



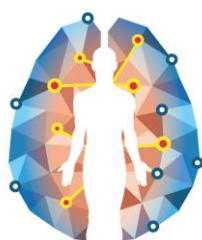
UNIONE EUROPEA



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



AIUTI A SOSTEGNO DEI CLUSTER TECNOLOGICI REGIONALI PER L'INNOVAZIONE



PERSON

PERSON - PERVERSIVE GAME FOR PERSONALIZED TREATMENT OF COGNITIVE AND FUNCTIONAL DEFICITS ASSOCIATED WITH CHRONIC AND NEURODEGENERATIVE DISEASES

Deliverable D10.4 – Progettazione del set-up di calcolo FPGA based e degli algoritmi programmazione FPGA e valutazioni comparative fra gli approcci usati



Codice Progetto	:	LQ8FBY0
Titolo Progetto	:	PERSON

Numero Deliverable	:	D10.4
Titolo Deliverable	:	Progettazione del set-up di calcolo FPGA based e degli algoritmi programmazione FPGA e valutazioni comparative fra gli approcci usati
Natura del Deliverable	:	Report
Mese di Rilascio da Capitolato	:	Luglio 2017
Livello di Disseminazione	:	Pubblico
Versione	:	1.0
Data Deliverable	:	4/8/2017

Abstract

Dopo aver individuato la piattaforma di calcolo FPGA, vengono descritte le fasi di progettazione ed implementazione del set-up di calcolo. Si spiega quindi il pacchetto software realizzato per interfacciarsi con il modulo hardware.

Infine si confrontano le prestazioni del SDVA con scheda FPGA e quelle del SDVA basato su GPU.

Keyword list

FPGA, frame grabber, elaborazioni di immagini, prestazioni di elaborazione

Tracking delle Versioni

Version	Changes	Author(s)
1.0	First Draft	Maria Francesca de Ruvo, Marco Colaprico

Indice

ABSTRACT	2
KEYWORD LIST.....	2
INDICE	3
INDICE FIGURE	5
INDICE GRAFICI	5
INDICE TABELLE.....	6
GLOSSARIO, ACRONIMI E ABBREVIAZIONI	7
1. INDIVIDUAZIONE DELLA PIATTAFORMA DI CALCOLO FPGA.....	8
1.1 BANDA RICHIESTA DAL SET-UP DI VISIONE ARTIFICIALE	8
1.2 CONFRONTO TRA GLI STANDARD HARDWARE DI VISIONE	8
1.2.1 Camera Link	9
1.2.2 Camera Link HS.....	9
1.2.3 CoaXPRESS	9
1.2.4 GigE Vision	9
1.2.5 USB3.....	9
1.2.6 Scelta dello standard	10
1.3 SCELTA DEL FRAME GRABBER CON SCHEDA FPGA A BORDO	10
2. OTTIMIZZAZIONE DI ALGORITMI DI ANALISI TESI ALLA RICOSTRUZIONE 3D DELLA GESTUALITÀ O DEL MOTO DEL PLAYER DEL SERIOUS GAMES, PARTENDO DA UN FLUSSO VIDEO STEREO @50FPS, PERCHÉ ESSI VENGANO MAPPATI IN HARDWARE	12
2.1 COMPONENTI HARDWARE DELL'SDVA	13
2.1.1 Markers.....	13
2.1.2 Illuminatori	13
2.1.3 Filtri.....	13
3. PROGETTAZIONE DI UN SET UP DI CALCOLO BASATO SU HARDWARE CONFIGURABILE.....	14
3.1 ARCHITETTURA DEL SISTEMA DI VISIONE ARTIFICIALE	14
3.2 FLOW CHART CONCETTUALE DELL'ALGORITMO DI VISIONE	15
4. IMPLEMENTAZIONE DEL SET UP INDIVIDUATO NEL CAPITOLO 3 SULLA PIATTAFORMA INDIVIDUATA NEL CAPITOLO 1	16
4.1 MAPPATURA HARDWARE DELL'ALGORITMO DI VISIONE	17
4.2 TEST E DEBUG DEL MODULO FPGA EMBEDDED	20
4.3 SINTESI E DOWNLOAD	22
5. INTEGRAZIONE DEL MODULO IMPLEMENTATO CON IL SET UP DI VISIONE ARTIFICIALE	22
5.1 SINCRONIZZAZIONE DELLE TELECAMERE	22
5.1.1 Modulo Hardware.....	23
5.1.2 Impostazioni software	24
5.1.3 Valutazioni sul modulo di sincronizzazione implementato.....	24
5.2 INTERFACCIA SOFTWARE.....	25
5.2.1 Modulo di acquisizione	25
5.2.2 Modulo di processing.....	28
6 TEST E DEBUG DEL SISTEMA INTEGRATO	29
6.1 PROBLEMA DELL'INSTABILITÀ DEL PUNTAMENTO.....	29

