
**UNIVERSITA' POLITECNICA DELLE MARCHE**

FACOLTA' DI INGEGNERIA

Corso di Laurea triennale in Ingegneria informatica e dell'automazione

# Elaborazioni di segnali video per il riconoscimento di abitudini alimentari utilizzando il sensore kinect

# Video signals processing for the recognition of food intake using kinect sensor

*Relatore:*

**Prof. Ennio Gambi**

*Correlatore:*

**Ing. Manola Ricciuti**

*Tesi di Laurea di:*

**Giulia Scoccia**

*Anno Accademico 2020 / 2021*

# Indice

# 1. Introduzione

Uno dei principali fattori per la salute di un individuo è una corretta e sana alimentazione, che incide nel processo di sviluppo di una persona e dell'intera popolazione.
Nei paesi industrializzati, in particolar modo negli stati occidentali, l'obesità è diventato un problema su scala globale: già nel 2007, 300 milioni di persone erano obese e un miliardo di adulti erano in sovrappeso. Numeri che nei successivi 8 anni sono raddoppiati.
Obesità e sovrappeso conducono a problemi di salute, come il diabete di tipo 2, disturbi della digestione, malattie cardiovascolari, depressione e tumori.
Se per i giovani seguire una dieta equilibrata è difficile, vista la vita frenetica che porta all'acquisto di cibo proveniente dai fast food, per le persone anziane lo è ancora di più.
Dati i progressi della medicina, l'età media della popolazione tende ad aumentare e patologie come demenza, sclerosi o Alzheimer possono colpire la fascia di età più alta, aumentando la domanda di servizi di assistenza all'individuo.
Sono molte le famiglie ad aver una persona anziana in casa che non è più in grado di prendersi cura di sé stessa ed ogni nucleo cerca di trovare una soluzione consona alle proprie esigenze.
Quando il ricovero in una struttura o l'assunzione di una badante h24 non è possibile o non necessario, nasce l'esigenza di trovare una soluzione per monitorare l'individuo nelle attività quotidiane, come bere e mangiare [1].

Negli ultimi anni la comunità scientifica si è mossa per trovare un sistema adeguato a monitorare la persona, utilizzando dispositivi installati all'interno delle abitazioni (smart home).
Nel 2008 nasce un programma di ricerca europeo chiamato "***Ambient Assisted Living***", terminato nel 2013 e portato avanti come "***Active Assisted Living***", basato sull'art. 185 del trattato sul funzionamento dell'Unione Europea.
AAL è focalizzato su tematiche di ricerca volte verso le tecnologie innovative di assistenza agli anziani in ambiente domestico. I settori coinvolti sono le telecomunicazioni, l'informatica, le nanotecnologie, i microsistemi, la robotica e i nuovi materiali. [2]
Questo programma garantisce:

- **Salute e protezione**: supportare l'anziano nelle azioni quotidiane monitorandolo per tutta la giornata.
- **Privacy e sicurezza**: il sistema deve risultare il più "discreto" possibile, rispettando la privacy della persona.
- **Ambiente sociale e comunicazione**: è necessario che sia garantita la possibilità di interazione

con altre persone, dentro e fuori l'abitazione.

- **Supporto alla mobilità**: un'applicazione AAL può essere usata anche come supporto agli anziani fuori dalle loro case tramite funzionalità come "navigatore" per istruzioni sull'orientamento.

Tali sistemi potrebbero evitare, in molti casi, il ricovero presso strutture sanitarie come ospedali o case di cura, permettendo una migliore qualità della vita e un risparmio dal punto di vista economico per il nucleo familiare che si occupa dell'anziano.

La mia tesi ha come obiettivo il monitoraggio e l'automazione del controllo della nutrizione, confrontando le varie acquisizioni effettuate con il sensore Kinect v1 di casa Microsoft a 10 soggetti in due modalità: con background statico e dinamico. Le due modalità considerate, sono elaborate ciascuna con frame rate differenti (da 3 a 30 frame per secondo, fps), per il conteggio e monitoraggio del numero delle azioni effettuate dal soggetto sotto test durante il pasto, al fine di valutare quale frame rate minimo può essere sufficiente impostare per ottenere il numero di azioni corretto al minimo costo computazionale.

Molti sono i dispositivi utilizzati per acquisire i dati utili al riconoscimento e all'eventuale valutazione dei movimenti e si distinguono in due principali categorie [3]:

- Dispositivi indossabili o wearable devices (***sensor-based AAL***)
- Dispositivi senza contatto o contactless devices (***visual-based AAL***)

I primi sono sensori fisici come accelerometri e identificatori in radio frequenza (RFID), mentre i secondi si basano sull'installazione fissa di sensori all'interno degli ambienti abitativi dei soggetti che hanno la necessità di essere monitorati.

I sensori contactless sono meno invadenti dei wearable devices, in quanto vengono installati all'interno degli ambienti abitativi degli individui presi in analisi, senza che il soggetto debba ricordarsi di indossarli o accorgersi della loro presenza. Dal punto di vista economico, presentano costi ridotti e sono meno soggetti a guasti rispetto ai sensori fisici.

Fanno parte di questa categoria le camere RGB, che estraggono le immagini (frame) composte da numerosi pixel contenenti l'informazione del colore. Tuttavia, prendendo in esame i frame estratti da questo tipo di camera, il problema del cambiamento o riduzione dell'illuminazione in un ambiente chiuso dovuti all'alternanza giorno/notte, possono ridurre l'efficienza di un sistema AAL.

I cambiamenti di illuminazione possono causare problemi per molti metodi di sottrazione di background.

L'informazione RGB subisce in negativo la variazione di luminosità al contrario della telecamera ad infrarossi, che consente di intercettare gli oggetti anche al buio. L'informazione RGB, essendo simile

al sistema di visione umano, può essere utile per classificare oggetti come bicchieri o piatti, indispensabili nel monitoraggio del food intake.

Un'immagine di profondità (depth frame) è invece un canale in cui ogni pixel dell'immagine si riferisce a una distanza tra il piano dell'immagine e l'oggetto corrispondente all'immagine RGB. In questo lavoro di tesi sono state considerate solo le immagini di profondità e l'utilizzo del Kinect è stato quello di acquisire tali immagini definite RGB-D (Red, Green, Blue più Depth).

Se l'utilizzo dell'hardware Kinect non fosse stato disponibile per condurre i test in laboratorio, sarebbe stato necessario stimare la profondità delle immagini proveniente dalla stessa scena, ma scattata da più telecamere, arrivando a un aspetto di computer vision.

In generale, se la camera è in posizione fissa, l'area monitorata risulterà ridotta e questo può rappresentare una limitazione nell'utilizzo di questo tipo di sensore, che può essere superata utilizzando più di una camera dove ci fosse bisogno di copertura. Nonostante la grande efficienza negli ambienti chiusi, in spazi aperti non è possibile prendere in considerazione l'utilizzo di sensori contactless, in quanto non riescono a coprire una vasta area. In questi casi, la soluzione potrebbe essere l'utilizzo di entrambi, il cosiddetto approccio data-fusion: combinando le informazioni fornite dai dispositivi indossabili e video è possibile migliorare le prestazioni garantendo un miglior monitoraggio.

Parlando dei wearable devices, la problematica delle grandi distanze viene superata, dato che il dispositivo può essere sempre indossato dalla persona. I dati estratti possono essere elaborati dal sistema andando ad incidere sulla sua batteria oppure inviati con i protocolli wireless (Bluetooth, ZigBee, Wi-Fi Direct) ad elaboratori. Come sensori vengono utilizzati gli accelerometri e gli identificatori in radio frequenza (RFID) che vengono posizionati in entrambe le braccia, sia nella parte alta che nell'avambraccio, avendo cura di non ridurre la mobilità del soggetto.

Le limitazioni di questi dispositivi si presentano quando è necessario monitorare specifici comportamenti della persona, che possono portare a situazioni di pericolo. Un altro impedimento consiste nel fatto che la persona esaminata deve ricordarsi di indossare il dispositivo ed essere quindi consapevole di essere monitorato, rendendo il controllo molto indiscreto; per questo motivo i ricercatori continuano ancora nella ricerca di soluzioni più compatte e meno invadenti (unobtrusive solutions).

In questa tesi viene utilizzato il Kinect di casa Microsoft, che include sia sensori contactless che sensori "fisici". Questo dispositivo è ottimale per questo lavoro, sia per la grande reperibilità sul mercato, sia per il costo ridotto: viene prodotto su scala mondiale per l'ambito gaming, ma viene

impiegato molto nella ricerca scientifica. Ottimo è il rapporto fra la qualità dei sensori montati sul dispositivo e il costo, dell'ordine del centinaio di euro.

### 1.1. Stato dell'arte

Molteplici sono gli studi effettuati mediante dispositivi indossabili e contactless.

I primi sono stati utilizzati in un lavoro presentato nel 2016 [27], dove gli autori hanno utilizzato un dispositivo indossabile portatile, implementato negli occhiali, collegati ad un sensore di deformazione piezoelettrico per il riconoscimento delle attività legate all'alimentazione. Il sistema ha rilevato il 99,85% dei movimenti legati al food intake, risultando molto efficace.

Tuttavia, il monitoraggio della nutrizione effettuato tramite dispositivi indossabili, nonostante l'alta accuratezza nel rilevare l'azione, può risultare impegnativo per soggetti con disabilità fisiche o cognitive. Negli studi successivamente descritti, si è cercato di fornire una valida alternativa ai sensori fisici.

In uno studio [28] gli autori hanno utilizzato gli spazi colore YCbCr e YIQ come trasformazione di RGB, per monitorare soggetti con Alzheimer o demenza durante le acquisizioni di cibo in vista frontale, a differenza del lavoro svolto in questa tesi che è il modalità top view. Hanno analizzato le regioni della pelle per rilevare le azioni con risultati promettenti ma, non permettendo di preservare la privacy del soggetto.

Un altro studio svolto in modalità frontale [29] è stato realizzato nel 2016 dove, al fine di analizzare i movimenti delle persone anziane che si alimentano, viene utilizzato un giovane come riferimento. Gli autori, al fine di stimare i gesti, definiscono la distanza mano – testa come parametro fondamentale per determinare se un'azione si è compiuta o meno.

Nel 2014 [30] hanno proposto un nuovo sistema di rilevamento basato esclusivamente sui dati di profondità di una telecamera RGB-D per il riconoscimento delle attività legate alla nutrizione: utilizzando queste informazioni è possibile risolvere i problemi legati all'illuminazione, fornendo una maggior efficienza e precisione del sistema, garantendo la privacy del soggetto. Parte del lavoro di questa tesi, si basa su uno studio [26] effettuato con il dispositivo kinect v1 in modalità top – view, che porta delle novità nella procedura di identificazione dei movimenti relativi al food intake. La prima riguarda la rilevazione automatica dello start frame e dell'end frame, che corrispondono all'inizio e alla fine di un pasto; la seconda è un miglioramento dell'algoritmo SOM_EX e la terza è la gestione della memorizzazione dei dati utilizzando una nuova interfaccia software.

# 2. Kinect

Il Kinect è un dispositivo di casa Microsoft configurabile per diversi scopi, oltre a supporto della console dei videogiochi Xbox. Grazie alla presenza della camera RGB, camera IR, sensore di profondità e dati relativi allo skeleton, è stato usato non solo nel gaming, ma anche nell'arte, nella musica, nell'ambito militare, scientifico e archeologico [4].

La distribuzione del Kinect è iniziata nel 2010 come periferica per Xbox 360, che ebbe finalmente modo di gareggiare insieme a PS3 e Wii per il miglior sistema di motion tracking. Con l'arrivo di Xbox One, il device fu incluso nell'acquisto della console, offrendo, oltre a una serie di giochi, la possibilità di comandare il gioco in remoto grazie al riconoscimento della voce e dei movimenti del corpo. Purtroppo, la vita di Kinect nel mondo delle nuove generazioni di gaming consoles è durata poco, in quanto le console d'ultima generazione hanno ridotto l'attenzione ai sensori di movimento per orientarsi alla realtà virtuale. Sony ha scommesso su PlayStation VR e Microsoft introdotto la compatibilità di Xbox One con Oculus Rift, così Kinect appare ormai come un accessorio superfluo per il futuro dell'intrattenimento.

È un dispositivo versatile ed è stato usato in altri campi e per diversi scopi, tra cui [5]:

- **Scanner 3D**: la tecnologia integrata "Kinect Fusion" ha permesso di usare il device per scansionare piccoli e grandi oggetti, in maniera low-cost. Una scansione singolare è stata fatta da Nora Al-Badri e Jan Nikolai Nelles nell'ottobre del 2015 al Neues Museum di Berlino: utilizzando il Kinect v1, hanno scansionato in 3D il busto della regina Nefertiti che poi hanno reso pubblica, senza il consenso del museo, compiendo un vero e proprio "furto d'arte" [6].
- **Sorveglianza militare**: un programmatore sud-coreano utilizzò il sensore kinect per controllare il confine tra Corea del Nord e Corea del Sud.
- **Ambito musicale**: un compositore ebbe l'idea di collegare il kinect ad un organo a tubo a 4 piani, avendo così la possibilità di suonarlo senza toccare alcun tasto.
- **Ambito chirurgico**: con il software GestSure è possibile usare delle "gesture", captate ed elaborate da Kinect, per manipolare delle immagini; ad esempio un team di chirurghi durante un'operazione potrebbe aver bisogno di informazioni sul paziente, da reperire con rapidità. Questo sistema permette di semplificare l'operazione tramite l'uso delle mani come input per un PC.

