oppure il paziente sta puntando fuori dallo schermo) e quindi il campo “pixel” perde validità.
- **monitorSize**: è la dimensione dello schermo in pixel. La prima componente del vettore rappresenta la larghezza in pixel dello schermo (W) e la seconda l’altezza (H).
- **point3D**: sono le coordinate 3D (in cm) del baricentro del palmo della mano. In base alle esigenze del modulo Game, le coordinate sono riferite ad un sistema di riferimento, la cui origine si ottiene traslando il centro del monitor di “- Delta z” lungo l’asse z e di “- Delta y” lungo l’asse y, e gli assi sono orientati come in Figura 5. Il range dei valori delle coordinate è limitato dallo spazio fisico inquadrato dalle telecamere, nel volume fra il paziente e il monitor. Quindi supponendo che il centro di riferimento del sistema, specificato da Delta z e Delta y (i cui valori vengono indicati nella fase di calibrazione del SdVA), sia approssimativamente negli occhi del player, e questo sia seduto in una posizione “usuale” rispetto al monitor, i valori di sicurezza per circoscrivere il range delle coordinate 3D restituiti saranno nel range da -100 a +100. Questo campo non è valido se il campo “openFingers” è pari a -1.
- **normalPlane**: è il versore normale al piano del palmo della mano. I 3 float di normalPlane variano da 0 ad 1. Questo campo non è valido se il campo “openFingers” è pari a -1.
- **openFingers**: è il numero di dita della mano aperte. L’intervallo dei possibili valori di questo campo è [-1, 5]. Se openFingers vale -1, significa che il SdVA non è riuscito ad interpretare la gesture del paziente (perché per esempio la mano è fuori dall’area di lavoro), e quindi il campo “openFingers” non è valido; mentre se i valori assunti vanno da 0 a 5, il Game può interpretare tali valori come il numero di dita che il paziente ha aperto.

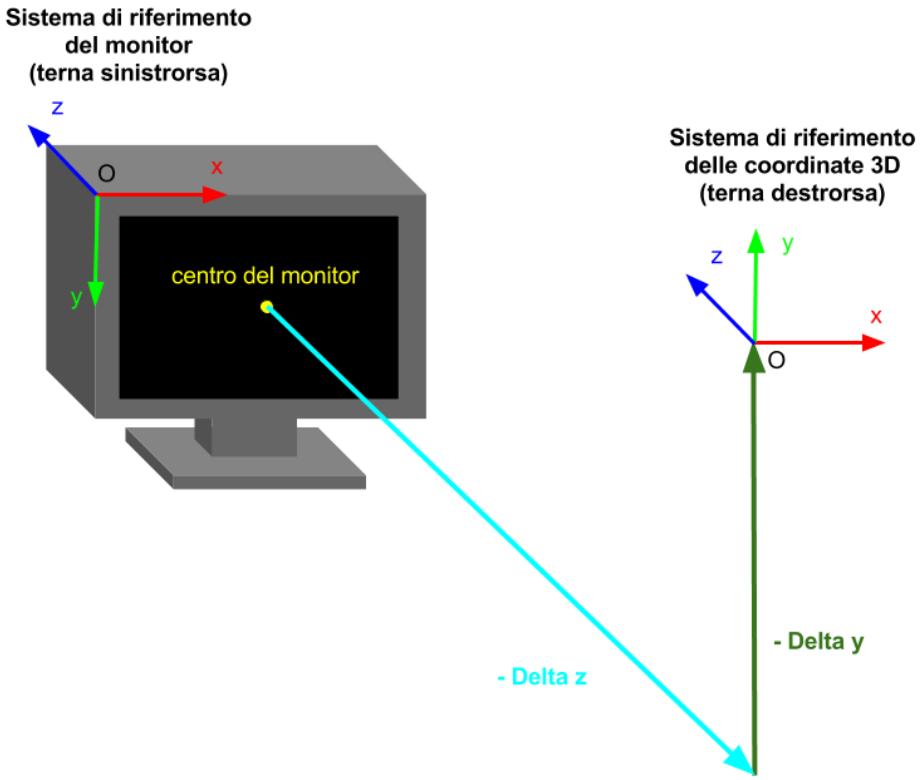


Figura 5 - Sistema di riferimento in cui sono restituite le coordinate 3D del baricentro del dorso della mano

Il campo "monitorSize" è stato introdotto in quanto il SdVA e il Game utilizzano 2 sistemi di riferimento diversi per le coordinate in pixel del monitor. L'SdVA, quando calcola la posizione di ciò che sta puntando il paziente, considera come origine dello schermo l'angolo in alto a sinistra, mentre Unity3D, quando restituisce le coordinate 2D del mouse, considera come origine l'angolo in basso a sinistra. Questo rende necessario, da parte del Game, passare dal sistema di riferimento Top Left Corner, al sistema di riferimento Bottom Left Corner. Per effettuare tale conversione il Game deve conoscere la dimensione dello schermo.

4.2 Code di controllo

Come anticipato le code di controllo sono utilizzate per coordinare le fasi di gioco. L'elenco dei possibili messaggi scambiati su tutte le code di controllo e la definizione completa e dettagliata delle fasi di gioco è già stata presentata nel documento D2.1, ci si limita ora ad elencare solo i messaggi di controllo scambiati tra SdVA e Sensor Hub e a descrivere brevemente le fasi di gioco in cui è coinvolto il SdVA.

4.2.1 Messaggi di controllo

La struttura di messaggi scambiati sulle code di controllo è la seguente:

{

"state": string

}

Il campo "state" può assumere i seguenti valori:

- ready

- start
- stop
- interrupt
- shutdown
- calibrateSdva

4.2.2 Fasi di gioco in cui è coinvolto il SdVA

Le fasi di gioco in cui è coinvolto il SdVA sono essenzialmente tre:

1. Fase 0: Calibrazione del SdVA
2. Fase 1: Ready
3. Fase 2: Gioco

4.2.2.1 Fase 0: Calibrazione del SdVA

Quando l'applicazione parte, entra nella fase di calibrazione, in cui l'utente deve scegliere se effettuare una nuova calibrazione, oppure confermare quella attuale. La scelta di ricalibrare il sistema deve essere effettuata ogni qual volta viene modificata la posizione reciproca del monitor e del sistema di telecamere stereo. In questa fase tutti i messaggi di controllo del Sensor Hub verranno ignorati. Confermata o rinnovata la calibrazione, il SdVA avvia l'acquisizione dei dati dal frame grabber e passa nella fase di ready.

4.2.2.2 Fase 1: Ready

Durante la fase di ready, il SdVA acquisisce i dati provenienti dal frame grabber, senza inviare nessun pacchetto dati al Game, ed è in ascolto dei messaggi di controllo provenienti dal Sensor Hub.

In questa fase il SdVA può ricevere quattro tipi di messaggi:

1. state: ready
2. state: start
3. state: calibrateSdva
4. state: shutdown

Ai messaggi di ready, risponde con un messaggio di ready.

Al messaggio di start, risponde con un messaggio di start e passa alla fase di gioco.

Al messaggio calibrateSdva, risponde con un messaggio calibrateSdva e ritorna alla fase di calibrazione.

Al messaggio di shutdown, risponde con un messaggio di shutdown e esce dall'applicazione.

4.2.2.3 Fase 2: Gioco

In questa fase, il SdVA invia al Game, ad un frequenza di 50Hz, un pacchetto dati contenente informazioni sul movimento della mano del paziente nell'area di gioco e il pixel dello schermo puntato. Inoltre è in continuo ascolto dei messaggi di controllo del Sensor Hub, e nel caso riceva un messaggio di "state: stop" o "state: interrupt", ritorna nella fase di ready.

5. Implementazione dell'interfaccia

5.1 Definizione della UI

L'applicazione di visione artificiale mostra all'avvio un semplice ed intuitiva UI che permette di

- impostare la mano utilizzata dall'utente (in questo prototipo impostata solo su "mano destra")

- calibrare il sistema per individuare il monitor prima di avviare l'applicazione
- verificare l'ultima calibrazione effettuata prima di avviare l'applicazione.

Figura 6 - Schermata di avvio SdVA

Nel caso in cui l'applicazione di visione artificiale sia avviata per la prima volta (oppure è stato alterato il set up) è necessario effettuare la calibrazione del sistema. Questa operazione consiste nell'individuare l'area del monitor puntata dall'utente. Per espletarla, è necessario:

1. rimuovere i filtri ad infrarosso dalle telecamere
2. verificare che il monitor e l'area di lavoro siano inquadrate da entrambe le telecamere prestando attenzione all'inclinazione tra le telecamere ed il display (una vista laterale del display provoca una riduzione della luminosità dello stesso percepita dalle telecamere).

Verificati i due punti, premendo "Rileva monitor" è mostrata a schermo un pattern (una scacchiera bianca e nera); le telecamere sono in grado di rilevarla è quindi di individuare nello spazio la posizione dei 4 vertici del monitor. Questa informazione è indispensabile per fornire correttamente le coordinate dei pixel indicati dall'utente, la posizione 3D del centro del dorso della mano nonché il versore normale al piano del dorso della mano (informazioni che verranno inviate al game).