6.2 SOLUZIONE ALL'INSTABILITÀ DEL PUNTAMENTO	32
6.3 VALUTAZIONE PRESTAZIONI.....	35
7 APPLET PERSON1.6.....	37
7.1 CARATTERISTICHE	37
7.2 PARAMETRI	37
7.3 BANDA	38
7.4 REQUISITI.....	38
7.4.1 <i>Requisiti Software</i>	39
7.4.2 <i>Requisiti Hardware</i>	39
7.4.3 <i>Licenza</i>	39
7.5 INTERFACCIA DELLE TELECAMERE.....	39
7.6 TRASFERIMENTO DELLE IMMAGINI E DEI DATI ALLA MEMORIA DEL PC.....	39
8 GBECAMERASDK	39
8.1 STRUTTURA	39
8.2 REQUISITI.....	40
8.3 INTERFACCIA DELLE TELECAMERE.....	40
APPENDICE 1 – STRUTTURA DEL LOG DEL MODULO DI PROCESSING	40
APPENDICE 2 – STRUTTURA DEL LOG DEL MODULO DI ACQUISIZIONE	41
APPENDICE 3 – STRUTTURA DEL LOG DEL TEST DI INSTABILITÀ DEL PUNTAMENTO	42
ALLEGATI DEL MODULO HARDWARE	43
ALLEGATI DEL PACCHETTO SOFTWARE.....	43
ALLEGATI LOG MODULO PROCESSING	43
ALLEGATI LOG MODULO ACQUISIZIONE.....	43
ALLEGATI LOG TEST DELL'INSTABILITÀ DEL PUNTAMENTO	44
BIBLIOGRAFIA	44