L'hardware de Kinect si basa su tecnologie della 3DV, una compagnia israeliana specializzata in tecnologie di riconoscimento dei movimenti tramite videocamere digitali che Microsoft ha prima

finanziato e poi acquisito nel 2009, e sul lavoro della israeliana PrimeSense, che ha poi dato in licenza la tecnologia a Microsoft. Il software di Kinect invece, è stato sviluppato ai Microsoft Game Studios, più precisamente dai programmatori della Rare, la quale ha dovuto cancellare altri progetti per dedicarsi interamente alla periferica.

L'uscita di Kinect ha provocato un grande movimento e consenso nella comunità scientifica, tanto che nel 2011 Microsoft ha deciso di rilasciare le API (Application Programming Interface) per poter sfruttare al meglio l'hardware del dispositivo e consentire il riconoscimento automatico di una o più persone attraverso lo scheletro (o skeleton), sviluppato come modello anatomico dei giunti del corpo, implementato nel SDK (Software Development Kit) liberamente scaricabile online sul sito ufficiale della Microsoft.

Con 8 milioni di unità vendute nei primi 60 giorni di mercato (4 novembre 2010 – 3 gennaio 2011) è entrato nel Guinness World Record [7].

Per le acquisizioni fatte in laboratorio e utilizzate nel mio elaborato, il sensore viene posto sul soffitto con posizione fissa a 3 metri dal pavimento per inquadrare il soggetto seduto a tavola.

Questa configurazione in top-down view risulta migliore di quella "frontale" utilizzata nei precedenti studi dove il kinect veniva posizionato di fronte al soggetto, sullo stesso tavolo.

Questo posizionamento rappresentava delle limitazioni nelle prestazioni del sistema di riconoscimento e per questo venne sostituita.

Le acquisizioni presenti nella mia tesi sono state effettuate con il Kinect v1. Tuttavia, esiste una versione più aggiornata del dispositivo, il Kinect v2.

Per lo scopo di questa tesi è stata scelta la v1 perché offre un costo computazionale minore avendo una risoluzione più bassa delle immagini, che è uno degli obbiettivi della tesi.

## 2.1 Kinect V1

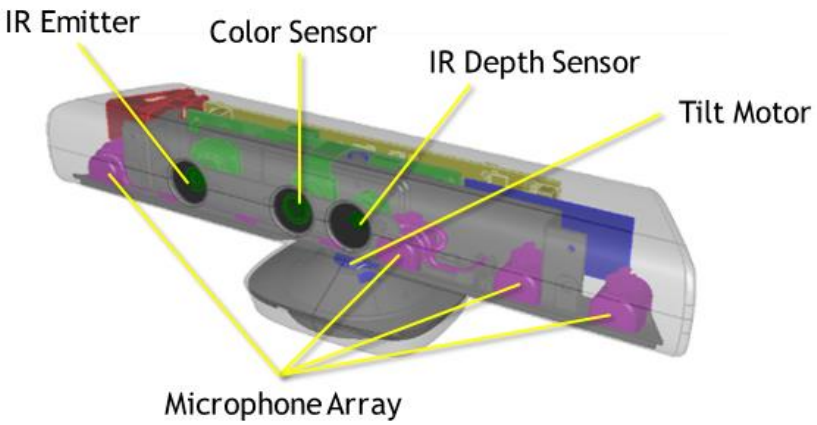


*Figura 1. Componenti kinect v1*

La prima versione del dispositivo è così composta [8]:

1. ***Array di microfoni***: quattro microfoni per catturare e registrare i suoni. Indicano il punto della sorgente e la direzione delle onde sonore.
2. ***IR Emitter***: emette un fascio di raggi infrarossi che, una volta a contatto con una superficie, vengono riflessi.
3. ***IR Depth Sensor***: il sensore di profondità legge i raggi riflessi per convertirli in misurazioni di distanza tra un oggetto ed il sensore. Questo consente di catturare un'immagine di profondità costituita da 640x480 pixel a 30 fps ed a ognuno di questi sarà associato il suo valore di distanza dall'oggetto inquadrato.
4. ***Color Sensor***: camera RGB in grado di catturare immagini a colori con una risoluzione di 640x480 pixel a 30 fps. Insieme alle informazioni rilevate dal sensore di profondità permette di costruire il modello dell'ambiente e degli oggetti o persone inquadrate.
5. ***Tilt Monitor***: un accelerometro a 3 assi che determina l'orientamento del Kinect nello spazio; infatti permette di variare l'inclinazione verticale del sensore stesso.

La posizione del sensore di profondità e dell'emettitore IR permette al dispositivo la visione stereo, cioè l'osservazione avviene da due punti differenti. Tuttavia, possono portare a errori e rumore: i primi derivano dalla precisione dell'emettitore e del sensore, mentre il secondo è dovuto alla presenza di elementi esterni in grado di attenuare o disturbare il percorso degli infrarossi come ad esempio, la luce del sole o altri oggetti molto scuri. Altro limite riguardante la visione stereo si traduce nell'impossibilità di controllare oggetti troppo vicini al sensore di profondità, ma d'altro canto non è in grado di gestire oggetti troppo lontani.
In conclusione, il sistema è in grado di elaborare l'immagine di profondità direttamente all'interno del device, per poi inviare tramite connessione USB 2.0 il flusso dei dati dopo l'installazione dei driver.

### 2.2 Kinect V2


*Figura 2. Componenti kinect v2*

La versione successiva del kinect, chiamata Kinect One, è stata presentata insieme a Xbox One il 21 maggio 2013, durante un evento Microsoft e il suo lancio sarebbe dovuto avvenire il 22 novembre

2013. A causa di alcune controversie e successive modifiche, la periferica venne venduta separatamente a partire dal 6 Ottobre 2014.

Nella seconda versione è stata migliorata la tecnologia dietro la creazione di immagini di profondità, grazie all'introduzione di un sistema chiamato "Time of Flight" [9] (TOF, tempo di volo) .

Il TOF consente una maggior accuratezza nel rilevamento dei dettagli, andando a diminuire drasticamente il rumore che caratterizza la prima versione del dispositivo, sia nelle brevi che nelle lunghe distanze; permette la stima in tempo reale della distanza tra la camera RGB-D e gli oggetti che vengono inquadrati, misurando il tempo che impiega un impulso luminoso per effettuare il tragitto camera-oggetto-camera.

I sensori hanno flussi in uscita con risoluzioni migliorate rispetto alla prima generazione, come si può vedere dalla tabella in cui vengono messe a confronto le caratteristiche delle due versioni [10]:

| Feature | Kinect for Windows 1 | Kinect for Windows 2 |
|---|---|---|
| Color Camera | 640 x 480 @30 fps | 1920 x 1080 @30 fps |
| Depth Camera | 320 x 240 | 512 x 424 |
| Max Depth Distance | ~4.5 M | ~4.5 M |
| Min Depth Distance | 40 cm in near mode | 50 cm |
| Horizontal Field of View | 57 degrees | 70 degrees |
| Vertical Field of View | 43 degrees | 60 degrees |
| Tilt Motor | yes | no |
| Skeleton Joints Defined | 20 joints | 26 joints |
| Full Skeletons Tracked | 2 | 6 |
| USB Standard | 2.0 | 3.0 |
| Supported OS | Win 7, Win 8 | Win 8 |
| Price | $299 | TBD |

*Figura 3. Confronto fra le due versioni.*

Kinect V2 è capace di calcolare l'informazione di profondità, in modo ottimale fino a una distanza dal sensore di 1.5 m, sia in condizioni di luce naturale che artificiale. A distanze più elevate e fino a 3 m, l'informazione di profondità è ancora poco rumorosa, mentre a distanze più elevate dal sensore, il rumore aumenta e la qualità dell'immagine inizia a diminuire sensibilmente.

Kinect V1 per l'estrazione delle immagini di profondità, utilizza una proiezione di pattern di luce IR che non garantisce la stessa resa: mancano dettagli, l'immagine risulta confusa e, in lunghe sequenze di frame, esso tende a "sporcarsi", presentando un notevole rumore diffuso.

Kinect v2 presenta la stessa predisposizione di microfoni direzionali, della versione precedente.

La periferica, grazie alla tecnologia più avanzata, è in grado di rilevare e leggere fino a sei corpi in simultaneo.
Per quanto riguarda l'immagine RGB sicuramente la risoluzione migliore si ha con Kinect V2 che permette infatti di avere un'acquisizione più dettagliata che non sarebbe possibile ottenere con il V1, ma dall'altro lato i file esportati occupano maggior memoria.
L' occupazione di memoria di questo genere di dati è di circa 8Mbyte per ogni singolo frame.
In entrambi i dispositivi le telecamere RGB-D (dove D sta per Depth), sono utili per riconoscere particolari azioni che un soggetto compie in ambiente in-door.

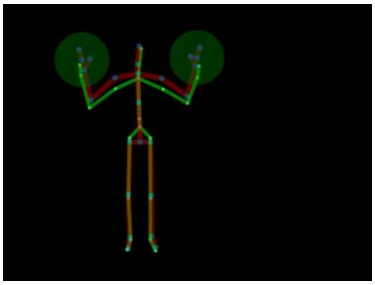

*Figura 4. Confronto scheletri rilevati con il v1 e il v2.*

Mettendo a confronto due scheletri rilevati con il v1 (verde) e con il v2 (rosso e blu), vediamo che la loro posa è simile, ma quella del v2 è più naturale e precisa. Il v1 presenta molti errori di rilevamento e un tempo di risposta del sistema complessivamente più lento.

### 2.3 Alternative

Microsoft ha cessato la produzione di Kinect il 25 ottobre 2017, con all'attivo 35 milioni di unità vendute tra la versione Xbox 360 ed Xbox One.
Nel Maggio del 2018 durante la conferenza Build, Microsoft aveva annunciato nuove innovazioni che avrebbero consentito agli sviluppatori di creare esperienze AI, multi-device e multisensoriali, e lanciò il nuovo programma da 25 milioni di dollari "*AI for Accessibility*" [11]. Nel corso della conferenza annuale dedicata agli sviluppatori, i leader di Microsoft presentarono nuove tecnologie per consentire di diventare facilmente sviluppatori di intelligenza artificiale su Microsoft Azure**,** Microsoft 365 **e** qualsiasi altra piattaforma.
Microsoft propose *Kinect for Azure***,** un pacchetto di sensori progettato per AI, nel quale è inserita una telecamera di profondità di nuova generazione e sono incluse capacità computazionali. Sviluppato sulla base del tradizionale Kinect, tramandato mediante HoloLens, Project Kinect for Azure aprì nuove prospettive agli sviluppatori che tutt'ora operano con l'intelligenza ambientale.
L'unione tra il sensore Time of Flight di Microsoft, che detta gli standard del settore, e altri sensori, tutti in formato ridotto ed efficienti dal punto di vista energetico, permise a Project Kinect for Azure di sfruttare le potenzialità dell'AI di Azure, migliorando così drasticamente sia intuizione che

operatività.

Rilasciato nel mercato a marzo 2020 ad un prezzo lancio di $399, Azure Kinect Developer Kit (DK), al contrario del Kinect, è una periferica per PC che include i più recenti sensori per l'intelligenza artificiale di Microsoft come il sensore di profondità degli HoloLens 2 e 7 array di microfoni per la computer vision e speech. È presente una fotocamera RGB da 12MP e una fotocamera da 1MP per la profondità [12].