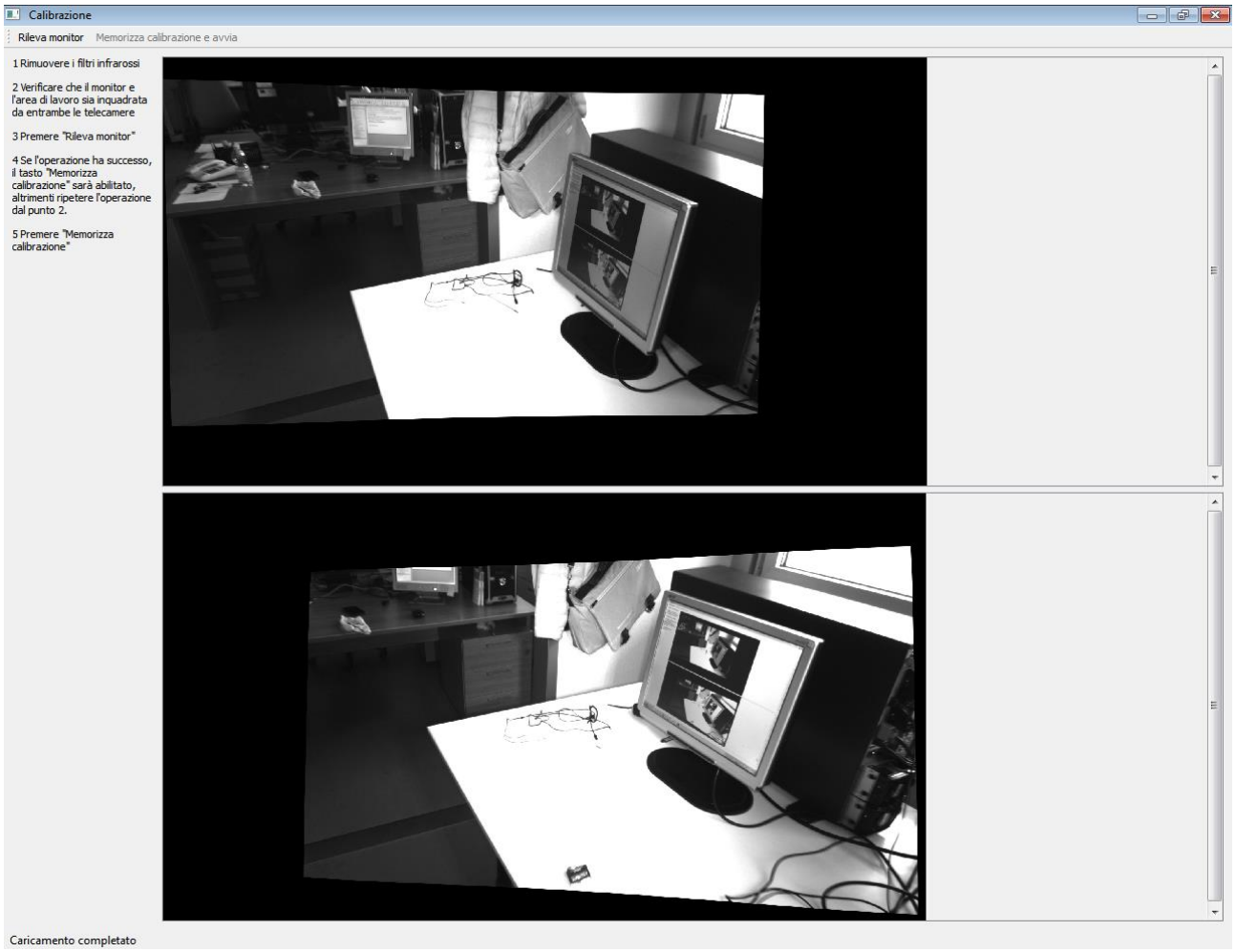


Figura 7 - Schermata di calibrazione SdVA

La ricerca del pattern viene notificata nella parte inferiore della UI. Se la ricerca ha avuto esito positivo viene inoltre disegnata sulle immagini acquisite la regione riconosciuta come monitor.

L'operatore ha la possibilità di confermare l'esito e di validare la qualità della procedura di calibrazione premendo il pulsante "Memorizza calibrazione e avvia".

Se non vengono apportate modifiche al set up (e.g. spostamento del monitor o delle telecamere) tale calibrazione risulterà valida in tutti gli utilizzi successivi.

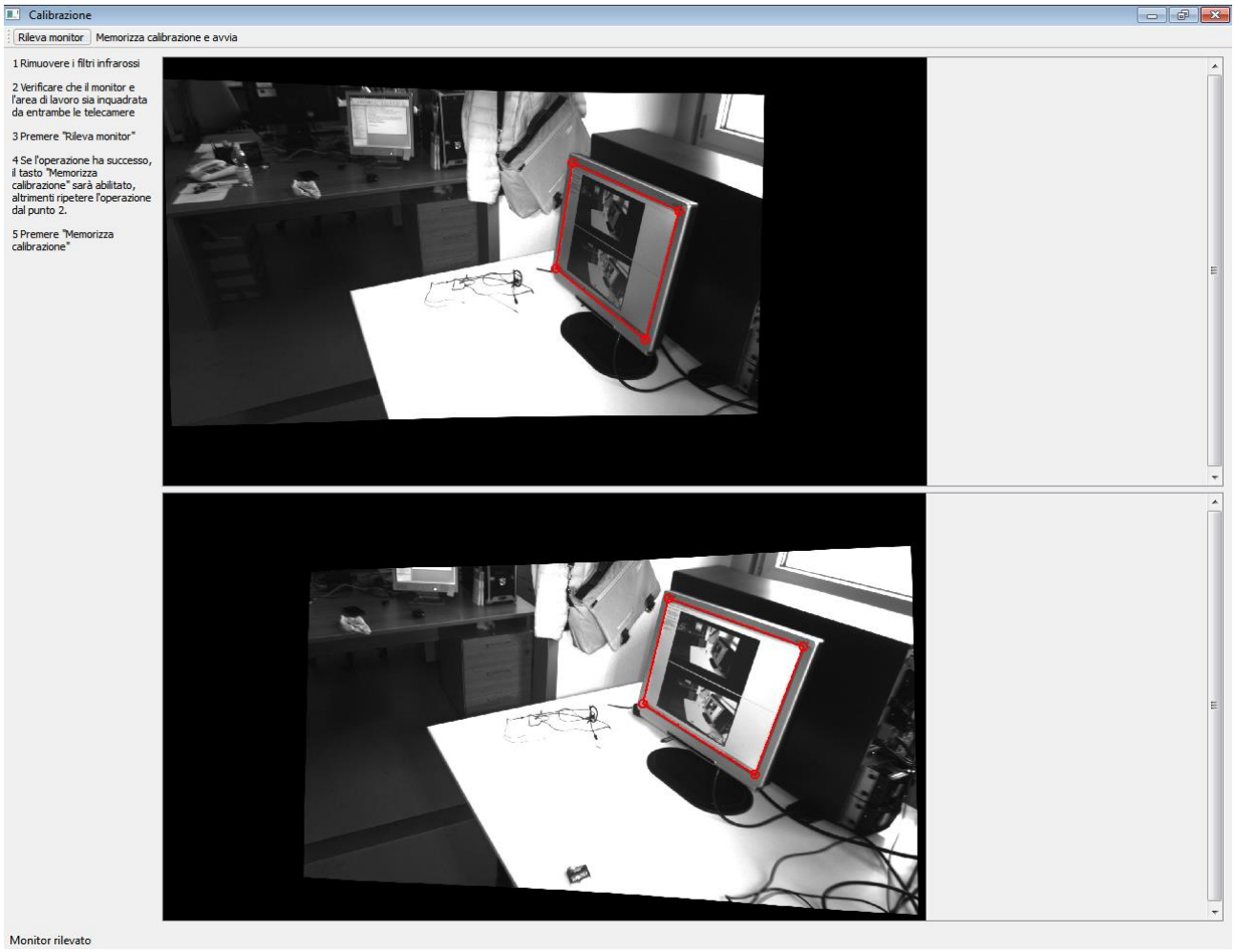


Figura 8 - Schermata di calibrazione effettuata

In fase di avvio, attraverso il pulsante “Verifica e avvia” l’operatore ha la possibilità di controllare l’ultima calibrazione effettuata ed avviare direttamente l’applicazione.