Indice figure

FIGURA 1. SCHEMATIZZAZIONE DELLA CONFIGURAZIONE PROTOTIPALE DEL SISTEMA PERSON, IN CUI IL GAME SARÀ FRUITO SU UN NORMALE SCHERMO LCD.	8
FIGURA 2. MEIV VQ4-GPoE CONNESSO AL MODULO DI ALIMENTAZIONE.	11
FIGURA 3. VIDEO PROCESSOR MEIV-VQ4GE DELLA SILICON SOFTWARE.	12
FIGURA 4. MARCATORE PASSIVO DOTATO DI RIVESTIMENTO RIFLETTEnte 3M 7610 E FILETTATURA INTERNA PER IL FISSAGGIO.....	13
FIGURA 5. OVERDRIVE RING LIGHT 850NM IR - EZ MOUNT RING LIGHT.....	13
FIGURA 6. CARATTERISTICA “TRASMISSIONE= $f(\lambda)$ ” DEL FILTRO BN850-27.	14
FIGURA 7. ARCHITETTURA DEL SDVA (SENSORE DI VISIONE ARTIFICIALE).	15
FIGURA 8. FLOW CHART CONCETTUALE DELL’ALGORITMO DI VISIONE.	16
FIGURA 9. DIAGRAMMA A BLOCCHI DEL FLUSSO DI LAVORO.....	17
FIGURA 10. TELECAMERA JAI GO-5000-PGE.....	17
FIGURA 11. PROGETTO VISUAL APPLLET DELL’ALGORITMO DI VISIONE RELATIVO AL PROCESSO 0.....	18
FIGURA 12. DISEGNO RELATIVO AL BLOCCO MERGE DEL PROCESSO 0.	19
FIGURA 13. RISORSE FPGA UTILIZZATE DALL’ALGORITMO DI VISIONE.....	20
FIGURA 14. IMMAGINE BINARIA UTILIZZATA PER IL TEST FUNZIONALE DELL’APPLET.....	21
FIGURA 15. RISULTATI DELL’ANALISI DEI BLOB. È POSSIBILE LEGGERE IL CENTRO DI MASSA, L’AREA E IL BYTE RELATIVO AGLI ERRORI DELL’ANALISI, DELLE PRIME TRE REGIONI DELL’IMMAGINE IN FIGURA 13.....	21
FIGURA 16 RISULTATO DELL’ANALISI DI BANDA DELL’APPLET PERSON1.3 OTTENUTO IMPOSTANDO COME ROI LA DIMENSIONE MASSIMA DEL FRAME 2560*2048	22
FIGURA 17 DIAGRAMMA DELLA PARTE ESTERNA DEL CIRCUITO DI INGRESSO	23
FIGURA 18 DIAGRAMMA DELLA PARTE ESTERNA DEL CIRCUITO DI USCITA	23
FIGURA 19 VISTA DALL’ESTERNO DEL CONNETTORE HIROSE A 6 PIN E TABELLA DELL’ASSEGNAmento DEI PIN	23
FIGURA 20 CIRCUITO PER LA SINCRONIZZAZIONE DELLE TELECAMERE.....	24
FIGURA 21 COMPONENTI DEL SISTEMA DI VISIONE ARTIFICIALE	25
FIGURA 22 DIAGRAMMA DELLE CLASSI DI GbESTEREO	26
FIGURA 23 DIAGRAMMA DELLE CLASSI DI BLOB2D.....	27
FIGURA 24 DISEGNO RELATIVO AL BLOCCO CALCULATE_NORMALIZED_COG	32
FIGURA 25 PROGETTO VISUAL APPLET DELL’ALGORITMO DI VISIONE RELATIVO AL PROCESSO 0.....	33
FIGURA 26 RISORSE FPGA UTILIZZATE DALL’APPLET PERSON1.6	34
FIGURA 27 CONFRONTO DEI TEMPI DI ELABORAZIONE	35
FIGURA 28 INTERFACCIA DELLE TELECAMERE E SETUP DEI CAVI.....	39
FIGURA 29 INTERFACCIA DELLE TELECAMERE E SETUP DEI CAVI.....	40

Indice grafici

GRAFICO 1 DIFFERENZA TRA L’ISTANTE DI ACQUISIZIONE DELLA TELECAMERA MASTER E DELLA SLAVE.....	25
GRAFICO 2 GRAFICO DELLA COORDINATA X DEL PIXEL SELEZIONATO DAL SDVA AL VARIARE DEL TEMPO, MENTRE IL PIXEL PUNTATO DALL’UTENTE RIMANE COSTANTE PER 24 SECONDI	30
GRAFICO 3 GRAFICO DELLA COORDINATA Y DEL PIXEL SELEZIONATO DAL SDVA AL VARIARE DEL TEMPO, MENTRE IL PIXEL PUNTATO DALL’UTENTE RIMANE COSTANTE PER 24 SECONDI	30
GRAFICO 4 GRAFICO DELLA COORDINATA X DEL PIXEL SELEZIONATO DAL SDVA AL VARIARE DEL TEMPO, MENTRE IL PIXEL PUNTATO DALL’UTENTE RIMANE COSTANTE PER 24 SECONDI. RELATIVO AL TRONCAMENTO DEI CENTRI DI MASSA DEI MARKERS RISPETTIVAMENTE A 0, 1, 2, 3 E 6 CIFRE DECIMALI.....	30
GRAFICO 5 GRAFICO DELLA COORDINATA Y DEL PIXEL SELEZIONATO DAL SDVA AL VARIARE DEL TEMPO, MENTRE IL PIXEL PUNTATO DALL’UTENTE RIMANE COSTANTE PER 24 SECONDI. RELATIVO AL TRONCAMENTO DEI CENTRI DI MASSA DEI MARKERS RISPETTIVAMENTE A 0, 1, 2, 3 E 6 CIFRE DECIMALI.....	31
GRAFICO 6 GRAFICO DELLA COORDINATA X DEL PIXEL SELEZIONATO DAL SDVA AL VARIARE DEL TEMPO, MENTRE IL PIXEL PUNTATO DALL’UTENTE RIMANE COSTANTE PER 24 SECONDI. RELATIVO AL TRONCAMENTO A 0 CIFRE DECIMALI DEI CENTRI DI MASSA DEI MARKERS	31
GRAFICO 7 GRAFICO DELLA COORDINATA Y DEL PIXEL SELEZIONATO DAL SDVA AL VARIARE DEL TEMPO, MENTRE IL PIXEL PUNTATO DALL’UTENTE RIMANE COSTANTE PER 24 SECONDI. RELATIVO AL TRONCAMENTO A 0 CIFRE DECIMALI DEI CENTRI DI MASSA DEI MARKERS.....	32