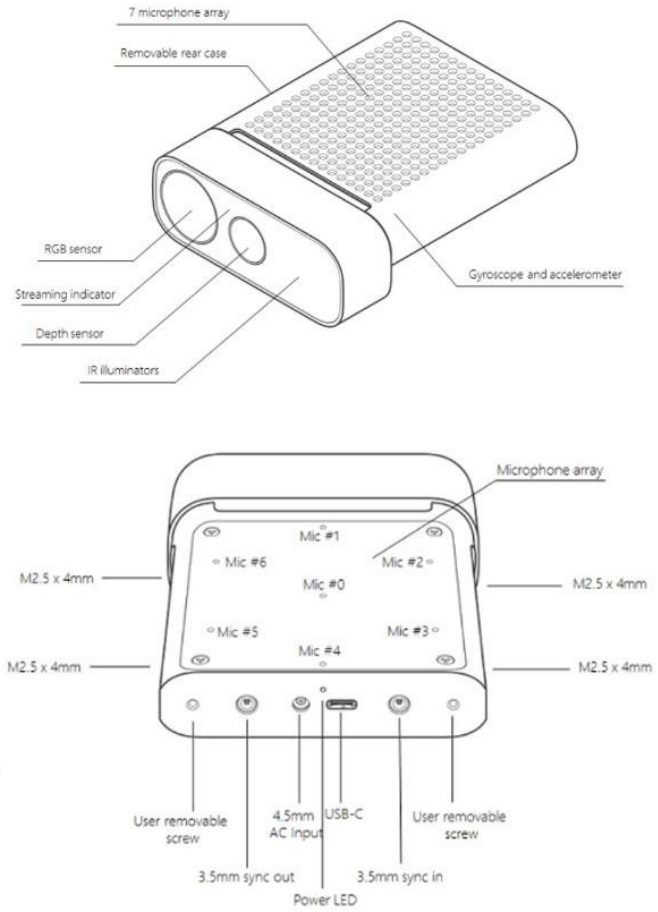


*Figura 5. Componenti Azure Kinect DK.*

Azure Kinect DK è destinato agli sviluppatori e alle attività commerciali piuttosto che ai consumatori e dovrebbe essere utilizzato a una temperatura ambiente compresa tra 10 e 25 gradi Celsius.

Azure Kinect DK è stato progettato per semplificare il lavoro degli sviluppatori con Azure, ma funziona con qualsiasi provider di servizi cloud o in modo autonomo.

Il nuovo dispositivo non dispone di elaborazione integrata e i requisiti di sistema sono PC Windows® 10 o Ubuntu 18.04 LTS con processore Intel® Core ™ i3 di settima generazione (Dual Core 2,4 GHz con GPU HD620 o più veloce), porta USB 3.0 e 4 GB di RAM. Non è disponibile su Windows 10 in modalità S. Il rilevamento del corpo e altre esperienze possono richiedere un hardware PC più

avanzato.

Con dimensioni quasi dimezzate rispetto a Kinect per Windows v2, Azure Kinect DK è stato progettato per riunire i migliori sensori di intelligenza artificiale in un singolo dispositivo e a differenza del predecessore non è stato progettato per essere usato con Xbox.

Di seguito, vengono illustrate e descritte altre alternative al kinect [13].

**VicoVR** è un sensore destinato alla realtà virtuale e ai giochi mobili. Utilizzandolo, si possono trasmettere i dati di profondità e corpo dal dispositivo allo smartphone Android o iOS e ciò rappresenta una grande opportunità per il mondo della telefonia.
Come svantaggio, VicoVR trasmette i dati tramite Bluetooth, rendendo lenta la trasmissione di alcuni scenari e al firmware manca il supporto per vari modelli Android.

- Prezzo: $ 399

*Figura 6. VicoVR.*

**Orbbec** viene fondata nel 2013 e fornisce 2 diversi tipi di sensori di profondità: ***Orbbec Astra*** *(Astra, AstraS, AstraPro, Astra+, AstraStereoS)* e ***Orbbec Persee*** [14].

*Figura 7. Orbbec.*

*Orbbec Astra* è un dispositivo che fornisce il rilevamento della profondità al computer connesso. La serie Astra è stata progettata per migliorare ulteriormente gli attributi che distinguono le telecamere

3D Orbbec dalle telecamere 3D esistenti sul mercato. Le telecamere Astra 3D forniscono una visione artificiale che abilita dozzine di funzioni come il riconoscimento facciale, il riconoscimento dei gesti, il tracciamento del corpo umano, la misurazione tridimensionale, la percezione dell'ambiente e la ricostruzione di mappe tridimensionali.

*Astra, Astra S e Astra Pro* offrono reattività di fascia alta, misurazione della profondità, gradienti uniformi e contorni precisi, nonché la capacità di filtrare pixel di profondità di bassa qualità. Le diverse versioni offrono agli sviluppatori la libertà di mettere a punto le proprie esigenze con opzioni di telecamere RGB a corto raggio, lungo raggio e ad alta risoluzione.
Astra + è l'ultima soluzione di percezione della profondità 3D di Orbbec: utilizzando la luce strutturata e l'elaborazione delle immagini, Astra + calcola un'immagine 3D dell'ambiente osservato in tempo reale. Fornisce una percezione 3D accurata anche in condizioni difficili, come misurazioni a lungo raggio.

*Astra Stereo S U3* combina tutti i vantaggi di Astra Stereo S ma con una USB 3.0 per fornire una fotocamera più potente ma con le stesse straordinarie capacità outdoor e multicamera. Molte applicazioni di profondità richiedono la necessità di più telecamere, soprattutto negli ambienti di vendita al dettaglio, ma si risolve questo problema consentendo di posizionare le telecamere 3D l'una vicino all'altra. Il dispositivo fornisce la visione artificiale che abilita funzioni per applicazioni ad alta precisione a distanza ravvicinata come vendita al dettaglio, automobili, picking robot, misurazione oggettiva e sicurezza domestica.

*Orbbec Persee* è un sensore autonomo con un sistema operativo integrato che include un SDK per il monitoraggio del corpo. È il primo computer con fotocamera al mondo. Questo dispositivo versatile e a basso costo può essere collegato al televisore / monitor oppure può funzionare senza display e puoi interagire interamente con esso tramite la videocamera Astra Pro 3D integrata. La sua capacità di vedere, ascoltare e capire lo rende molto più intelligente di un normale computer e il suo processore ARM integrato lo rende molto più utile dei dispositivi monouso.
Degno di nota per Orbbec, è la progettazione del Body Tracking SDK che consente ai computer di utilizzare i dati 3D delle proprie telecamere per analizzare e comprendere i corpi umani. Questo programma effettua un tracciamento scheletrico 3D veloce ed altamente accurato, è compatibile con ogni camera 3D Orbecc e riesce a rilevare fino a 5 persone nella scena. Tutto questo rende Orbecc il miglior sostituto del Kinect attualmente disponibile sul mercato.

**Zed** è una fotocamera 2K all'avanguardia. A differenza di Kinect e Orbbec, ZED non utilizza un sensore a infrarossi per misurare la profondità, imita il funzionamento dell'occhio umano: è dotato di 2 sensori ad altissima risoluzione che stimano la percezione della profondità.


*Figura 8. ZED.*

La fotocamera ZED non ha SDK per il tracciamento del corpo quindi bisogna considerare l'idea di acquistarlo se si utilizzano solo i flussi RGB e Depth del Kinect.

**OpenPose** non è, in realtà, un dispositivo in quanto utilizza qualsiasi semplice webcam per tracciare il corpo umano, il viso e le dita.

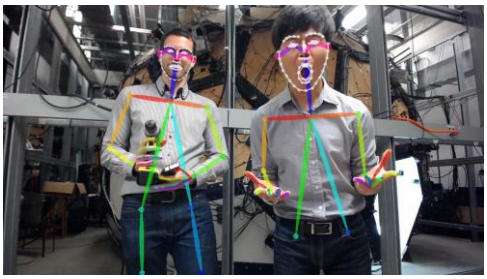

*Figura 9. Dimostrazione di un tracciamento tramite OpenPose.*

OpenPose è stato sviluppato dal Perception Computing Lab della Carnegie Mellon University ed è un software open source, ospitato su GiftHub .

# 3. Self Organizing Map e la sua versione estesa

Riconoscere i movimenti è la base del progetto AAL: monitorando la persona analizzata, possiamo garantire assistenza all'individuo, curandone ad esempio l'alimentazione.

Utilizzando immagini RGB-D per l'analisi, viene preservata la privacy del soggetto, in quanto l'immagine di profondità non ne rivela l'identità e non necessita che la persona indossi sensori, poiché l'elaborazione si effettua solo su immagini fornite dal sensore Kinect posto all'interno dell'abitazione.

Per avere un'analisi completa dei movimenti, l'immagine di profondità non è sufficiente, quindi viene applicato un tracciamento della parte alta del corpo, dove braccia e testa, parti del corpo che riguardano il food intake, sono bene in vista.

Si passa da una configurazione bidimensionale ad una tridimensionale: ogni frame viene considerato come una matrice a due dimensioni ma, avendo per ogni posizione un valore, la raffigurazione dell'individuo analizzato deve avvenire in uno spazio a tre dimensioni [23]. Questa raffigurazione viene chiamata **PointCloud**: un insieme di punti nello spazio definiti nella loro posizione secondo gli assi X, Y, Z in un determinato sistema di coordinate e caratterizzati da eventuali valori di intensità (RGB o altre lunghezze scalari.) ad essi associati.

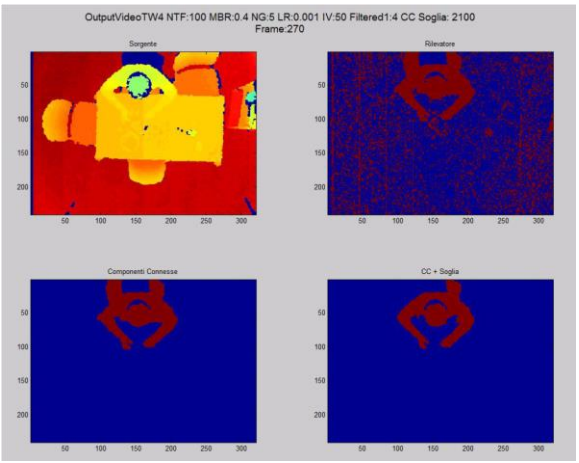


*Figura 10, Frame di profondità*

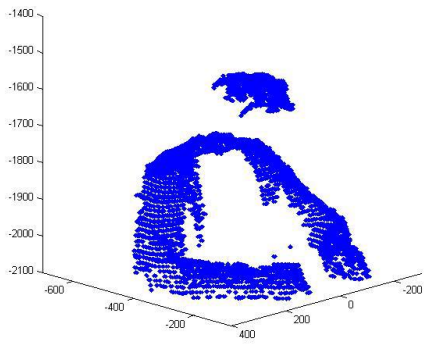


*Figura 11, PointCloud*

Il PC (figura 11) viene calcolato per ogni frame della sequenza (figura 10) e viene utilizzato come argomento per gli algoritmi di tracciamento quali Self-Organizing Map (SOM), Extended SOM (SOM Ex), reti neurali non supervisionate basate sui principi della self – organization: [24]

- ***Self-amplification***: la robustezza in un collegamento sinaptico aumenta se un neurone è attivato ripetutamente e costantemente da un altro neurone. Il risultato è un miglioramento tra due celle;
- ***Competizione***: a causa delle risorse limitate, ogni neurone è in stretta competizione con gli altri. Questo induce a una crescita della robustezza delle sinapsi tra neuroni che sono stati

attivati mentre le celle meno utilizzate tendono ad essere eliminate dai collegamenti della rete. La fase di competizione porta, solitamente, a quella di cooperazione;

- ***Cooperazione***: quando un neurone viene azionato esso trasmetterà anche una piccola modifica alla rete attorno alla cella attivata;

- ***Informazioni strutturali***: ci deve essere una correlazione tra gli input, non possono essere di origine completamente casuale, ma provenire dalle stesse fonti e in modo adeguato alla loro interpretazione. Ad esempio, non è possibile ricevere in ingresso dati sulla posizione della persona provenienti da due flussi di Kinect, devono essere elaborati e portati almeno sullo stesso piano per ricostruire la posizione di quest'ultima. Questo è un requisito fondamentale, se ciò non è verificato l'apprendimento non supervisionato non darà esito positivo, cedendo sul lato efficienza e pertinenza dei risultati ottenuti.

### 3.1 SOM

Il SOM [25] è un algoritmo che riceve in ingresso il modello iniziale caratterizzato da 17 elementi nello spazio 3D, corrispondenti alla posizione anatomica di una persona seduta su una sedia con le mani sul tavolo. Un solo nodo è utilizzato per identificare la testa e vi sono due nodi fondamentali che indicano i palmi delle mani. Di questi ultimi verrà valutata la distanza rispetto al nodo posto al centro della testa che andrà a designare una probabile azione di food intake. Considerata la posizione in top-view del Kinect sarà necessario separare il nodo della testa dalle altre parti anatomiche del modello, per impedire che i nodi della rete vengano attratti tra di loro e si uniscano erroneamente. L'algoritmo prevede che il joint della testa sia posizionato nel punto più alto del PC e gli altri siano ruotati in base alla direzione della persona rispetto al tavolo.
Il SOM ha lo scopo di adattare il modello al PC. In figura 12 vengono illustrati i vari step.
Siano:

- '**w**' il peso del vettore di riferimento che rappresenta la posizione in input relazionata ad una specifica configurazione dei neuroni, ad esempio la classica posizione che viene assunta da una persona che si appresta a mangiare un pasto;
- '$\xi$' Segnale di input o stimolo, è il segnale di ingresso all'algoritmo di self-organization.