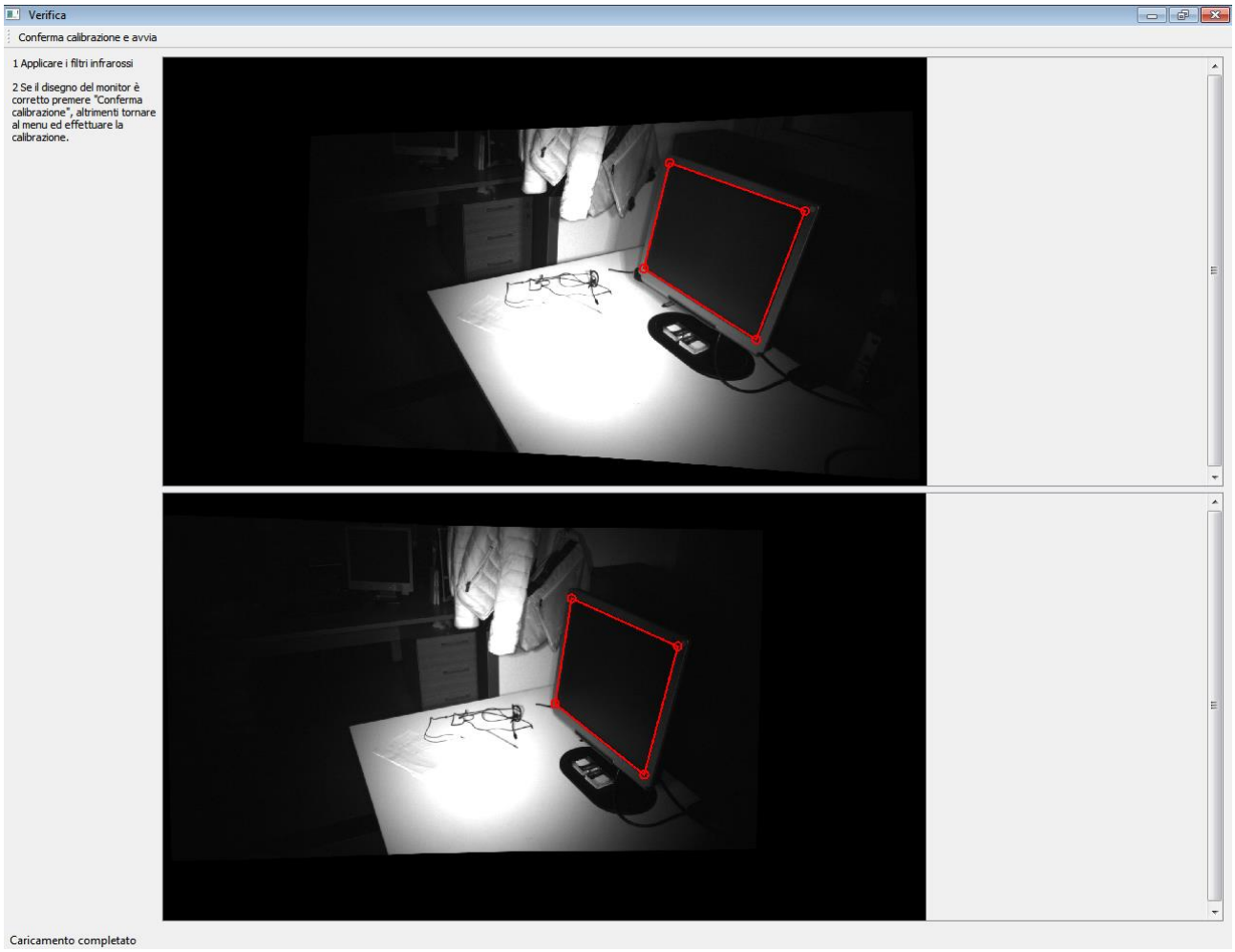


Figura 9 - Schermata di verifica della calibrazione

Avviata l'applicazione, ricompare la schermata iniziale. Lo stato dell'applicazione è "PRONTO".

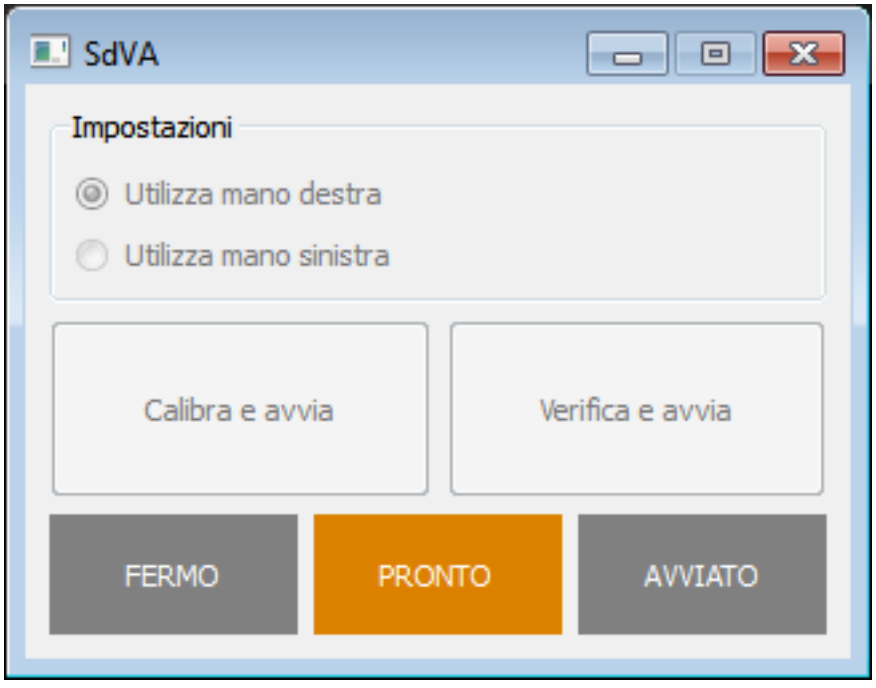


Figura 10 - SdVA in attesa di avvio della fase di gioco

Quando il gioco sarà avviato, l'applicazione di visione artificiale passerà sullo stato "AVVIATO" e invierà i dati di gioco elaborati in tempo reale.

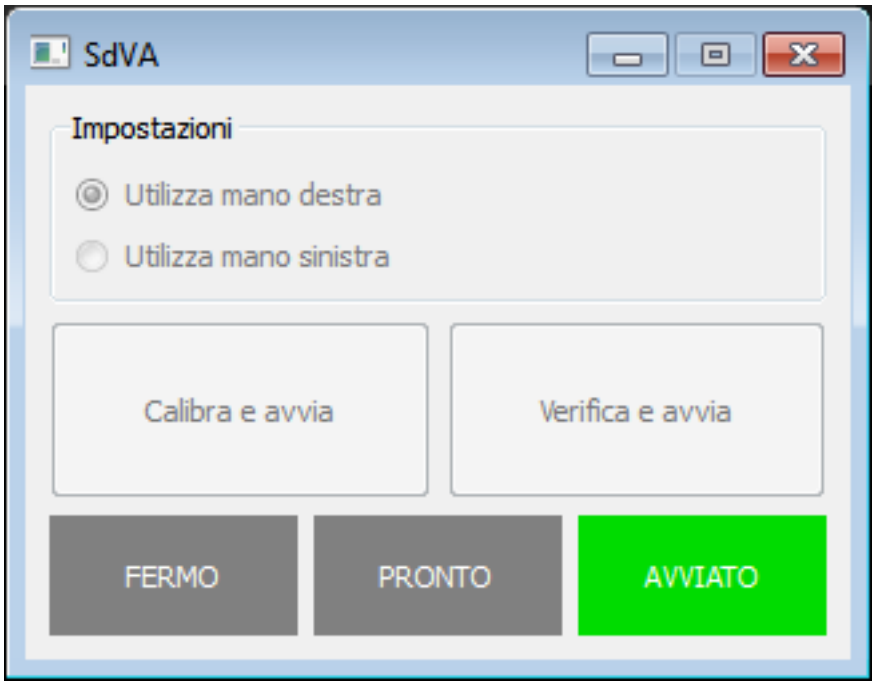


Figura 11 - SdVA in fase di gioco

5.2 Definizione della logica applicativa

Sulla base delle considerazioni fatte nei capitoli precedenti, si riporta il diagramma di stato del SdVA in cui sono riportati gli stati dell'applicazione, gli eventi che generano un cambiamento di stato nonché l'utilizzo delle code in ogni stato.

In dettaglio si evince che:

- all'avvio il SdVA è in fase 0: in questa fase l'applicazione è in attesa di un intervento dell'operatore (che può effettuare la calibrazione o confermare l'ultima effettuata); in questa fase il SdVA non invia alcun messaggio sulla coda di controllo con il Sensor Hub
- confermata la calibrazione e verificata la corretta inizializzazione del frame grabber, il SdVA passa alla fase 1; risponderà con "state" pari a "ready" al messaggio "state" del Sensor Hub
- soltanto la ricezione del messaggio "state" pari a "start" inviato dal Sensor Hub porta il SdVA in fase 2; in questa fase il SdVA invia sulla coda verso il gioco i dati elaborati
- un messaggio "state" pari a "interrupt" o "exit" del Sensor Hub porta il SdVA in fase 1; da qui è possibile chiudere l'applicazione con un messaggio "state" pari a "shutdown".