GRAFICO 8 TEMPI DI ELABORAZIONE DELLA CPU DEL SDVA CON FRAME GRABBER.....	36
GRAFICO 9 DISTANZA INTERFRAME DEL SDVA CON FRAME GRABBER, NEL CASO DI FRAME RATE DI 50FPS.....	36
GRAFICO 10 DISTANZA INTERFRAME DEL SDVA CON FRAME GRABBER, NEL CASO DI FRAME RATE DI 56FPS.....	37

Indice tabelle

TABELLA 1. CONFRONTO TRA GLI STANDARD HARDWARE DI VISIONE ARTIFICIALE. IN VERDE SONO EVIDENZIATE LE CARATTERISTICHE DEI PROTOCOLLI CHE VERIFICANO LE SPECIFICHE DEL SET UP DI VISIONE PER LA RICOSTRUZIONE DELLA GESTUALITÀ DEL PAZIENTE.	10
TABELLA 2. CARATTERISTICHE DEL DISPOSITIVO MICROENABLE IV VQ4-GE.	11
TABELLA 3. SOMMARIO DEGLI ATTRIBUTI DEI DISPOSITIVI DELLA FAMIGLIA SPARTAN-3 FPGA. IN VERDE È EVIDENZIATO L'FPGA MONTATO SUL FRAME GRABBER MICROENABLE IV VQ4-GE.....	12
TABELLA 4. TABELLA SINTETICA DELL'UTILIZZO DELLE RISORSE DELL'FPGA.....	19
TABELLA 9 CONFIGURAZIONE DELLA TELECAMERA MASTER.....	24
TABELLA 10 CONFIGURAZIONE DELLA TELECAMERA SLAVE	24
TABELLA 11 TABELLA SINTETICA DELL'UTILIZZO DELLE RISORSE DELL'FPGA NEL PROGETTO CHE PREVEDE 6 DMA.....	33
TABELLA 12 TABELLA SINTETICA DELL'UTILIZZO DELLE RISORSE DELL'FPGA NEL PROGETTO CHE PREVEDE 4 DMA.....	33
TABELLA 13 TABELLA DEGLI INDICI DELLE RISORSE UTILIZZATE	34
TABELLA 5 SOMMARIO DELLE CARATTERISTICHE DELL'APPLET PERSON1.3.....	37
TABELLA 6 PROPRIETÀ DEI PARAMETRI DEL PROCESS0.....	38
TABELLA 7 PROPRIETÀ DEI PARAMETRI DEL PROCESS1.....	38
TABELLA 8 BANDA DI PERSON1.3	38