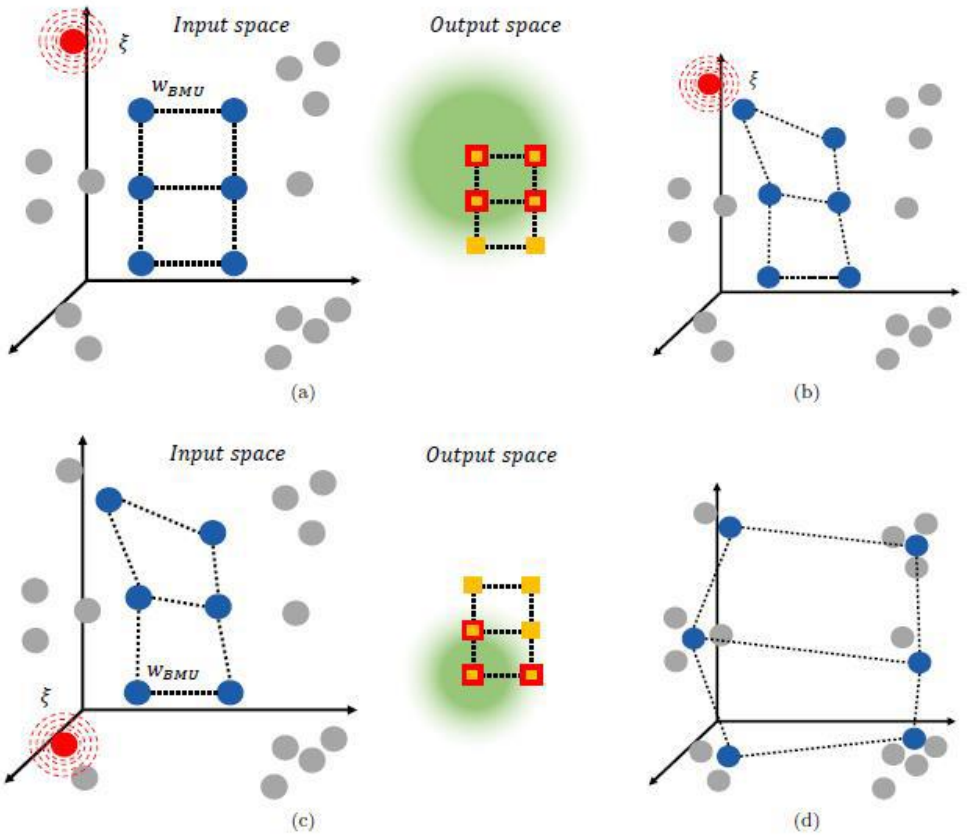


*Figura 12, A) Primo vettore in input all'algoritmo, identificazione del BMU, b) processo di adattamento, c) secondo vettore in input e funzione di adattamento per neuroni vicini, d) network finale.*

I passi dell'algoritmo SOM sono i seguenti:

1. Identificare il nodo che più si avvicina al modello passato in input all'algoritmo, calcolando la distanza Euclidea tra tutti i punti del PC e del modello;
2. Trovare il numero adatto di nodi adiacenti al Best Matching Unit (BMU) o nodo fondamentale, che dovranno essere adattati. Per semplicità computazionale si è scelto di prendere solo i diretti vicini e non considerare una più generica scelta con distribuzione Gaussiana.
3. Aggiornamento del modello in una direzione che approssimi meglio l'input iniziale, così da fornire un aggiornamento valido ed efficace.

Quando un nuovo PC è calcolato dal sistema, il modello in ingresso al SOM non sarà la prima rete assoluta: il sistema userà la rete aggiornata dal PC corrente come punto di partenza per il processo. In questo modo, i cambiamenti nella disposizione dei nodi sono molto contenuti, in quanto le differenze tra due PC consecutivi sono trascurabili.

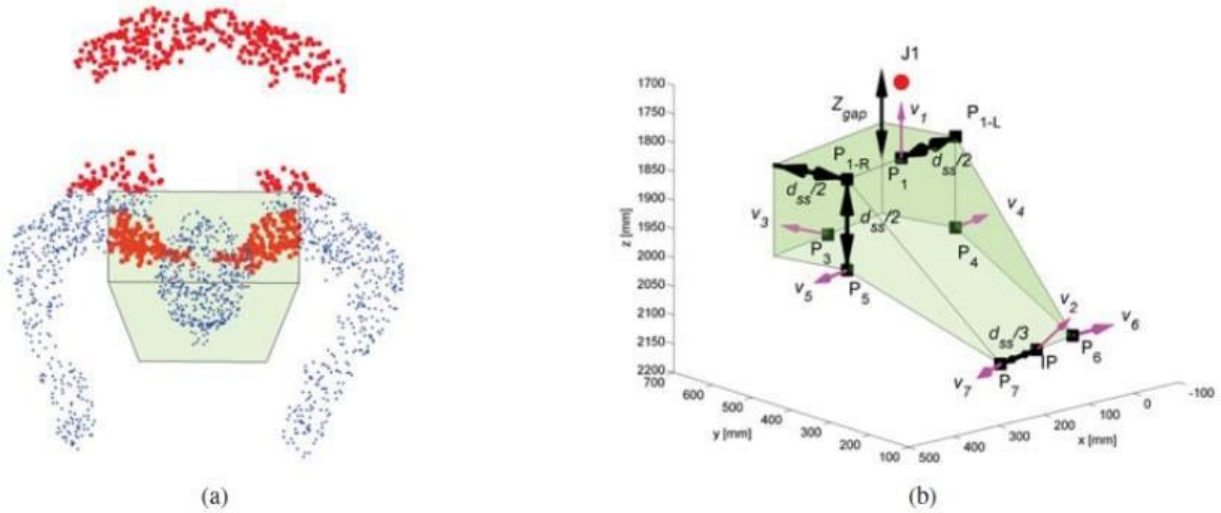


*Figura 13.Point Cloud 3D e Box del tronco, a sinistra i punti colorati in rosso sono quelli scartati*

Tuttavia, il SOM non è risultato efficiente, a causa dell'instabilità che manifesta in uno spazio tridimensionale come quello del PointCloud e per questo motivo la scelta si è spostata sul SOM Extended.

### 3.2 SOM Extended

Il **SOM Extended** è un'estensione del SOM dove il PC ottenuto dal frame di profondità è usato direttamente come input, senza nessun tipo di pre-processing.

Il modello è differente dal precedente, è composto da 50 nodi, per un totale di 47 piani, e 9 di questi sono usati per modellizzare la testa. Il numero corretto di nodi è stato trovato svolgendo alcuni test e risponde ad un tracciamento della persona sufficientemente accurato.

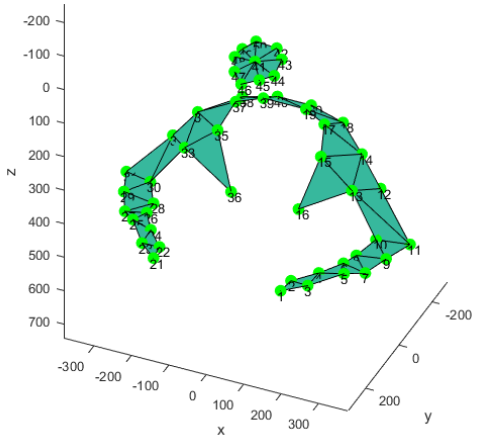


*Figura 14, modello SOM_EX*

Un confronto diretto con il SOM evidenzia un incremento di 3 volte del numero di nodi usati. Le catene di nodi prima usate nel SOM, ora sono unite tutte in un singolo gruppo. In questo modo non è più necessario effettuare un pre-processing in cui eliminare attraverso il box parte del PC, perché ora è considerato facente parte del modello stesso.

L'algoritmo SOM Extended produce in output il **fitted model** come risultato del tracciamento, a partire dal modello iniziale **skeleton model** e dal **PC** [26], come mostrato nella figura 15.
L'elaborazione delle prove con il SOM Ex hanno confermato la sua efficacia, ma per ottenere un buon risultato bisogna avere un PointCloud costruito correttamente e il più pulito possibile.

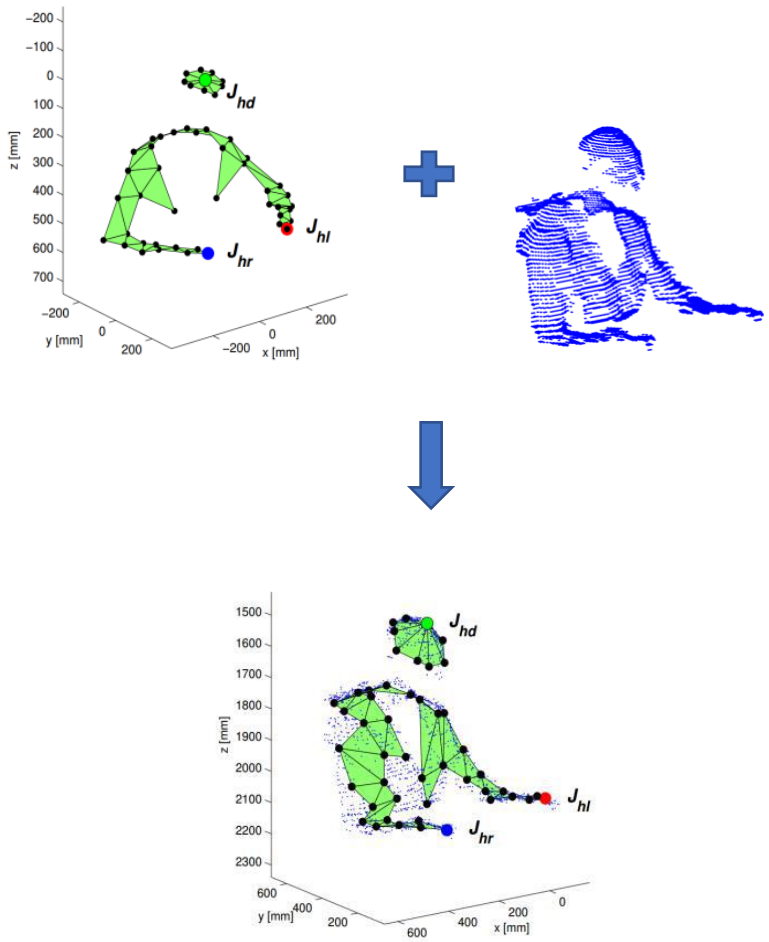


*Figura 15. A partire dal modello iniziale e dal PointCloud abbiamo in output il modello adattato*

Il mio lavoro di tesi sfrutta l'algoritmo SOM Ex per rilevare le azioni food intake all'interno dei vari test effettuati in modalità statica e dinamica, conteggiando il numero di azioni (figura 17) e facendo un confronto video tra i vari frame rate utilizzati (figura 16).

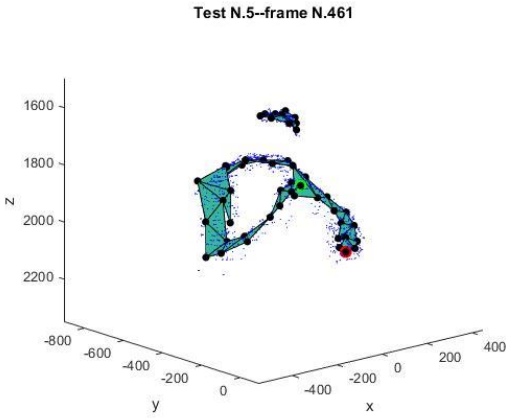


*Figura 16, SOM_EX video*

```
Test N.5--frame N.499--Save video? !!!YES!!!--Save Model? !!!YES!!!
Total Action Number N.3 Folder N.5
N.1 Action Duration:22
N.2 Action Duration:17
N.3 Action Duration:19
fx >>
```

*Figura 17, azioni rilevate dalla funzione ActionNumbers*

Sono stati processati 20 test di estrazione dinamica e 4 test di estrazione statica a 30 fps, 15fps, 6fps e 3fps, studiandone il comportamento ai vari frame rate, impostando una soglia (**OnesCons**) sulla durata minima per considerare che un'azione sia valida.

# 4. Estrazione statica e dinamica del foreground

Nel mio elaborato abbiamo preso in considerazione 30 acquisizioni fatte da dieci soggetti diversi, che riproducono l'azione food intake, con una durata che varia tra i 500 e i 3600 frame.
Per ogni ripresa dei test abbiamo posizionato il sensore kinect v1 sul soffitto a tre metri dal pavimento, in una modalità detta **top-view**, così da poter osservare dall'alto la persona nei suoi movimenti.