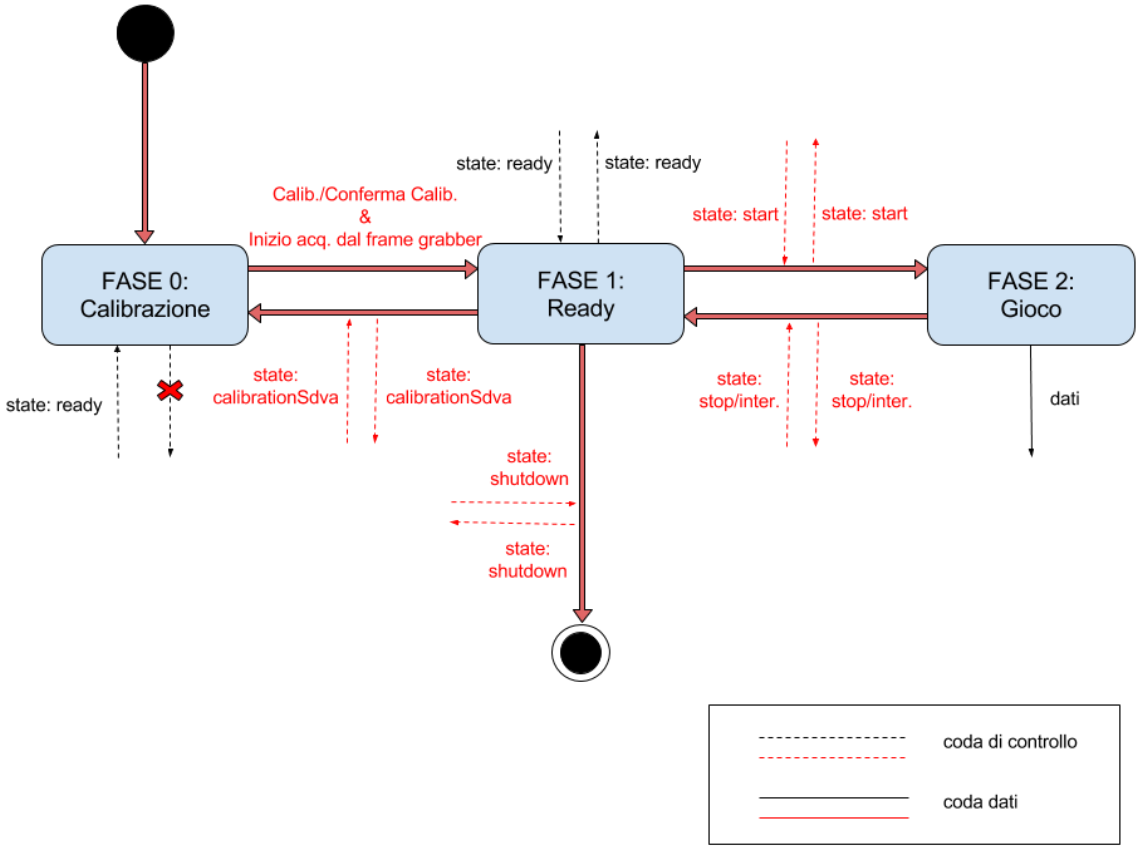
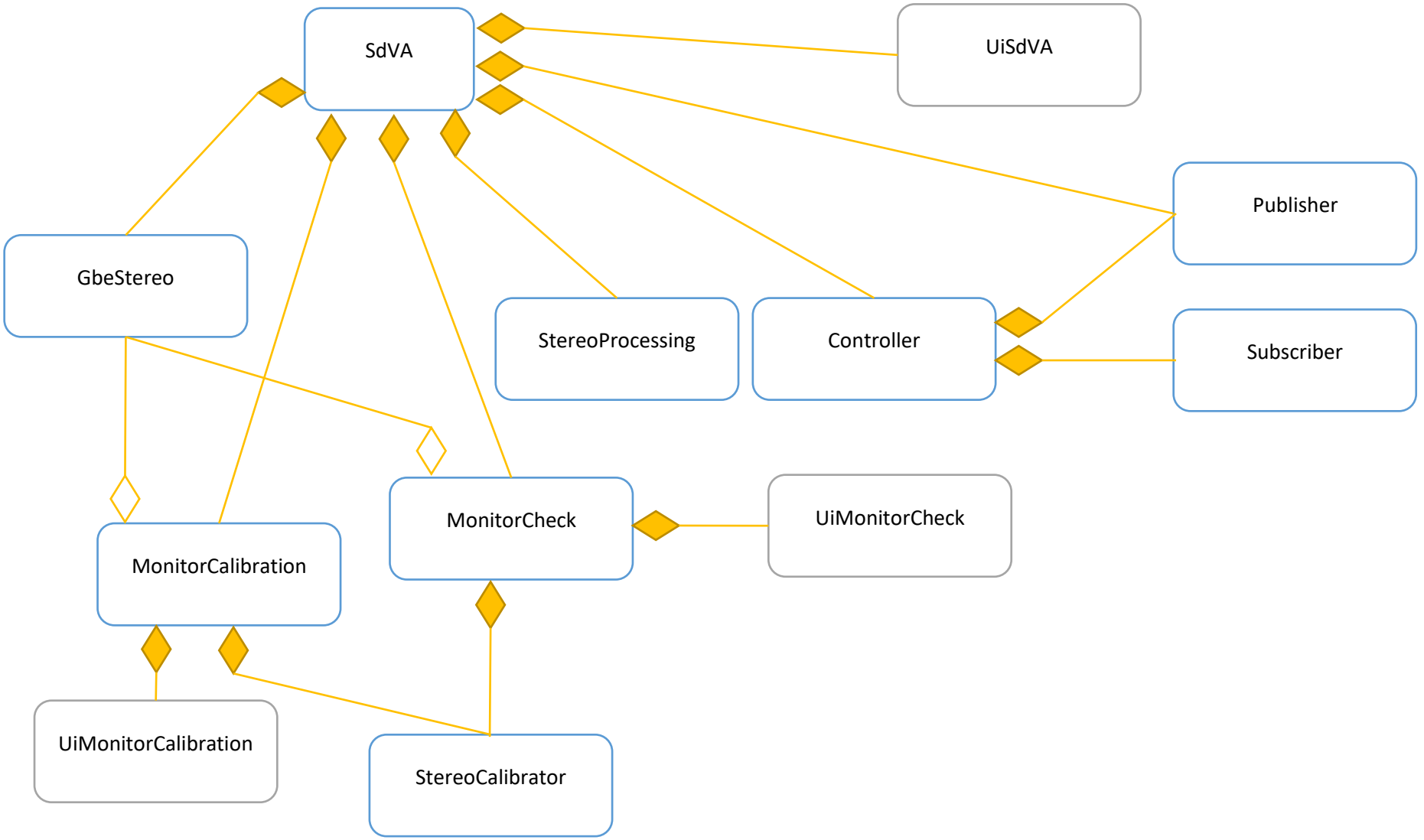


Figura 12 - Diagramma di stato del SdVA

Il diagramma delle classe mostra i soggetti principali coinvolti nell'applicazione di visione artificiale e le loro funzionalità.


SdVA
UiSdVA
Publisher
GbeStereo
StereoProcessing
Controller
Subscriber
MonitorCheck
UiMonitorCheck
MonitorCalibration
UiMonitorCalibration
StereoCalibrator

Segue una descrizione delle singole classi:

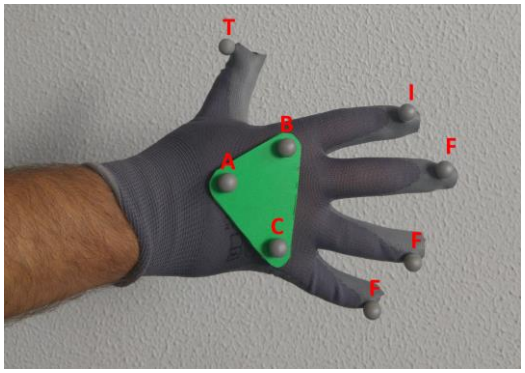
- SdVA: è la classa interfaccia dell'intero sistema di visione artificiale:
  - gestisce la UI principale
  - alloca le risorse e coordina i moduli di calcolo
  - memorizza lo stato dell'applicazione e gestisce le transizioni di stato
  - si occupa dell'invio dei dati di gioco
- UiSdVA: è la classe che definisce la UI principale
- GbeStereo: è la classe che si occupa della gestione del frame grabber; essa fornisce i dati elaborati dall'unità di calcolo (FPGA) presente sul dispositivo
- StereoProcessing: è la classe che si occupa della ricostruzione 3D dei punti individuati dal processing sul frame grabber nonché del riconoscimento della gesture effettuata dall'utente
- Controller: è la classe che gestisce la comunicazione con il Sensor Hub
  - notifica lo stato dell'applicazione al Sensor Hub
  - propaga la richiesta di transizione di stato dal Sensor Hub all'applicazione
- Publisher: è la classe che si occupa della pubblicazione dei messaggi al broker implementando il protocollo AMQP
- Subscriber: è la classe che si occupa della sottoscrizione dei messaggi dal broker implementando il protocollo AMQP
- MonitorCalibration: è la classe che si occupa della calibrazione del monitor
- UiMonitorCalibration: è la classe che definisce la UI della calibrazione del monitor
- MonitorCheck: è la classe che offre la funzionalità della verifica dell'ultima calibrazione effettuata
- UiMonitorCheck: è la classe che definisce la UI della verifica della calibrazione
- StereoCalibrator: è la classe che espone le funzionalità necessarie per la calibrazione del monitor nonché della verifica della stessa

6. Riconoscimento delle gesture

Sulla base delle considerazioni fatte riguardo alle gesture da riconoscere, il SDVA deve fornire al Sensor Hub ogni 20 ms:

- le coordinate del pixel puntato
- il numero di dita che il paziente ha aperto

Per poter estrapolare tali informazioni, si è rivelato indispensabile effettuare come prima operazione, la classificazione dei markers della mano in base al modello mostrato in Figura 13.



- **A**: centro del palmo della mano
- **B**: nocca dell'indice
- **C**: nocca del mignolo
- **T**: punta del pollice
- **I**: punta del dito indice
- **F**: punta del generico dito (medio, anulare e mignolo)

Figura 13 Modello della mano adottato per il labelling dei markers

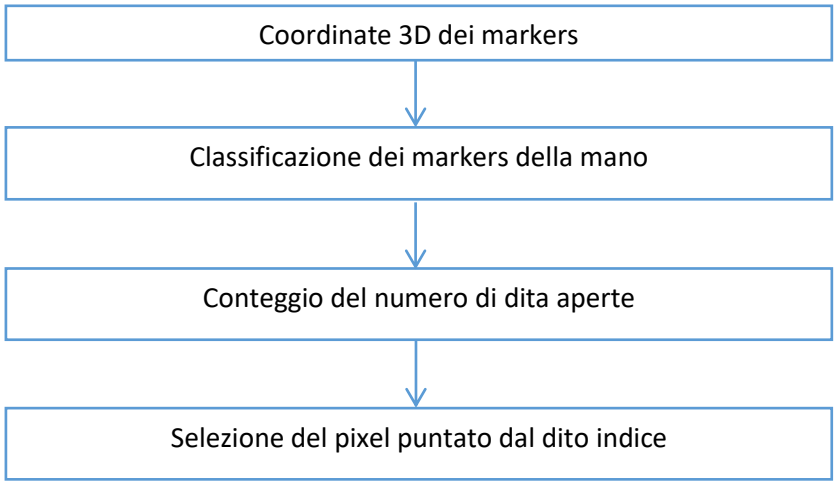


Figura 14 Flow chart concettuale dell'algoritmo di riconoscimento delle gesture

6.1 Classificazione dei markers della mano

La classificazione dei markers della mano avviene in tre step. Prima viene effettuata una classificazione grossolana che distingue i markers del dorso della mano da quelli delle dita e poi, tra i markers delle dita viene individuato quello relativo alla punta del pollice ed infine quello corrispondente alla punta dell'indice.