Glossario, acronimi e abbreviazioni

Item	Descrizione
SDVA	Set up di visione artificiale
3D	Tridimensionale
2D	Bidimensionale
LED	Light Emitting Diode
IR	Infrarosso
ROI	Region Of Interest, cioè la regione di interesse rispetto all'intero frame
nm	Nanometro ($1 \text{ nm} = 10^{-9} \text{ m}$)
ms	Millisecondi ($1 \text{ ms} = 10^{-3} \text{ secondi}$)
GigE	Standard hardware di visione Giga Ethernet Vision
HS	High Speed
USB	Universal Serial Bus
PoCL	Power over Camera Link
PoE	Power over Ethernet
fps	Frame per second, cioè fotogrammi al secondo
Mpixel	Megapixel
λ	Lunghezza d'onda
FPGA	Field Programmable Gate Array, è un circuito integrato le cui funzionalità sono programmabili via software
Gbits/s	Giga bits al secondo
MB/s	Mega bytes al secondo
MHz	Mega Hertz
GPU	Graphical Processing Unit
CPU	Central Processing Unit
PCIe	Peripheral Component Interconnect Express, standard di interfaccia d'espansione a bus seriale per computer usato per connettere periferiche alla scheda madre
DMA	Direct Memory Access
MEIV	microEnable IV
RAM	Random Access Memory
M	Master
S	Slave
DC	Direct Current
GPI	General Purpose Input/Output
Hirose 6	Connettore Hirose a 6 pin

1. Individuazione della piattaforma di calcolo FPGA

La necessità di elaborare immagini sempre più grandi e a frequenze sempre più elevate, ha portato alla richiesta di performances sempre più proibitive dei sistemi host. A questa esigenza il mercato della visione ha risposto con l'introduzione di frame grabber che montano a bordo un FPGA, consentendo di ridurre il carico di lavoro sull'host, scaricando le operazioni di image processing ripetitive ed intensive sulla scheda FPGA e liberando così risorse di calcolo preziose per il resto dell'applicazione con un conseguente miglioramento delle performances. Le operazioni candidate ad essere eseguite sulla piattaforma FPGA includono il filtraggio spaziale e temporale, il guadagno e la correzione dell'offset, la correzione dei pixel morti, la correzione della distorsione ottica e della prospettiva, le conversioni spazio colore e le trasformazioni nel dominio della frequenza.

La scelta della piattaforma di calcolo FPGA, verrà effettuata tra quelle montate sui frame grabber disponibili sul mercato, in modo tale da avere un'unica scheda hardware che effettua sia l'acquisizione che una parte dell'elaborazione a bordo dell'FPGA.

Prima di effettuare la scelta dell'FPGA è quindi necessario scegliere il protocollo di visione del frame grabber. Tale scelta verrà effettuata tra gli standard hardware in uso nell'industria della visione, alla luce delle specifiche di banda richieste del set up dell'SDVA.

1.1 Banda richiesta dal Set-up di visione artificiale

Il set up di visione artificiale per la ricostruzione della gestualità del paziente, deve determinare in tempo reale e al throughput di 50 rilevazioni al secondo, quale è il pixel che il player sta indicando con la propria mano su uno schermo di tipo convenzionale (Figura 1), e fornire al sensor hub le coordinate di detto pixel, per permettere alle altre componenti del game una fluida interazione con il player.

Dato il volume oggetto dell'osservazione, inferiore ad 1 m^3 , si ritiene che telecamere con sensore di almeno 2-Mpixel (1920x1080) garantiscano una risoluzione adeguata. Esse dovranno avere compattezza, appropriata sensitività, oltre che consentire un frame rate di almeno 50 fps.

Il set-up di visione appena stabilito richiede una banda minima di 1Gigabit/s (1920x1080x8x50).



Figura 1. Schematizzazione della configurazione prototipale del sistema PERSON, in cui il game sarà fruito su un normale schermo LCD.