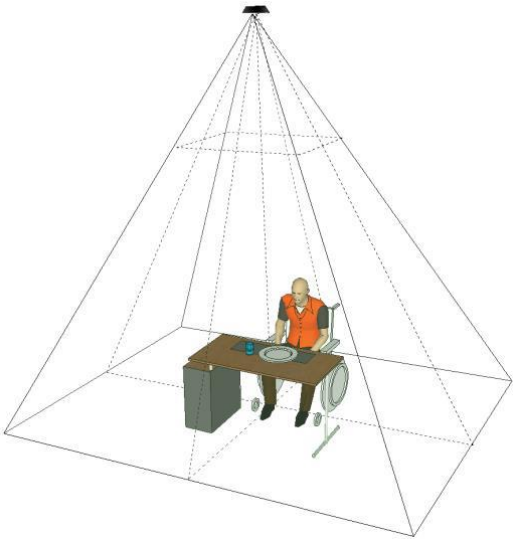

*Figura 18. Configurazione top-view.*

In questa ripresa la persona non viene osservata integralmente come nella modalità a vista frontale, in quanto la sua applicazione principale è quella di osservare un soggetto seduto che compie movimenti periodici e circoscritti: durante il pasto, a parte qualche piccola variazione, testa e spalle tendono a rimanere nella stessa posizione ed è la mano che porta acqua o cibo verso la bocca del soggetto analizzato.
In ogni test preso in analisi, possiamo fare una distinzione fra ***foreground*** e ***background*** all'interno di una sequenza di frame. Con foreground o *primo piano*, si indicano tutti quegli elementi di interesse all'interno di un'immagine; con background o *sfondo*, rappresentiamo invece tutti quegli elementi che non lo sono.
Ci sono tre diversi approcci per la rilevazione del foreground [15]: per pixel, per regioni o per frame. Gli ultimi due sono considerati dispendiosi, più robusti (per regioni) o utili da supporto (per frame) e quindi l'approccio per pixel viene considerata la scelta migliore, essendo più preciso ed economico dal punto di vista computazionale.
La successione di frame all'interno di un'acquisizione la possiamo ispezionare pixel per pixel. Ogni successione di pixel rappresenta una distribuzione di valori indipendenti dagli altri pixel e possiamo rappresentarle tramite un insieme di gaussiane [16]. Ogni distribuzione avrà un certo andamento, con un picco di valori (monomodale) o con più picchi (multimodale), ognuno dei quali può essere

approssimato tramite una Gaussiana definita da:

$$\eta(X_t; \mu; \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t-\mu_t)^T \Sigma^{-1}(X_t-\mu_t)}$$

dove n è la dimensione con cui si sta lavorando (tridimensionale nel caso si lavori con pixel che assumono valori secondo i canali RGB)

Σ è la matrice di covarianza del tipo: $\Sigma_{k,t} = \sigma_k^2 \boldsymbol{I}$, dove $\boldsymbol{I}$ è la matrice identità (assumiamo che i valori RGB siano indipendenti tra loro e che abbiano la stessa varianza)

Consideriamo un pixel come riscontrato rispetto alla k-esima componente se:

$$\left| X_t^{(i)} - \mu_{k,t} \right| \leq \beta \sigma_{k,t}$$

dove $\beta = 2.5$ tipicamente.

Esposta la modellazione dei pixel, si deve capire se il pixel sta rappresentando un oggetto di primo piano o di sfondo. Le componenti gaussiane vengono ordinate secondo il valore $\omega/\sigma$ in maniera decrescente. Infatti, una componente di sfondo è caratterizzata dall'avere o pixel sempre simili ($\sigma$ piccolo) o dall'avere numerosi riscontri ($\omega$ grande).

Le gaussiane per il modello di BG vengono scelte con la seguente relazione:

$$Numero\ Componenti\ BG = argmin_b(\sum_{k=1}^{b} \omega_k > T)$$

$T$ è una misura della minima porzione di dati da considerare come sfondo. È un valore arbitrario con le seguenti proprietà:

- $T$ piccolo → sfondo monomodale, adatto per sfondi statici
- $T$ grande → sfondo multimodale, più adatto per sfondi in movimento ripetitivo.

Il pixel assumerà quindi la classificazione di Primo Piano se rispetta la seguente condizione:

$$\sum_{k=1}^{matched} \omega_{k,t} > T$$

Altrimenti sarà classificato come sfondo.

Ci sono diverse tecniche di sottrazione del background (*filtro medio temporale, esecuzione della media gaussiana, stima della densità del kernel, approssimazione KD sequenziale*), ma si è scelto di usare la miscela delle gaussiane in quanto

1. risulta un algoritmo versatile in diverse situazioni, applicando i giusti parametri.
2. E'presente un'ampia documentazione a riguardo.
3. L'algoritmo è in grado di "imparare": le variazioni in un'immagine vengono gestite correttamente.

4. L’uso delle distribuzioni gaussiane le rende piuttosto semplici da utilizzare.

Le acquisizioni sono state realizzate con due modalità: ***sottrazione dinamica e statica del background***.
Nel primo i soggetti partono in piedi e finisco per sedersi al tavolo dove muovono la mano destra portandola alla bocca per simulare l’azione del mangiare o del bere;
nel secondo la scena inizia direttamente con l’individuo già seduto e pronto ad iniziare l’azione.
La condotta che la persona ha quando è seduta, è la stessa nelle acquisizioni fatte con background statico che con quello dinamico, ciò che cambia è la parte iniziale dell’acquisizione, lo sfondo.

Nella sottrazione dinamica del background si estrae il soggetto che frame per frame è in movimento rispetto al background: abbiamo una sequenza di immagini di sfondo in “movimento” dove il soggetto raggiunge il tavolo.

Nella sottrazione statica del background lo sfondo viene catturato in assenza del soggetto seduto a tavola: è costituito da pavimento, tavolo e sedia, per cui la sottrazione frame per frame fa emergere il foreground.

In generale la sottrazione dello sfondo statica non è applicabile in ambienti reali. Con le acquisizioni in interno, i riflessi o le immagini animate sugli schermi portano a cambiamenti di sfondo. Allo stesso modo anche all’esterno presenta problematiche: cambiamenti climatici come pioggia, vento o cambiamenti di illuminazione causati dal tempo, rendono questa modalità non praticabile. Nel nostro caso è stato possibile fare il confronto con le due estrazioni in quanto le acquisizioni sono state fatte in laboratorio, cercando il più possibile di mantenere l’ambiente privo di interferenze.

Andiamo ora ad illustrare nello specifico le due tipologie di estrazione del foreground (sottrazione del background).

### 4.1 Sottrazione dinamica del background

Matlab, tra i numerosi strumenti che mette a disposizione per vari campi di studio, fornisce il pacchetto Vision. **Vision** è un toolbox con algoritmi, funzioni e applicazioni per la progettazione e la simulazione di sistemi di computer vision e di elaborazioni video, fra cui la rilevazione del foreground [17].

Per estrarre dinamicamente il primo piano, si utilizza la classe ***ForegroundDector***, [18] presente nel pacchetto vision, che confronta un fotogramma video a colori o in scala di grigi per determinare se i

singoli pixel fanno parte dello sfondo o del primo piano, restituendo una "maschera" di primo piano utilizzando i modelli di miscela gaussiana. Prima di procedere all'utilizzo, il modello di riferimento ha bisogno dell'inizializzazione di alcuni parametri:

- **NumTrainingframe:** numero intero di frame video per l'apprendimento del modello del background. Più è grande questo valore, più frame concediamo al modello per configurare lo sfondo e la migliore sarà la sua modellizzazione.
- **LearningRate:** numero scalare che indica la velocità con cui i parametri delle componenti della miscela di gaussiane vengono modificati; controlla la rapidità con cui il modello si adatta alle condizioni che mutano nel tempo. Occorre impostare questo parametro in modo appropriato per garantire la stabilità dell'algoritmo.
- **MinimumBackgroundRatio:** definito da un numero scalare, rappresenta la soglia per determinare il modello del background, cioè la probabilità a priori minima per pixel che devono essere considerati come sfondo. Nella teoria viene definito come T.
- **NumGaussians:** numero di mode di gaussiane nel modello della mistura (valore tra i 3 e i 5)
- **InitialVariance:** indica la varianza con cui i componenti della miscela vengono inizializzati. E' necessario al sistema poiché all'inizio il modello è vuoto: non avendo ancora letto alcun frame, non ci sono dati su cui creare le gaussiane, per cui la varianza viene specificata dall'utente nel costruttore di ForegroundDector. Più grande è questo valore, maggiore sarà la probabilità che i primi valori dei pixel diano una corrispondenza e questo aumenta la velocità con cui i pixel iniziali sono catalogati come sfondo. Deve essere inizializzato tenendo conto dei valori assunti da NumTrainingframe.

Dopo aver inizializzato media, varianza, il numero di gaussiane da considerare e la soglia minima della porzione di sfondo, si possono processare i nuovi frame.

Nell'estrazione dinamica, sull'immagine di foreground agiranno due filtri, **gaussiano** e **erosione + dilatazione** con le relative operazioni di *opening, closing* e *gradiente morfologico*.

Applicando un filtro è possibile migliorare la qualità dell'immagine di primo piano estratta da una sequenza di frame, eliminando il rumore o estraendo particolari caratteristiche del segnale [19]. L'implementazione di operazioni per modificare o trasformare un'immagine (definite nella loro forma digitali come matrici) è realizzata mediante una seconda matrice, di dimensioni inferiori, chiamata *kernel.* Questa matrice viene fatta scorrere sopra l'immagine sorgente e, in base alla trasformazione richiesta, gli elementi sovrapposti subiranno una modifica. Lo scorrimento avviene

considerando come cursore un elemento del kernel, definito *anchor point.* Il valore della sorgente che verrà modificato sarà solo quello al di sotto dell'anchor point.

Parlando di ***erosione*** **+** ***dilatazione***, l'effetto del kernel è quello di minimo e massimo locale. Il minimo/massimo valore a cui il kernel si sovrappone viene sostituito al valore del pixel al di sotto del punto di ancoraggio. Il risultato dell'operazione di dilatazione (erosione) è che le regioni con valori alti crescono (diminuiscono).

La ***dilatazione*** ha come effetto l'allargamento dei contorni di un oggetto di foreground: si avrà una crescita dell'area del soggetto e una riduzione dei "vuoti". Attenua le concavità e serve a fondere insieme due blob con caratteristiche di intensità simili, che possono essere stati separati da alcuni disturbi sovrapposti all'immagine originale [20].

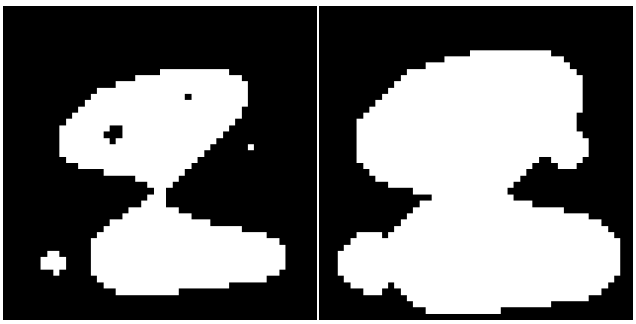

*Figura 19. Dilatazione*

L' **erosione** manifesta la concentrazione dei contorni di un oggetto di foreground: l'area del soggetto sarà ridotta, le sporgenze saranno rimosse e si avrà un allargamento dei "buchi" dell'oggetto (elementi con valori bassi).

*Figura 20, Erosione*

L' ***apertura*** e la ***chiusura*** sono due operazioni aggiuntive ottenute dalla combinazione di erosione e dilatazione: la trasformazione di opening permette la rimozione di piccoli oggetti, mentre quella di closing permette di riempire piccoli buchi. In particolare, a partire dall'immagine originale viene eseguita l'erosione prima della dilatazione, il risultato è l'apertura. Per la chiusura l'ordine è invertito.

Il risultato per l'apertura è simile all'erosione (l'erosione invece è simile alla dilatazione), ma l'area dei blob è meno alterata.

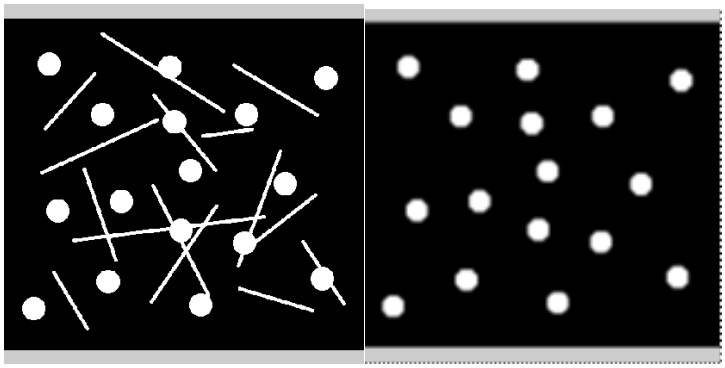

*Figura 21. Operazione di apertura.*

L'apertura seleziona le forme di interesse.

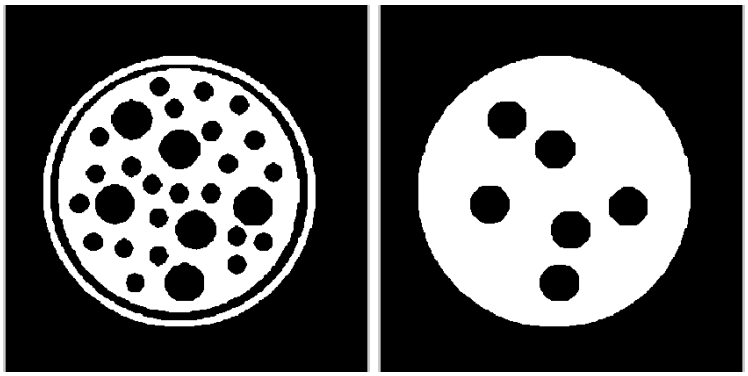

*Figura 22. Operazione di chiusura*

L'effetto della chiusura è speculare a quello dell'apertura: le cavità dell'immagine sono riempite in accordo all'elemento strutturante scelto.
Il filtro E+D utilizza un kernel quadrato, definito dalla funzione **strel**.