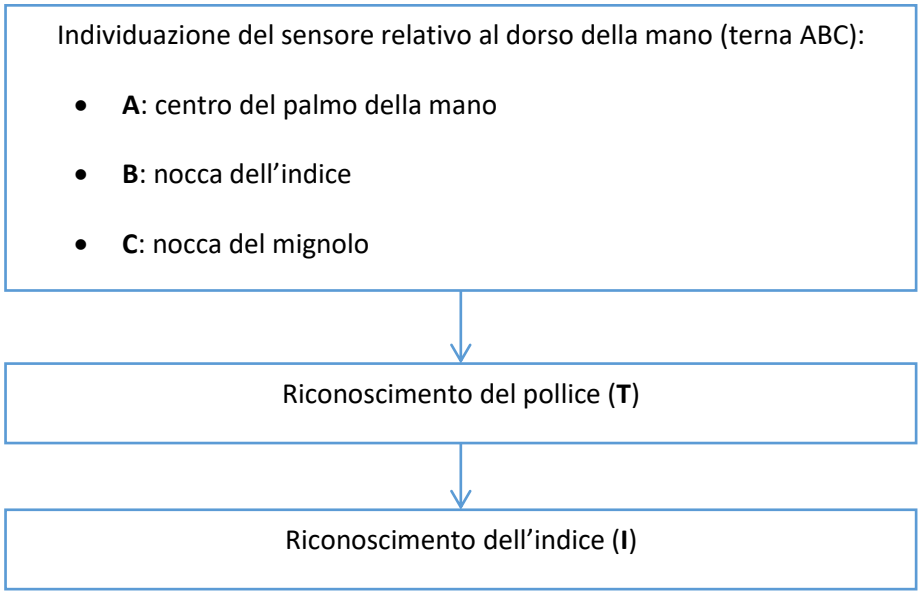


Figura 15 Flow chart concettuale dell'algoritmo di classificazione dei markers della mano

Il motivo per cui, come prima operazione, si ricerca il sensore costituito dai markers A, B e C, è legato al modello di mano adottato nella deliverable D10.1. Infatti, come è possibile osservare in Figura 13, i markers A, B e C, sono stati posizionati su un supporto rigido (di forma triangolare e di colore verde), questo fa sì che le loro distanze reciproche sono fisse e non variano al variare dell'utente che indossa il guanto. Tramite queste distanze è possibile determinare in modo univoco il sensore e quindi il dorso della mano del paziente. I markers rimanenti, sono invece quelli relativi alle punte delle dita.

Mentre per la ricerca dei tre markers A, B e C, sono state sfruttate le coordinate 3D dei punti, per determinare i markers associati alla punta del pollice e dell'indice, si è scelto di lavorare sulle coordinate bidimensionali.

Per individuare il pollice si sfrutta la geometria della mano (vedi Figura 16), è cioè il fatto che questo si troverà sempre nel semipiano che ha come origine la retta AB e che non contiene il punto C. Le restanti dita si troveranno invece nel semipiano complementare.

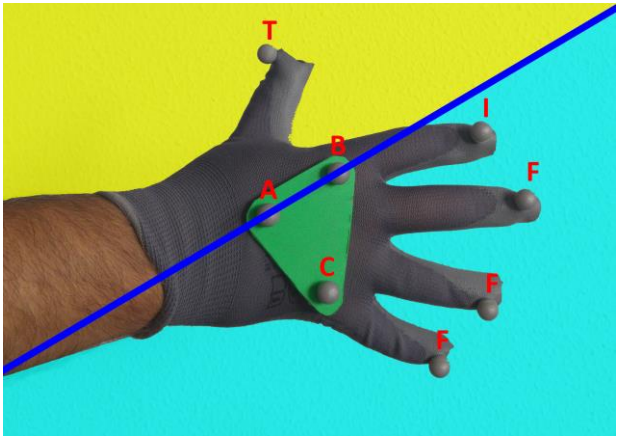


Figura 16 Geometria della mano. Il pollice è sempre contenuto nel semipiano che ha come origine la retta AB e che non contiene il punto C (semipiano evidenziato dal colore giallo).

La ricerca dell'indice si basa sull'osservazione che intersecando la retta contenente il segmento BC, con i segmenti che hanno come prima origine il marker A e come seconda origine la punta delle dita diverse dal pollice, l'intersezione più vicina al punto B, corrisponde a quella ottenuta dall'indice.

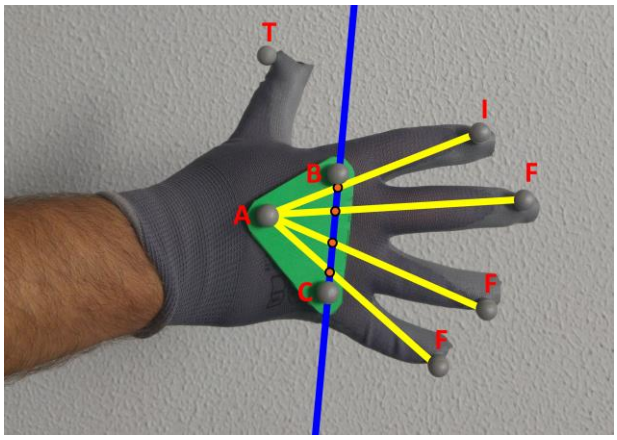


Figura 17 Geometria della mano. Se si intersecano i segmenti che hanno come primo estremo il marker A e come secondo estremo la punta delle dita diverse dal pollice, l'intersezione più vicina al punto B, corrisponde a quella ottenuta dall'indice.

La Figura 18 mostra il risultato dell'algoritmo di classificazione del SDVA.

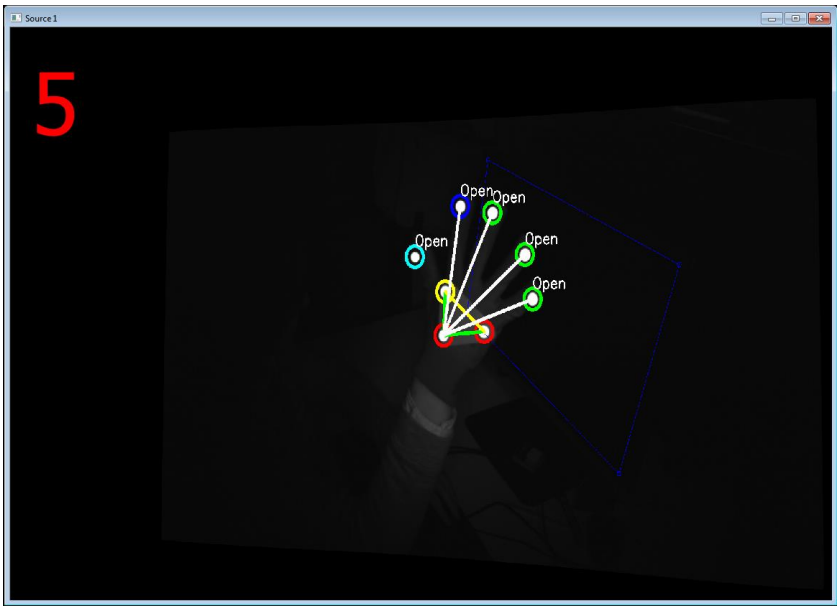


Figura 18 Classificazione dei markers della mano. In figura si può osservare come il SDVA ha classificato correttamente i markers. Il sensore del dorso della mano è stato riconosciuto correttamente, infatti sono stati colorati di rosso i markers A e C e di giallo il marker B, corrispondente alla nocca dell'indice. In celeste è stato evidenziata la punta del pollice, in blu la punta dell'indice e in verde la punta del medio, dell'anulare e del mignolo.

6.2 Calcolo del numero di dita aperte

Per capire se un dito della mano è aperto o chiuso, si sfruttano i versori normali ai due piani:

- Piano del dorso della mano, contenente i tre punti A, B e C
- Piano del dito, contenente i punti B, C e X, dove X corrisponde ai marker delle punta delle dita diverse dal pollice

Nel caso del pollice, il piano del dito che si considera è diverso e contiene i punti A, B e T. Le dita si considerano aperte se l'angolo sotteso dai due versori è maggiore di una soglia, che è stata fissata in base all'esito di diversi test.

Di seguito vengono mostrati alcuni esempi che testimoniano la robustezza dell'algoritmo implementato al variare della posizione della mano rispetto alle telecamere.