1.2 Confronto tra gli standard hardware di visione

Gli standard hardware, in uso nell'industria della visione artificiale, che garantiscono alte prestazioni, sono i seguenti (1):

- CameraLink
- CameraLinkHS
- CoaXPress
- GigE Vision
- USB3 Vision

1.2.1 Camera Link

Lo standard Camera Link è stato inizialmente rilasciato nel 2000. È un protocollo di comunicazione che standardizza la connessione tra la telecamera e il frame grabber e definisce un'interfaccia completa che comprende il trasferimento dei dati, il clock ed una segnalazione real time alla telecamera. Si tratta di un protocollo di comunicazione seriale ideato per le comunicazioni ad alta velocità, che garantisce una banda di 2.04Gbits/s nella modalità Base (un solo cavo) e 6.8Gbits/s nella modalità Full (due cavi). La lunghezza del cavo può essere al massimo di 10m. Inoltre una telecamera PoCL può essere alimentata mediante un frame grabber PoCL e un cavo Camera Link.

1.2.2 Camera Link HS

Lo standard Camera Link HS è stato rilasciato nel maggio 2012 per migliorare quello Camera Link utilizzando dei cavi di rame e ottici per estendere la portata e migliorare la banda. Garantisce una banda di 3.125Gbits/s per linea con il protocollo M (cavi di rame) e 10.3Gbits/s con il protocollo X (cavi ottici). La massima lunghezza dei cavi di rame può arrivare a 15m, mentre quella dei cavi ottici a 600m. Tale standard offre anche la possibilità di effettuare l'elaborazione parallelamente su più frame grabber. A differenza dello standard Camera Link non è possibile alimentare le telecamere direttamente dal frame grabber.

1.2.3 CoaXPress

Lo standard CoaXPress (CXP) è stato rilasciato nel dicembre 2010. Esso fornisce un'interfaccia ad alta velocità tra la telecamera e il frame grabber e consente grandi lunghezze dei cavi. Nella forma più semplice questo standard utilizza un singolo cavo coassiale (CXP-1) per trasmettere i dati dalla telecamera al frame grabber fino a 1.25Gbits/s; contemporaneamente trasmette i dati di controllo e i triggers dal frame grabber alla telecamera a 20.8 Mbits/s; e fornisce fino a 13W di potenza alla telecamera. Quando sono necessarie velocità più alte si ricorre all'aggregazione dei link che consente di arrivare a 6.25Gbits/s (CXP-6). Alla velocità del link di 1.25Gbits/s (CXP-1), CoaXPress supporta cavi lunghi fino a 100m, mentre alla massima velocità 6.25Gbits/s (CXP-6) si possono utilizzare cavi lunghi al massimo 35mm con un diametro di 6mm.

1.2.4 GigE Vision

GigE Vision è uno standard ampiamente utilizzato ed è stato sviluppato sfruttando il protocollo di comunicazione Ethernet (IEEE 802.3). È stato rilasciato nel maggio 2006 e rivisto nel 2010 (versione 1.2) e nel 2011 (versione 2.0). GigE Vision supporta più canali di stream e consente il trasferimento dell'immagine senza errori su distanze fino a 100m mediante cavi standard Ethernet e fino a 5Km su fibra ottica. Attualmente sono già disponibili sistemi che garantiscono una banda di 1 e 2 Gbits/s (usando 2 cavi) e stanno entrando nel mercato sistemi con una banda di 10Gbits/s. Con questo protocollo non è necessario utilizzare un frame grabber per l'acquisizione. Se si utilizza una scheda di rete o un frame grabber PoE è possibile alimentare le telecamere direttamente tramite cavo Ethernet.

1.2.5 USB3

Lo standard USB3 è stato introdotto a fine 2011 e una nuova versione è stata pubblicata nel gennaio 2013. L'interfaccia USB porta ad una installazione plug and play, ad alti livelli di performances (fino a 400Mbytes/s) e alla possibilità di trasferire direttamente le immagini dalla telecamera alla memoria del PC senza necessariamente utilizzare un frame grabber. La massima lunghezza dei cavi standard USB può arrivare a 5m.