Un oggetto strel rappresenta un elemento strutturante morfologico piano, che è parte essenziale delle operazioni di dilatazione ed erosione. È costituito da un insieme di valori binari vicini, 2-D o multidimensionali, in cui i pixel veri sono inclusi nel calcolo morfologico, mentre i falsi pixel non lo sono. Il pixel centrale dell'elemento strutturante, chiamato origine, identifica il pixel nell'immagine in fase di elaborazione. Utilizzare la funzione strel serve per creare elementi strutturanti piani ed è possibile utilizzarli con immagini binarie e in scala di grigi [21].

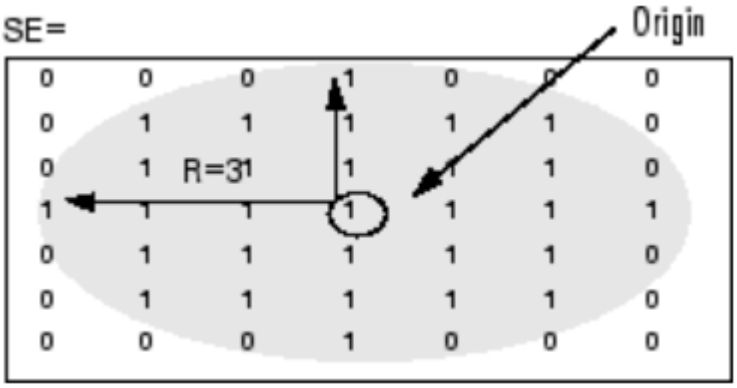


*Figura 23. Elemento strutturante piano*

In pratica si parte dal centro dell'immagine e nella dilatazione il valore massimo all'interno del kernel sostituisce il pixel nella posizione di ancoraggio (punto centrale dell'immagine binaria), mentre per l'erosione si assume il valore minimo.

I test di estrazione dinamica sono stati realizzati in laboratorio con il kinect v1 e processati tramite il tool **TST INTAKE MONITORING V1.**

Nel v_1 per estrarre dinamicamente il foreground, si ha bisogno di operazione matematiche di filtraggio (filtro gaussiano e E+D con le relative operazioni di opening e closing illustrate precedentemente). Si estrae il soggetto che frame per frame è in movimento rispetto al background. (Figura 24).

Il tavolo, parte del background, ha misurazioni predefinite, quindi un'acquisizione fatta senza rispettare i parametri non permette di far girare i test.
Per ottenere il point cloud del soggetto analizzato, ossia l'argomento di input per l'algoritmo SOM Ex, bisogna convertire i frame di profondità ottenuti dall'estrazione, in uno spazio a tre dimensioni: si passa da una configurazione bidimensionale ad una tridimensionale, in quanto ogni punto dello spazio considerato viene definito nella sua posizione, secondo gli assi X,Y e Z in un determinato sistema di coordinate e caratterizzati da eventuali valori di intensità ad essi associati come RGB o altre lunghezze scalari. Ottenuto il point cloud della persona, l'algoritmo SOM Ex fornisce in output il modello adattato. L'inizio e la fine di un pasto, denominati Start Frame e End Frame, vengono settati manualmente e permettono alla funzione ActionNumebers di calcolare il numero di azioni effettuate e la loro durata.
Ogni funzione presenta dei parametri che devono essere settati per una corretta acquisizione a seconda del tipo di acquisizione (frontale o in top-view), della versione del kinect utilizzata (v1 o v2) e dei test presi in analisi.

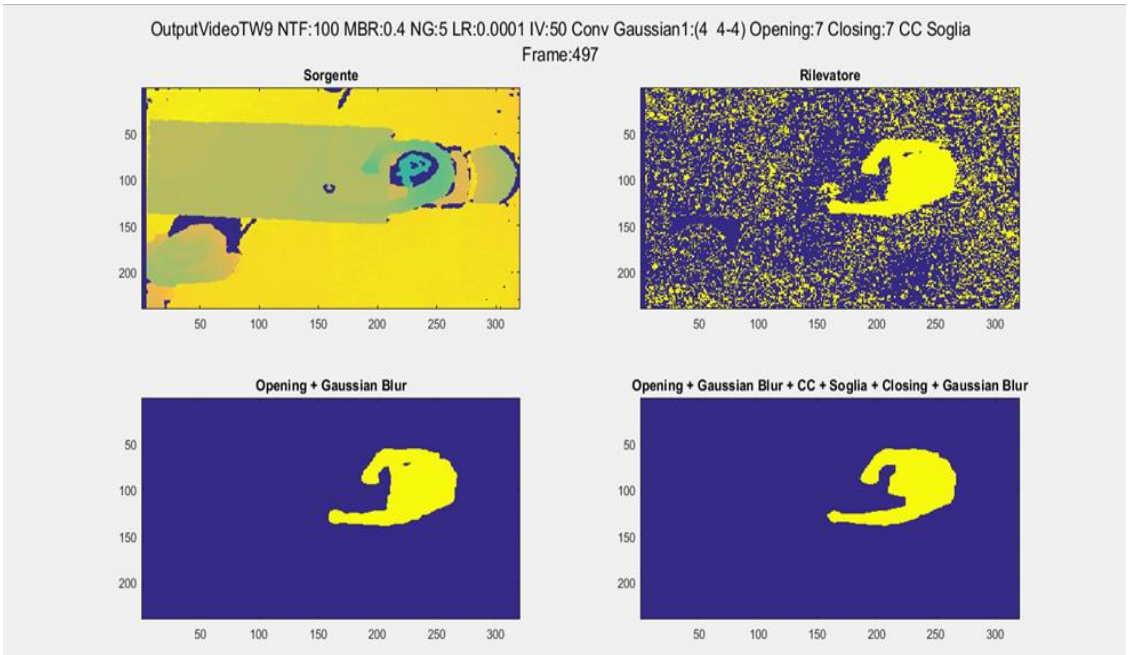


*Figura 24: immagine di primo piano estratta e processata tramite i filtri.*

### 4.2 Sottrazione statica del background

Nell'estrazione statica, a differenza della dinamica si ha un background predefinito. Si effettua una semplice sottrazione dello sfondo acquisito in una sola immagine rispetto al soggetto in movimento nei frame successivi [22]. Attraverso le immagini delle figure 25, 26 e 27 illustriamo il processo di estrazione statica.

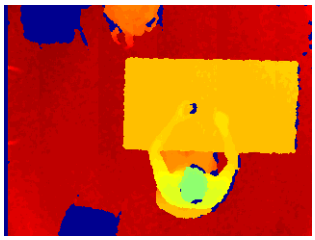

*Figura 25. Dati grezzi acquisiti*

*Figura 26. Operazione di riempimento*

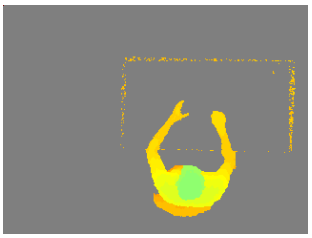

*Figura 27. Estrazione di foreground*

L'output è visibile nella figura 27, dove i pixel di primo piano hanno un colore diverso dal grigio. Nel foreground viene applicato un filtro mediano con una finestra di 5x5 pixel per rimuovere gli elementi di rumore. Nell'immagine il rumore è costituito dai pixel presenti al bordo del tavolo: qui le informazioni sulla distanza hanno una maggiore variabilità rispetto ad altre aree simili come il centro del tavolo o del pavimento e per questo motivo questi pixel possono essere erroneamente

contrassegnati come primo piano.
L'algoritmo poi troverà il blob più grande nell'area monitorata, assumendo che coincida con il blob della persona. Finalmente, i parametri intrinsechi della telecamera di profondità vengono sfruttati per calcolare il Point Cloud della persona.
Ogni punto del PC verrà selezionato casualmente come valore di input per gli algoritmi di SOM e SOM Ex che verranno descritti nel prossimo capitolo.

Per acquisire i dati necessari all'estrazione statica del foreground, abbiamo utilizzato il tool 'SkeletalViewer.exe" che permette di esportare i seguenti stream di dati:

- ✓ RGB: Ciascun frame è memorizzato in un file .bmp (640x480) e presenta un nome del tipo 'FrameRGB_0.bmp', 'FrameRGB_1.bmp', …
- ✓ Depth: Ciascun frame è memorizzato in un file .bin e presenta un nome del tipo 'Filedepth_0.bin', 'Filedepth_1.bin'… E' possibile esportare sia frame 320x240 che 640x480
- ✓ Skeleton: Tutti i frame dello scheletro vengono esportati in 4 file, ma in questo lavoro non è stato utilizzato.

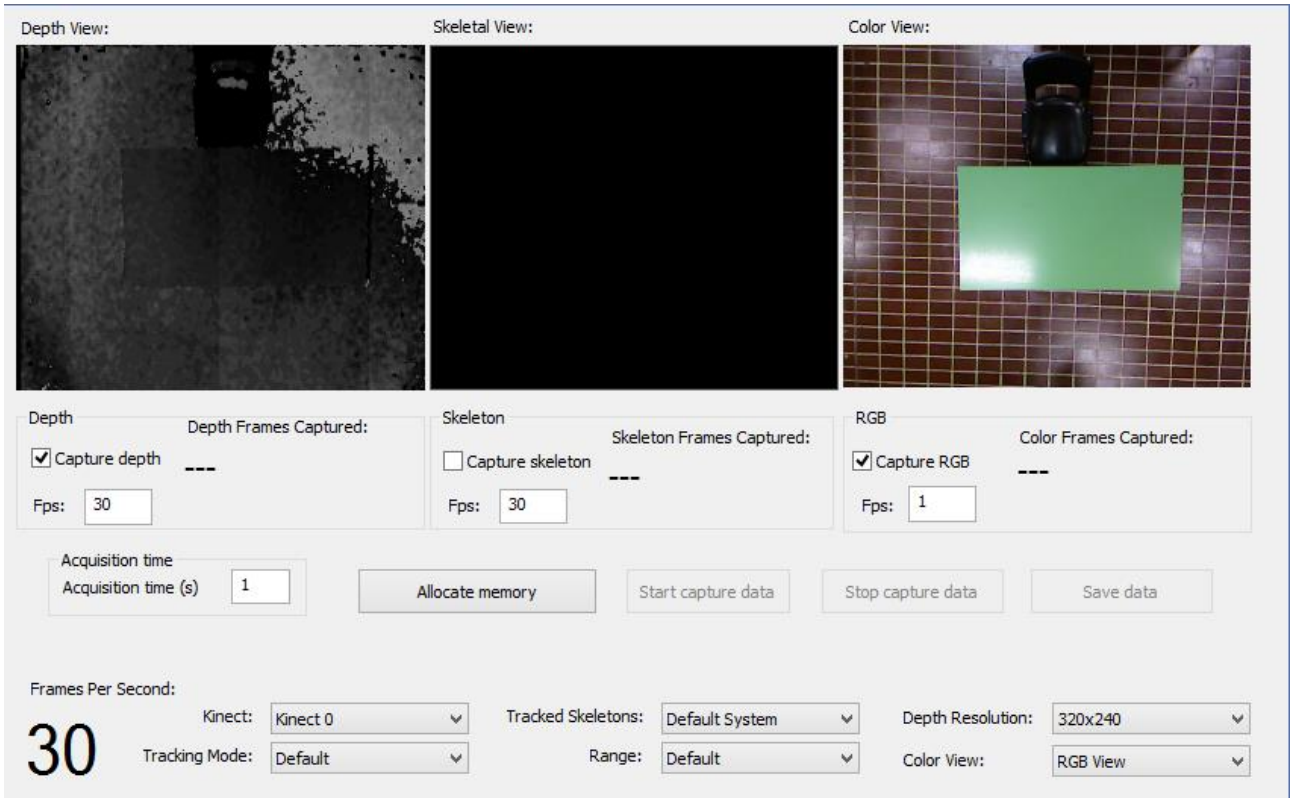


*Figura 28, Interfaccia*

Per prelevare i flussi di immagini di profondità e colore proveniente da kinect v1, è necessario:

1. Selezionare uno o più stream da esportare: vengono mostrati i 3 stream acquisiti dal sensore, partendo da sinistra si ha Depth, Skeleton, RGB.
2. Scegliere il frame rate a cui acquisire gli stream: di default si ha 30fps per Depth e Skeleton, 1 fps per RGB
3. Scegliere il numero di secondi da catturare. Se il numero di secondi supera quello consentito dalla RAM disponibile, il tempo massimo viene ridotto tenendo conto di questo
4. Il tempo massimo è limitato a 900 secondi (15 minuti)
5. Cliccare sul pulsante 'Allocate memory' per consentire al software di inizializzare le strutture in cui appoggiare i dati in RAM.
6. Cliccare sul pulsante 'Start capture' per avviare l'acquisizione degli stream selezionati.
7. È possibile stoppare l'acquisizione attraverso 'Stop capture'.
8. Se l'acquisizione non viene stoppata e si raggiunge il numero massimo di frame stabilito, il software si stoppa automaticamente.
9. Cliccare sul pulsante 'Save data' per esportare i dati e attendere che venga visualizzata la scritta 'Saved'.