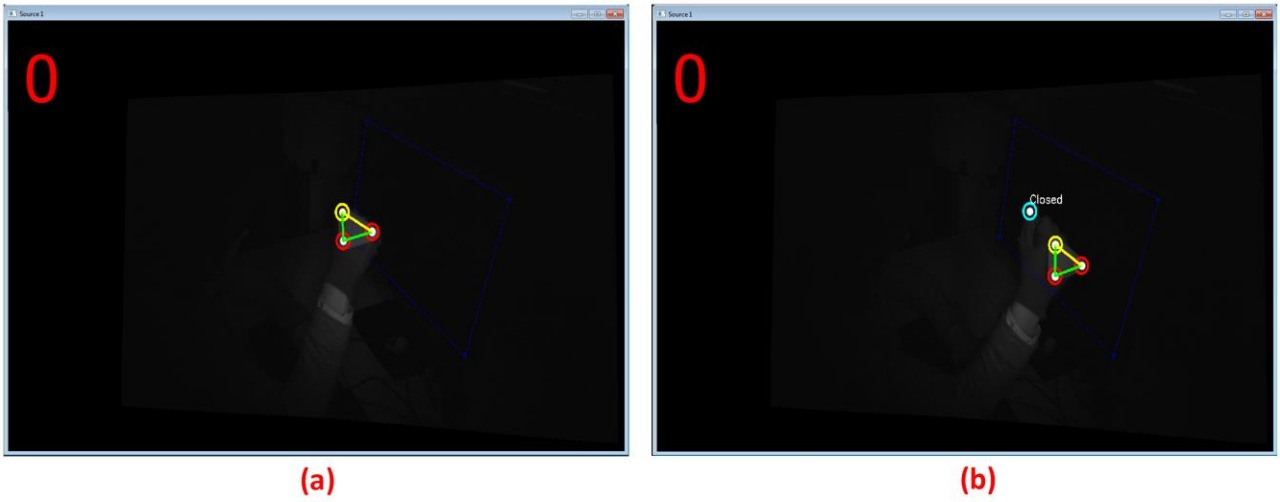


Figura 19 Gestualità: << 0 dita aperte >>. In figura (a) è semplice riconoscere che non ci sono dita aperte, in quanto non compaiono nell'immagine i markers relativi alle punta delle dita. In figura (b) è evidente che il SDVA,

per riconoscere che il paziente ha tutte le dita chiuse, ha dovuto rilevare che il pollice, il cui marker è visibile nell'immagine acquisita, è chiuso.

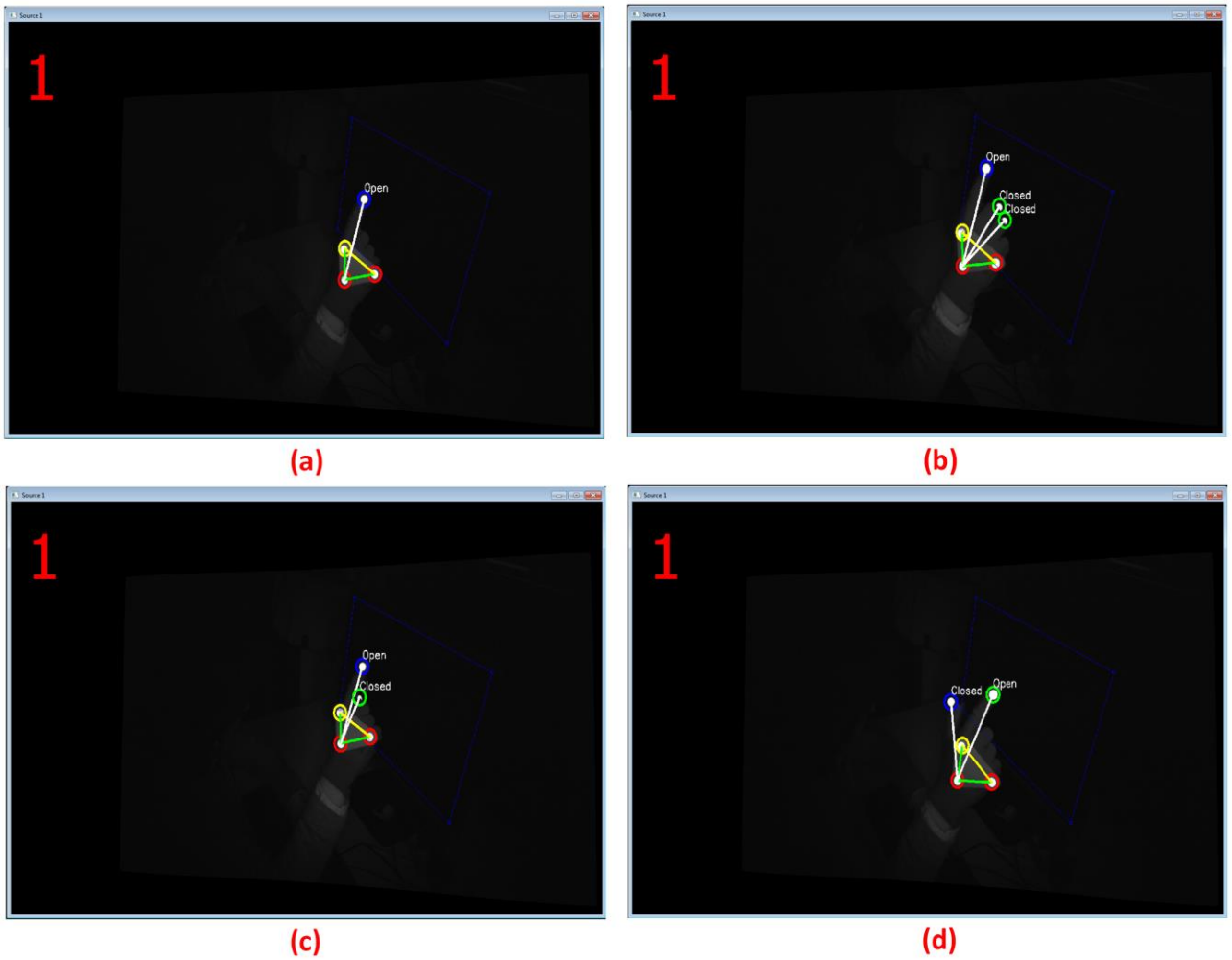


Figura 20 Gestualità << 1 dito aperto >>. In figura (a) il numero di dita aperte coincide con il numero di dita presenti nell'immagine. Nelle figure (b), (c) e (d) è evidente che il SDVA, per riconoscere che il paziente ha un solo dito aperto, ha dovuto rilevare quali, tra le dita presenti nell'immagine acquisita, sono chiuse. Nell'immagine (b) il dito medio e l'anulare, anche se visibili, sono chiusi. Nelle immagini (c) e (d), il pollice è chiuso.

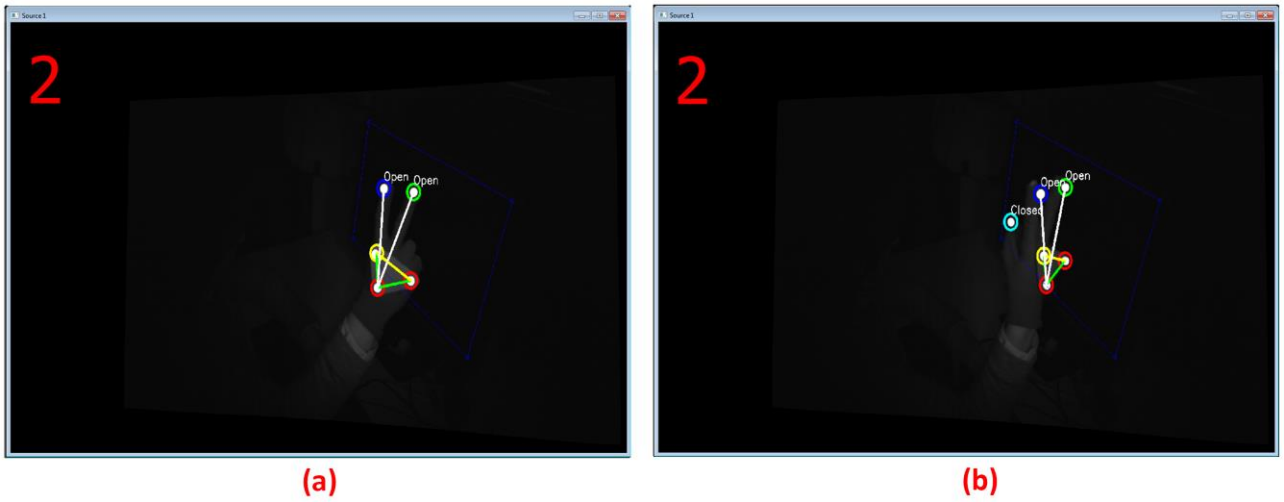


Figura 21 Gestualità << 2 dita aperte>>. In figura (a) il SDVA riconosce che entrambe le dita presenti nell'immagine sono aperte, mentre nella figura (b) verifica che tra le tre dita visibili, il pollice è chiuso e quindi il numero di dita aperte rimane 2 (numero in rosso in alto a sinistra).

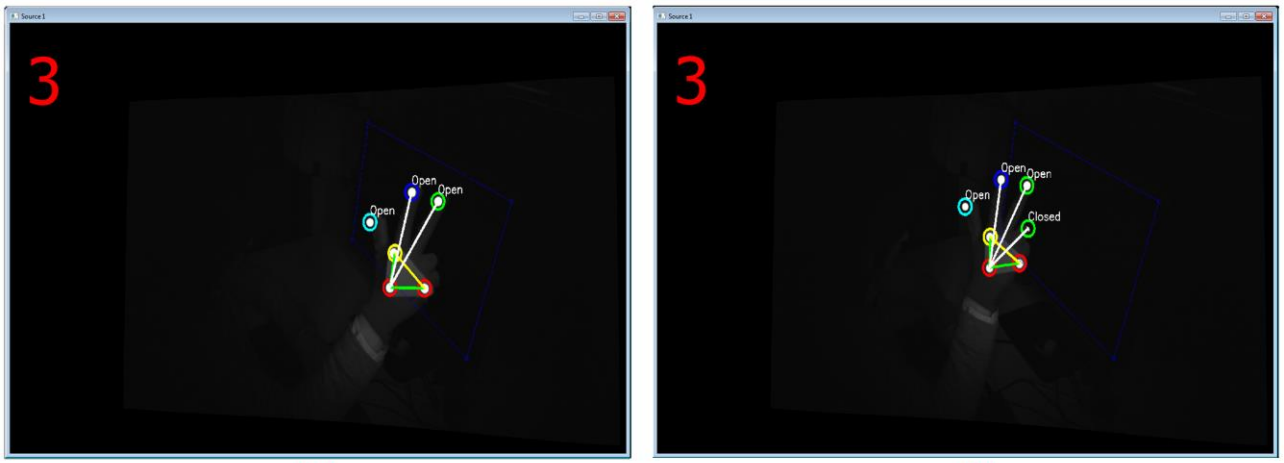


Figura 22 Gestualità << 3 dita aperte >>. Il SDVA rileva che tutte e tre le dita presenti in figura (a) sono aperte, mentre, tra le quattro dita presenti in figura (b), il dito medio è chiuso.

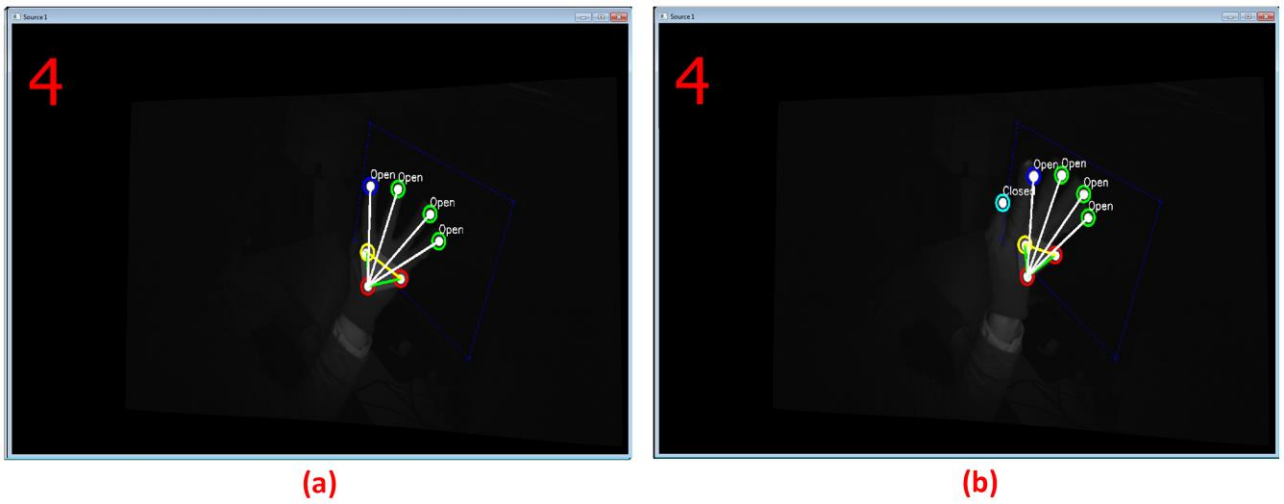


Figura 23 Gestualità << 4 dita aperte >>. Il SDVA, in figura (a), rileva che tutte è quattro le dita presenti nell'immagine acquisita sono aperte, mentre in figura (b) evidenzia che tra le cinque dita visibili, il pollice è chiuso e quindi il numero di dita aperte rimane 4.

6.3 Selezione del pixel puntato dal dito indice

È importante osservare che la direzione di puntamento non corrisponde esattamente con la direzione della retta individuata dai markers B ed I (vedi Figura 13), in quanto i markers sono delle sfere con un diametro di circa un centimetro, che quindi introducono un offset di circa mezzo centimetro nell'operazione di selezione del pixel. Risulta quindi necessario correggere tale offset prima di calcolare l'intersezione tra la retta del dito indice e il piano dello schermo. Le coordinate 3D del punto di intersezione, vengono poi convertite in termini di pixel dello schermo.

Figura 24 Flow chart concettuale dell'algoritmo di selezione del pixel puntato dal dito indice

La Figura 25 mostra il risultato dell'algoritmo di riconoscimento delle gesture. Come è possibile osservare, nel monitor è indicato sia il numero di dita aperte (numero di colore rosso in alto a sinistra), che il pixel puntato dall'utente rappresentato come con il simbolo di un mirino di colore rosso.

Figura 25 Gestualità << Puntamento >> Il pixel puntato dall'utente è evidenziato con il simbolo di un mirino rosso.

7. Test e prestazioni

Al fine di testare e verificare le comunicazioni che l'applicazione dovrà gestire con il Sensor Hub e con il Game, sono stati sviluppati degli stub per questi agenti.

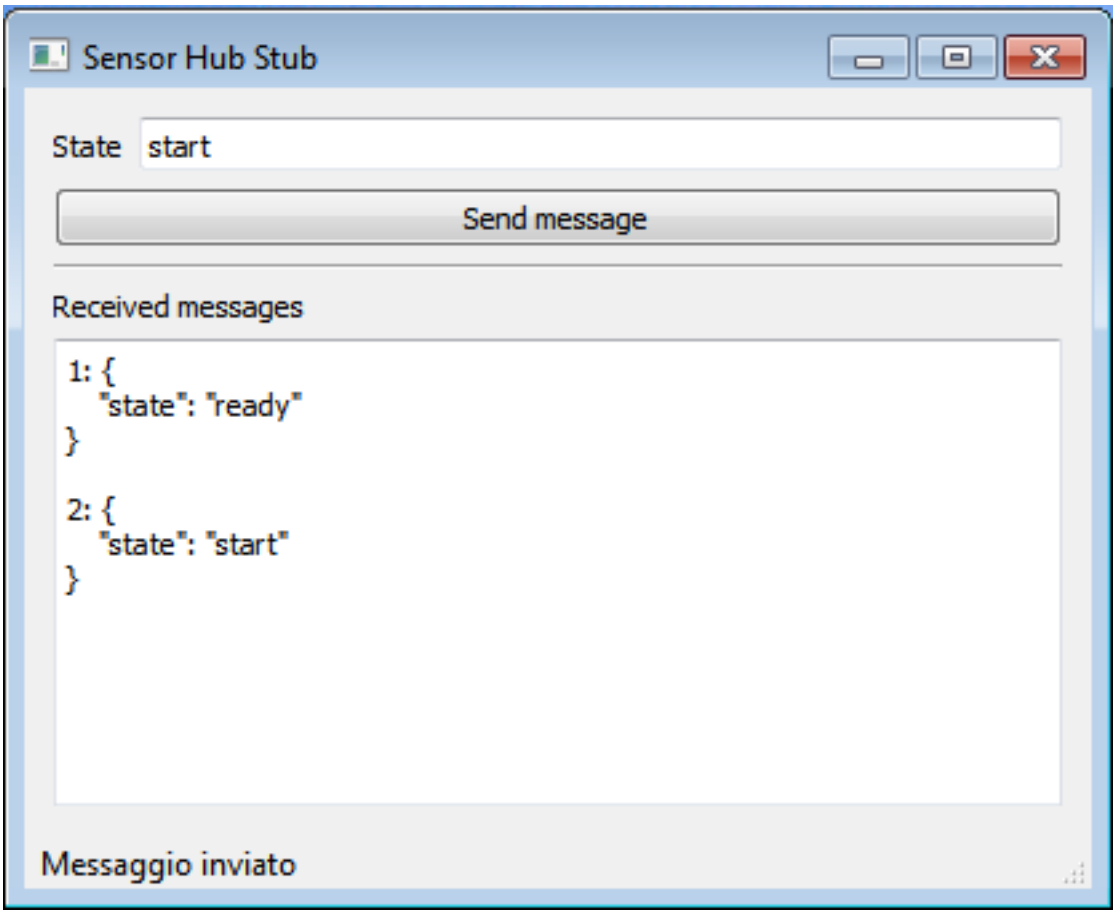


Figura 26 - Interfaccia dello stub del Sensor Hub

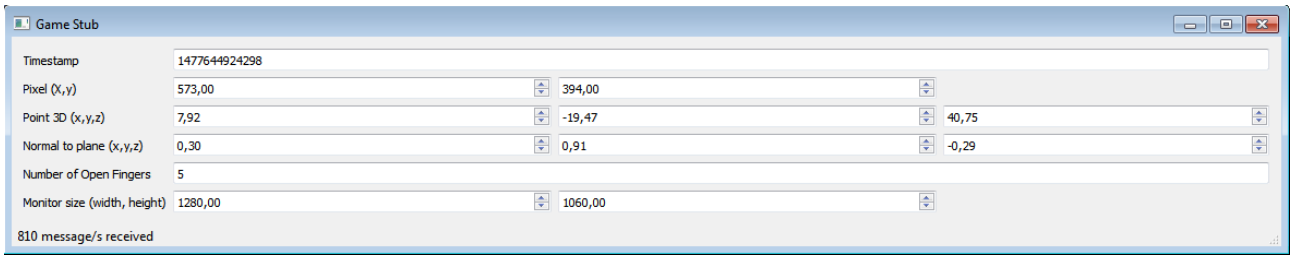


Figura 27 - Interfaccia dello stub del Game

È inevitabile che la comunicazione con gli altri agenti introduca un ulteriore calcolo nonché una latenza aggiuntiva nell'elaborazione delle coppie di immagini. La quantità di messaggi che il SdVA dipenderà dalla latenza della comunicazione che sarà inversamente proporzionale alla banda di trasmissione dati disponibile.

L'applicazione è stata progetta facendo un massiccio utilizzo di thread per effettuare le operazioni di comunicazione in modo asincrono. In questo modo, inevitabili latenza nella comunicazione non impattano sulle prestazioni del SdVA.