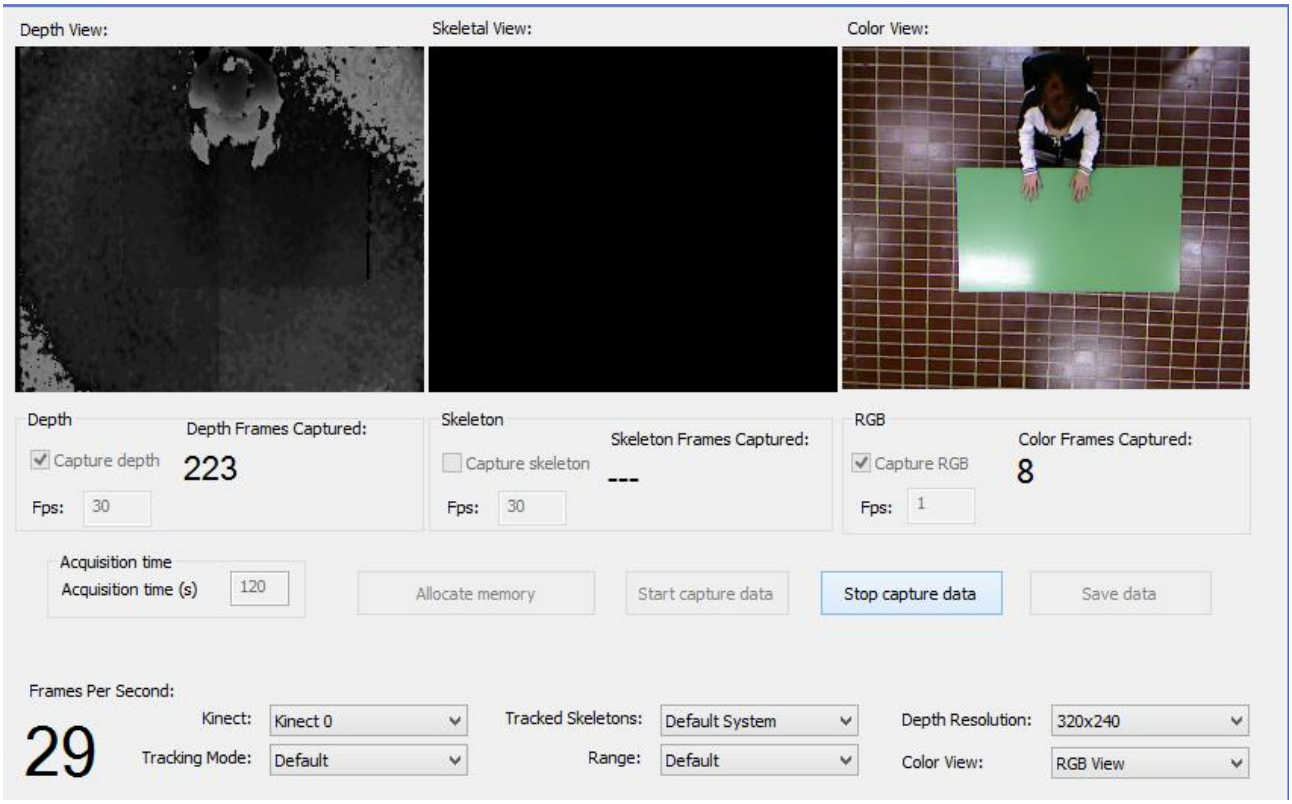


*Figura 29, Acquisizione*

Una volta salvati i dati, occorre elaborare i frame di profondità ricevuti dal sensore. Lo scopo di questa operazione è quello di estrarre il Point Cloud (PC) associato al blob (area contenente i pixel relativi alla sagoma del soggetto) nel dominio della profondità.

Alcuni punti del PC iniziale costituiscono il modello di input per l'elaborazione successiva con gli algoritmi SOM e/o SOM Ex, nello specifico 17 punti per il SOM e 50 per il SOM Extended costituiscono i modelli di input.
Queste acquisizioni sono state processate tramite il tool **TST INTAKE MONITORING V3.**

Il tool v_3, a differenza del precedente, è caratterizzato da una funzione principale, MainFunction_SF, in cui vengono richiamate tutte le altre funzioni utili per il processing. Aprendo il Main è possibile scegliere i parametri e modificarli, decidere quali algoritmi neurali utilizzare e in ultima istanza eseguire *ActionNumbers*, con il compito di conteggiare il numero delle azioni dopo aver elaborato i frame con il SOM o SOM Ex.
Fondamentali sono i parametri per il ***riconoscimento della persona***:

- **thBlob**: dimensione oltre al quale un blob viene considerato come appartenente ad una persona.
- **Distanza**: distanza dal tavolo, della stessa quantità sia lungo l'asse x e y della griglia dei pixel. Identifica l'area entro la quale una persona si può considerare vicina al tavolo (figura 30).

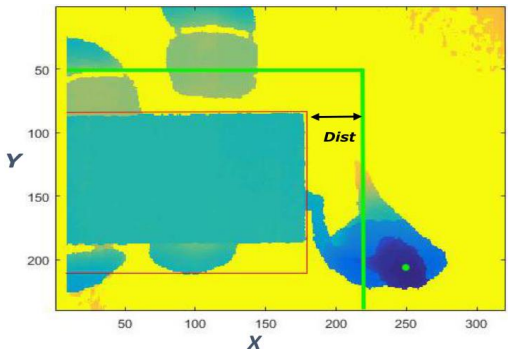


*Figura 30. Indica quando una persona viene considerata vicino al tavolo*

- **Rapporto**: indica di quanto l'altezza del soggetto in piedi si deve ridurre per poterlo considerare seduto

- **Hand**: dimensione oltre la quale un blob viene considerato come una mano e/o avambraccio
- **Hand_gap**: altezza in mm dal piano del tavolo dentro al quale osservare i blob delle mani

E i parametri del sensore utilizzato, Kinect v1:

- TableHeight = 2100
- SensorHeight = 3000
- rowPixel = 240
- columnPixel = 320

Il tool v_3 viene utilizzato per estrazioni statiche e estrazioni dinamiche: nel nostro caso abbiamo lo abbiamo utilizzato per delle acquisizioni effettuate in modalità statica, quindi il soggetto parte direttamente da seduto.

Riconosciuta la persona seduta con le mani appoggiate al tavolo, il PC viene estratto sottraendo il blob della persona al frame di profondità dello sfondo (figura 31), costituito da tavolo e sedia. Il tavolo, facente parte del background, non ha misurazioni predefinite come accade utilizzando il tool v_1. Per ogni nuova acquisizione la funzione Table calcola altezza e misure del tavolo in corrispondenza della persona.

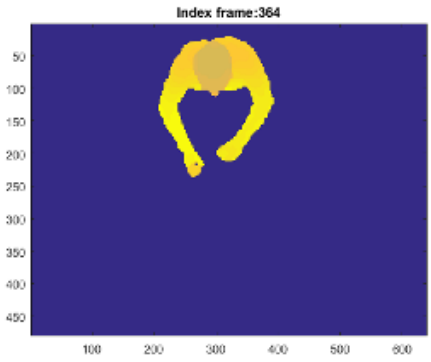


*Figura 31, Il PC viene estratto sottraendo il blob della persona dal frame di profondità dello sfondo*

A questo punto viene definito automaticamente lo Start Frame e l'algoritmo SOM Ex inizia l'elaborazione per il tracciamento del modello, adattando il modello iniziale. Tramite la funzione ActionNumbers vengono rilevate il numero di azioni svolte durante il pasto, che vengono salvate in una cartella dove vengono raccolti tutti i risultati ottenuti e viene impostato l'EndFrame.

# 5. Risultati

In seguito, vengono illustrate tramite delle tabelle, i risultati ottenuti analizzando le varie acquisizioni a diversi FrameRate, impostando per ogni tabella un valore per la variabile OnesCons.

### 5.1 Test di estrazione dinamica

I test effettuati sono stati eseguiti in laboratorio, processati con estrazione dinamica e sono tutti composti da 500 frame. Di seguito, grazie alla tabella, possiamo vedere come cambiano il numero di azioni con una diminuzione del FrameRate e con un valore più basso della variabile OnesCons. Diminuendo la soglia degli uni, in alcuni test le azioni aumentano, in quanto vengono considerate anche le azioni di breve durata, che in un confronto a video non vengono notate.

| OnesCons=9 | ActionNumbers FrameRate=30 | Azioni FrameRate=15 | Azioni FrameRate=6 | Azioni FrameRate=3 |
|---|---|---|---|---|
| Test_1 | 3 | 2 | 2 | 2* |
| Test_2 | 4 | 4 | 3* | 3* |
| Test_3 | 4 | 3 | 2 | 2* |
| Test_4 | 1 | 1 | 1* | 2* |
| Test_5 | 3 | 3 | 3* | 3* |
| Test_6 | 1 | 3 | 2 | 2* |
| Test_7 | 1 | 1 | 1 | 2* |
| Test_9 | 3 | 3 | 3 | 2* |
| Test_10 | 4 | 4 | 4 | 2* |
| Test_11 | 2 | 2 | 2 | 1* |
| Test_12 | 1 | 1 | 1 | 1* |
| Test_13 | 1 | 1 | 1 | 1* |
| Test_14 | 5 | 3 | 3 | 2* |
| Test_15 | 1 | 2 | 2 | 1* |
| Test_16 | 2 | 2 | 2 | 2* |
| Test_17 | 1 | 1 | 1 | 1* |
| Test_18 | 1 | 1 | 1 | 1* |
| Test_19 | 3 | 3 | 3 | 2* |
| Test_20 | 2 | 2 | 2 | 1* |

| OnesCons=6 | ActioNumbers FrameRate=30 | Azioni video FrameRate=15 | Azioni video FrameRate=6 | Azioni video FrameRate=3 |
|---|---|---|---|---|
| Test_1 | 3 | 2* | 2* | 2* |
| Test_2 | 4 | 4 | 3* | 3* |

| | | | | |
|---|---|---|---|---|
| **Test_3** | **5** | **3*** | **3** | **3*** |
| **Test_4** | **1** | **1*** | **1*** | **1*** |
| **Test_5** | **3** | **3** | **3** | **3*** |
| **Test_6** | **1** | **3** | **2** | **2*** |
| **Test_7** | **1** | **2*** | **2*** | **2*** |
| **Test_9** | **3** | **3** | **3** | **3*** |
| **Test_10** | **4** | **4** | **4** | **4*** |
| **Test_11** | **3** | **3** | **2** | **2*** |
| **Test_12** | **1** | **1** | **1** | **1*** |
| **Test_13** | **1** | **1** | **1** | **1*** |
| **Test_14** | **5** | **3** | **3** | **3*** |
| **Test_15** | **1** | **2** | **2** | **2*** |
| **Test_16** | **2** | **2** | **2** | **2*** |
| **Test_17** | **1** | **1** | **1** | **1*** |
| **Test_18** | **1** | **1** | **1** | **1*** |
| **Test_19** | **3** | **3** | **3** | **3*** |
| **Test_20** | **2** | **2** | **2** | **2*** |

| **OnesCons=3** | **ActioNumbers FrameRate=30** | **Azioni video FrameRate=15** | **Azioni video FrameRate=6** | **Azioni video FrameRate=3** |
|---|---|---|---|---|
| **Test_1** | **5** | **2*** | **2*** | **2*** |
| **Test_2** | **4** | **4*** | **4*** | **4*** |
| **Test_3** | **4** | **3*** | **3*** | **3*** |
| **Test_4** | **4** | **4** | **4*** | **4*** |
| **Test_5** | **3** | **3** | **3*** | **3*** |
| **Test_6** | **1** | **3** | **3*** | **3*** |
| **Test_7** | **1** | **1*** | **1*** | **1*** |
| **Test_9** | **3** | **3** | **3*** | **3*** |
| **Test_10** | **4** | **4** | **4*** | **4*** |
| **Test_11** | **4** | **3** | **3*** | **3*** |
| **Test_12** | **1** | **1** | **1*** | **1*** |
| **Test_13** | **1** | **1** | **1*** | **1*** |
| **Test_14** | **6** | **3** | **3*** | **3*** |
| **Test_15** | **1** | **2** | **2*** | **2*** |
| **Test_16** | **2** | **2** | **2*** | **2*** |
| **Test_17** | **1** | **1** | **1*** | **1*** |
| **Test_18** | **1** | **1** | **1*** | **1*** |
| **Test_19** | **3** | **3** | **3*** | **3*** |
| **Test_20** | **4** | **2*** | **2*** | **2*** |

Di seguito vengono illustrati lo StartFrame e l'EndFrame per ogni test:

| | TEST 1 | TEST 2 | TEST 3 | TEST 4 | TEST 5 | TEST 6 | TEST 7 | TEST 8 | TEST 9 | TEST 10 | TEST 11 | TEST 12 | TEST 13 | TEST 14 | TEST 15 | TEST 16 | TEST 17 | TEST 18 | TEST 19 | TEST 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SF | 215 | 170 | 200 | 179 | 250 | 204 | 148 | 390 | 112 | 140 | 221 | 155 | 122 | 140 | 135 | 130 | 190 | 160 | 100 | 130 |
| EF | 500 | 500 | 500 | 500 | 500 | 500 | 500 | 500 | 500 | 500 | 500 | 335 | 264 | 500 | 500 | 462 | 420 | 380 | 500 | 450 |

Con un FrameRate =3, indipendentemente dai valori della variabile OnesCons impostati (3,6,9),

occorre settare il parametro OnesCons a 12 in quanto il "grande salto" tra un frame e l'altro preclude la veridicità del numero di azioni riportate da ActionNumbers: i vari test presentano azioni da 4/9 frame ripetute, a seconda del settaggio degli uni, che non vengono notate a video (azioni contrassegnate con *).

Abbassando gli uni significativi a 3, il problema si presenta anche con un FrameRate=6.

Come possiamo vedere nel confronto video tra i vari FrameRate, aumentando il salto di frame, viene alterata la durata delle azioni e il numero delle stesse: in alcuni test il numero di azioni è differente anche a partire da una diminuzione non eccessiva (FrameRate=15) come nei Test 6 e 14.

Oltre a questi 20 test, resi disponibili nel sito del corso www.tlc.dii.univpm.it, abbiamo effettuato altre 4 acquisizioni in dinamico, tutte dell'ordine di 3000 frame. Purtroppo, queste acquisizioni sono risultate molto rumorose (Figura 32) e presentavano problemi sull'analisi di molti frame, che compromettevano l'estrazione del foreground.

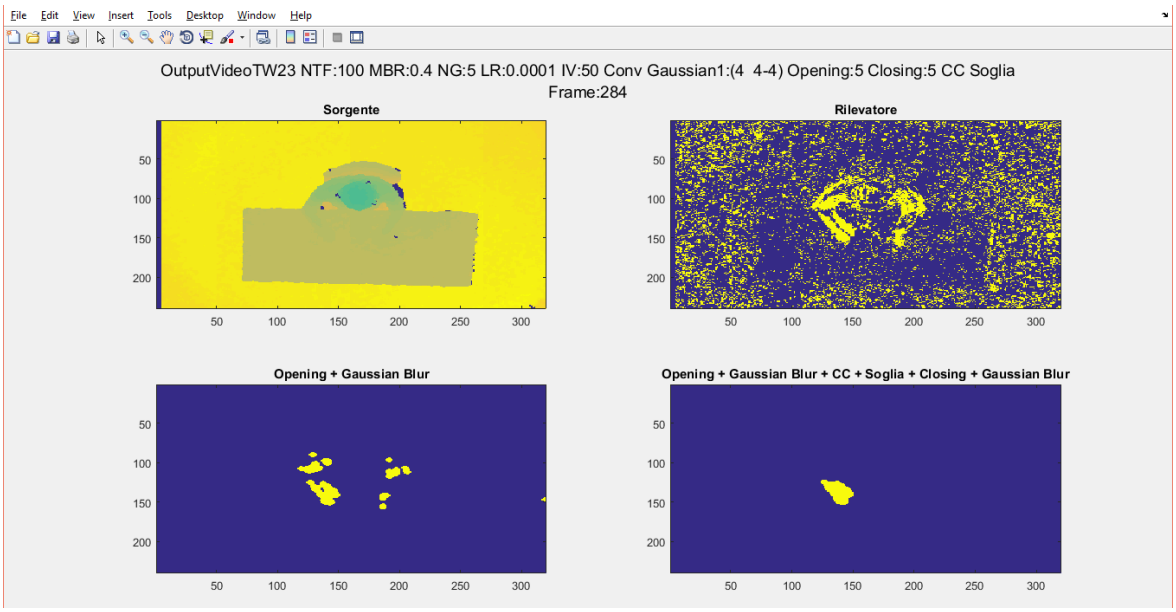


*Figura 32. Esempio di acquisizione in dinamico*

### 5.2 Test di estrazione statica

Per la riuscita di questo progetto, abbiamo preso in considerazione i test di estrazione statica, che come vediamo dai risultati delle tabelle sottostanti, risultano più funzionali al lavoro che è stato svolto.

| OnesCons=9 | ActionNumbers FrameRate=30 | Azioni FrameRate=15 | Azioni FrameRate=6 | Azioni FrameRate=3 |
|---|---|---|---|---|
| Test_21 | 3 | 3 | 2 | 2* |
| Test_23 | 19 | 19 | 19 | 13* |
| Test_24 | 21 | 21 | 21 | 18* |
| Test_25 | 31 | 26 | 26 | 14* |

| **OnesCons=15** | **ActionNumbers FrameRate=30** | **Azioni FrameRate=15** | **Azioni FrameRate=6** | **Azioni FrameRate=3** |
|---|---|---|---|---|
| **Test_21** | **2** | **2** | **2** | **2** |
| **Test_23** | **19** | **19** | **19** | **13** |
| **Test_24** | **21** | **21** | **21** | **18** |
| **Test_25** | **31** | **26** | **26** | **14** |

Di seguito vengono definiti lo StartFrame e l'EndFrame per ogni test.

| | Test_21 | Test_23 | Test_24 | Test_25 |
|---|---|---|---|---|
| SF | 162 | 1 | 56 | 138 |
| EF | 499 | 3599 | 3599 | 3599 |

I test ad estrazione statica, tranne il Test_21 che ne ha 500, presentano tutti 3600 frame. Un'acquisizione dell'ordine di 3000 frame permette di osservare in maniera ottimale un'azione food intake: le azioni sono maggiori sia in durata che in quantità e la diminuzione del FrameRate con il settaggio degli uni non rischia di fare perdere la maggior parte delle azioni visibili.

Anche qui, partendo con OnesCons=9, diminuendo il FrameRate, la funzione ActionNumbers rileva azioni da 9 ripetute (evidenziate nella tabella con un asterisco), non visibili a video. Per risolvere questa problematica, bisogna impostare OnesCons=15.

# 6. Conclusioni

La tesi presentata descrive l'implementazione di un algoritmo di estrazione delle azioni umane durante un pasto. Al fine di adattare l'algoritmo ai movimenti compiuti dai soggetti sotto test si è scelto di utilizzare l'algoritmo SOM Extended per l'elaborazione dei dati invece del SOM, in quanto risulta più preciso. L'algoritmo è risultato efficiente nel monitorare i movimenti del soggetto, tuttavia, è necessario un meccanismo di controllo, che evita l'unione dei giunti di mani e testa. Confrontando i due tool di elaborazione dati, il v_3 permette un'applicazione automatica in grado di elaborare i risultati in tempo reale: lo Start Frame e l'End Frame, ovvero rispettivamente i frame dai quali inizia e in cui finisce l'elaborazione dei test, vengono trovati automaticamente da una funzione sviluppata ad hoc che ne evita il setting manuale. L'inizio e la fine di un pasto risultano in questo modo calcolati con più precisione, così come il numero di azioni e la loro durata. Sono stati esaminati sia test ad estrazione statica del solo soggetto rispetto alla scena che test ad estrazione dinamica grazie ai quali si è notato come la durata di acquisizione possa influire sul conteggio e durata di un'azione rispetto ad una diminuzione del frame rate considerato. Un'acquisizione da 500 frame che presenta molte azioni di breve durata, verrà penalizzata in una riduzione del frame rate: le azioni verranno ridotte in quantità e in un confronto video, difficilmente si avrà una corrispondenza. Le azioni dell'ordine dei 3000 frame risultano più indicate per monitorare l'azione dell'abitudine alimentare a diversi frame rate. L'algoritmo ha ottenuto buone performance per la precisione con cui le azioni sono state calcolate durante il pasto. Il Test 25 è quello che ha presentato più errori, in quanto le azioni sono numerose e troppo ravvicinate, quindi ad una diminuzione del frame rate non si trova una corrispondenza precisa rispetto alle azioni realmente compiute dal soggetto. Un'ulteriore riduzione del frame rate (a 10 fps), come abbiamo visto dai risultati scritti nelle tabelle descritte nel Capitolo dei Risultati, non permette in nessun caso una visualizzazione corretta del numero di azioni. I movimenti di breve durata vengono eliminati dal confronto a video e nella maggior parte delle azioni dei test dell'ordine di 500 frame, il movimento della mano che si avvicina alla testa e che implica un'azione di introduzione del cibo viene persa.

# 7. Bibliografia


- [1] Enea Cippitelli, Samuele Gasparrini, Adelmo De Santis, Laura Montanini, Laura Raffaeli, Ennio Gambi e Susanna Spinsante "Comparison of RGB-D Mapping Solutions for Application to Food Intake Monitoring"
- [2] Ennio Gambi, Manola Ricciuti, Adelmo De Santis "Food Intake Actions Detection: An Improved Algorithm Toward Real-Time Analysis" Marzo 2020
- [3] Jie Sheng Tham, Yoong Choon Chang, Mohammad Faizal Ahmad Fauzi "Automatic Identification of Drinking Activities at Home using Depth Data from RGB-D Camera"
- [4] http://www.gamecompass.it/gli-esperimenti-cambiarono-modo-usare-kinect
- [5] https://www.hongkiat.com/blog/innovative-uses-kinect/
- [6] https://www.eurogamer.it/articles/2018-03-15-il-grande-furto-darte-fatto-con-kinect-articolo
- [7] https://it.wikipedia.org/wiki/Microsoft_Kinect
- [8] "Kinect for Windows Sensor Components and Specifications" https://msdn.microsoft.com/en-us/library/jj131033.aspx
- [9] "Kinect hardware key features and benefits" https://developer.microsoft.com/en-us/windows/kinect/hardware
- [10] https://skarredghost.com/2016/12/02/the-difference-between-kinect-v2-and-v1/
- [11] https://www.industriaitaliana.it/intelligenza-artificiale-microsoft-vara-un-progetto-da-25-milioni-di-dollari/
- [12] https://docs.microsoft.com/it-it/azure/kinect-dk/
- [13] https://lightbuzz.com/body-tracking-sensors-2017/
- [14] https://orbbec3d.com/orbbec-is-the-replacement-for-kinect-skeletal-tracking/
- [15] Marco Cristiani, "Sistemi avanzati per il riconoscimento: Lezione4 https://profs.sci.univr.it
- [16] Chris Stauffer e W.E.L Grimson " Adaptive background mixture models for real-time tracking" 1999
- [17] Computer Vision System Toolbox https://it.mathworks.com/products/computer-vision/
- [18] vision.ForegroundDetector System object https://it.mathworks.com/help/vision/ref/vision.foregrounddector-class.html
- [19] Samuele Gasparrini"Activity Monitoring and Behaviour Analysis Using RGB-Depth Sensors and Wearable Devices for Ambient Assisted Living Applications" Appendice

- [20] Raffaele Gaetano, Corso di Elaborazione di Segnali Multimediali: Elaborazione Morfologica delle Immagini
- [21] https://www.mathworks.com/help/images/ref/strel.html
- [22] Samuele Gasparrini“Activity Monitoring and Behaviour Analysis Using RGB-Depth Sensors and Wearable Devices for Ambient Assisted Living Applications” Paragrafo 5
- [23] https://en.wikipedia.org/wiki/Point_cloud
- [24] T. Kohonen, Self-Organizing Maps. Springer-Verlag Berlin Heidelberg, 2001
- [25] SOM, "Unobtrusive Intake Actions Monitoring Trough RGB and Depth Information Fusion"
- [26] SOM_EX, Ennio Gambi, Manola Ricciuti, Adelmo De Santis “Food Intake Actions Detection: An Improved Algorithm Toward Real-Time Analysis” Marzo 2020
- [27] Farooq, M.; Sazonov, E. A novel wearable device for food intake and physical activity recognition. Sensors
- [28] Al-Anssari, H.; Abdel-Qader, I. Vision based monitoring system for Alzheimer’s patients using controlled bounding boxes tracking. 2016 IEEE International Conference on Electro Information Technology (EIT).
- [29] Cunha, A.; Pádua, L.; Costa, L.; Trigueiros, P. Evaluation of MS Kinect for elderly meal intake monitoring.
- [30] Tham, J.S.; Chang, Y.C.; Fauzi, M.F.A. Automatic identification of drinking activities at home using depth data from RGB-D camera. The 2014 International Conference on Control, Automation and Information Sciences (ICCAIS 2014).