Analizzando i dati ottenuti duranti i test si evince che per l'85% del tempo di utilizzo il SdVA ha inviato dati di gioco con una velocità di 50 pacchetti al secondo (uno per ogni coppia di immagini elaborate).

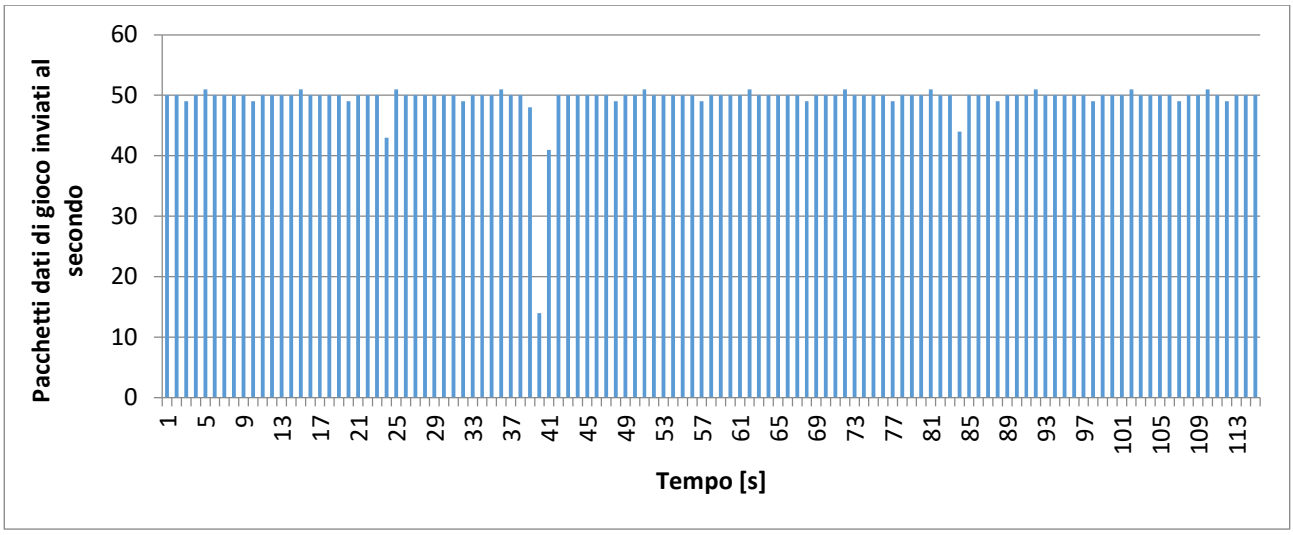


Grafico 1 – Velocità di trasmissione dei pacchetti di gioco

Dall'analisi delle prestazioni si evince che l'aggiunta di altri componenti (Ui, comunicazione e gestione dello stato) non ha ripercussioni sulle performance del SdVA, infatti solo nello 0,05% dei casi il tempo di elaborazione di una coppia di frame è stato superiore ai 20 ms (periodo di acquisizione).

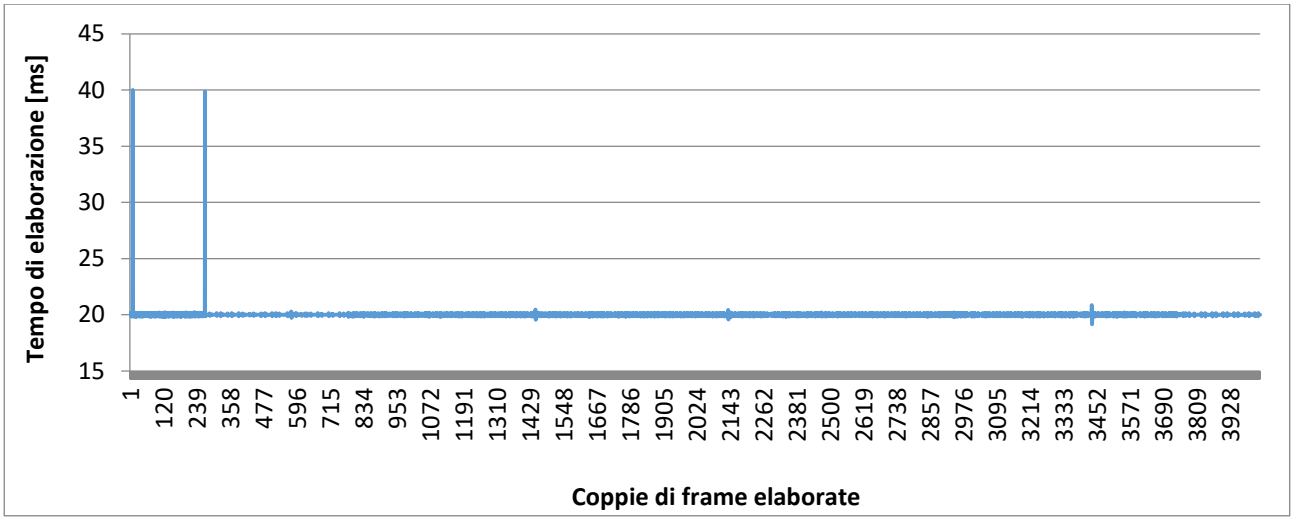


Grafico 2 - Prestazioni del SdVA

Da ciò si deduce che il calo della velocità di trasmissione dei pacchetti dei dati di gioco è causato unicamente dalla latenza della comunicazione. L'applicazione è stata progettata in modo che i pacchetti non inviati vengono immediatamente scartati, evitando quindi di accumulare un carico di lavoro da smaltire in seguito. Questo potrebbe comportare una riduzione della fluidità del movimento nel gioco ma eviterà un effetto ritardo.

8. Integrazione con il Sensor Hub e test di fruizione immersiva del game

Il sistema è stato infine integrato con il Sensor Hub e testato facendo vivere l'esperienza del game in modo immersivo.

Per fare ciò si è usato un video proiettore, calibrando il SdVA in modo da ricostruire sullo schermo di proiezione (una parete) la posizione puntata dal dito del giocatore.

L’integrazione ha permesso di verificare il completo successo dell’intera attività che ha consentito la realizzazione di un interfaccia intuitiva e semplice da usare anche da persone anziane, affette da disturbi neurodegenerativi che possono così interagire con il serious game gesticolando e indicando senza usare mouse, joystick o altri dispositivi di puntamento poco familiari, anche su un grande schermo a parete, vivendo una esperienza immersiva.

Un video dimostrativo dell’esperimento è stato prodotto e allegato al documento.

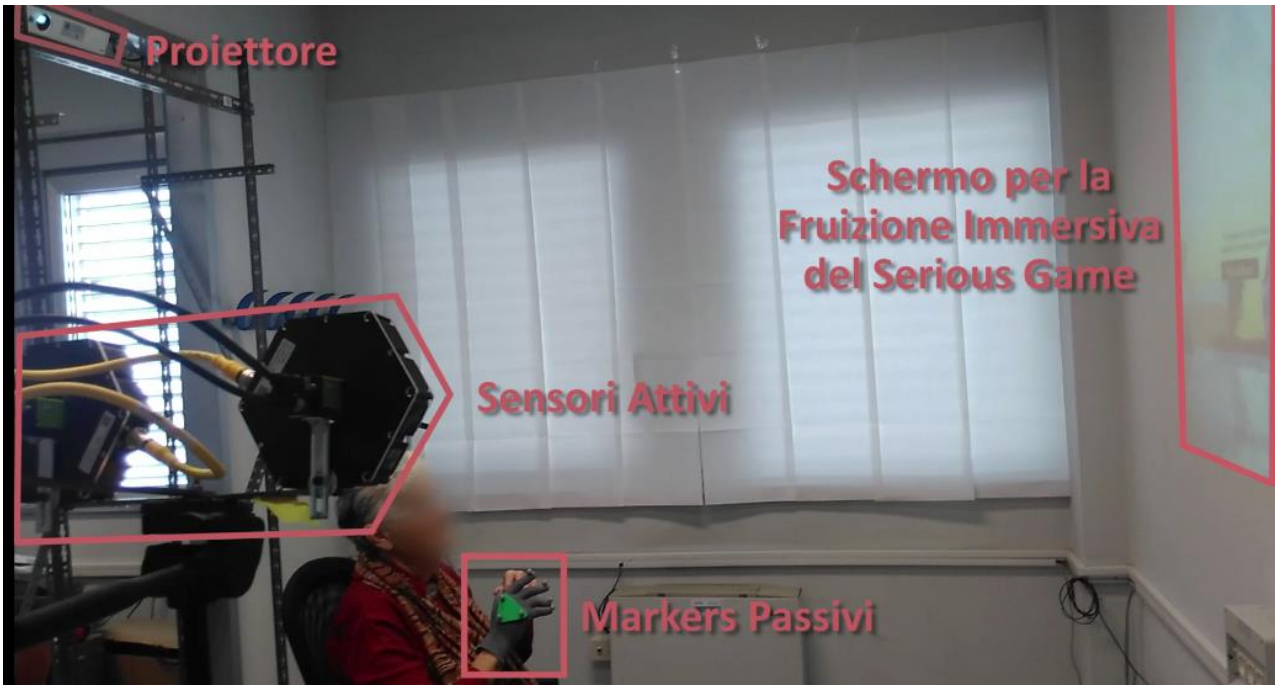


Figura 28 – Installazione del dimostratore in un contesto immersivo

Figura 29 – Fruizione del serious game. Nel riquadro, il particolare dei marker passivi come inquadrati dal SdVA.