1.2.6 Scelta dello standard

Tutti e cinque gli standard considerati garantiscono la banda di 1Gbits/s richiesta dal set-up di visione artificiale presentato nel primo paragrafo di questo capitolo. Ma dal confronto delle caratteristiche dei vari protocolli si è scelto di optare per lo standard GigE Vision perché sul mercato ci sono numerose telecamere che utilizzano tale standard, non necessita di un frame grabber e quindi consente di effettuare una comparazione tra gli algoritmi di visione sviluppati per FPGA e quelli per GPU, a differenza dello standard USB3 e del CameraLink consente una maggiore connettività dato che la massima lunghezza dei cavi può arrivare a 100m e, se si utilizza un frame grabber o una scheda di rete PoE, è possibile alimentare la telecamera direttamente dal cavo di rete in modo da semplificare il cablaggio e la complessità del sistema.

Standard					
Banda	Base Mode: 2.04Gbits/s	M protocol (cavo di rame): 3.125Gbits/s	CXP-1: 1.25Gbits/s	Un solo cavo: 1Gbits/s	5Gbits/s
	Full Mode: 6.8Gbits/s	X protocol (fibra ottica): 10.3Gbits/s	CXP-6: 6.26Gbits/s	Due cavi: 2Gbits/s	
Lunghezza massima dei cavi	10 m	Cavo di rame: 15m	CXP-1: 100m	100m	5m
		Fibra ottica: 500m	CXP-6: 35m		
Alimentazione sul cavo	Opzionale	No	Sì	Opzionale	Sì
Necessità del frame grabber	Sì	Sì	Sì	No	No

Tabella 1. Confronto tra gli standard hardware di visione artificiale. In verde sono evidenziate le caratteristiche dei protocolli che verificano le specifiche del set up di visione per la ricostruzione della gestualità del paziente.

1.3 Scelta del frame grabber con scheda FPGA a bordo

GigE Vision si è rivelato essere lo standard più appropriato al set-up di visione scelto. La scelta della piattaforma di calcolo è quindi limitata ai frame grabbers GigE Vision che montano a bordo un FPGA.

Un studio di mercato ha rivelato che la maggior parte dei produttori di video processor realizzano frame grabber con a bordo schede FPGA solo per gli standard CameraLink e CoaXPress (2) (3) (4). Mentre Silicon Software (5) realizza frame grabbers dotati di scheda FPGA per quasi tutti gli standard analizzati (CameraLink, CameraLinkHS, CoaXPress, GigE Vision). Per gli standard CameraLink e CoaXPress, la Silicon Software mette a disposizione la possibilità di poter scegliere tra più serie di prodotti (microEnable 5 ironman (6) che monta a bordo uno Xilinx FPGA Virtex 6 series, microEnable 5 marathon (7) che monta a bordo uno Xilinx FPGA of Kintex 7 series, LightBridge (8) e microEnable IV (9)). Per lo standard GigE Vision è disponibile sono la serie microEnable IV (9) e in particolare sono disponibili 4 prodotti:

1. microEnable IV AQ4-GE (10)
2. microEnable IV AQ4-GPoE (11)

3. microEnable IV VQ4-GE (12)
4. microEnable IV VQ4-GPoE (13)

I primi 2 dispositivi elencati appartengono alla serie dei frame grabber non programmabili (A series) e vanno quindi scartati, mentre gli ultimi 2 fanno parte della serie programmabile (V series). La Silicon Software per la programmazione dell’FPGA mette a disposizione un ambiente di sviluppo grafico (Visual Applet) che rende semplice l’implementazione di algoritmi di elaborazione dell’immagine e che può essere utile per ridurre i tempi di sviluppo.

Gli ultimi due dispositivi dell’elenco differiscono solo per la capacità di supportare delle telecamere Power over Ethernet (PoE). Il frame grabber microEnable IV VQ4-GPoE però, per il corretto funzionamento necessita un ulteriore modulo di alimentazione, come è possibile notare in **Errore. L'origine riferimento non è stata trovata.** Per evitare di montare sul pc anche il modulo di alimentazione, si può utilizzare il frame grabber microEnable IV VQ4-GE e alimentare le telecamere GigE mediante il cavo di alimentazione.

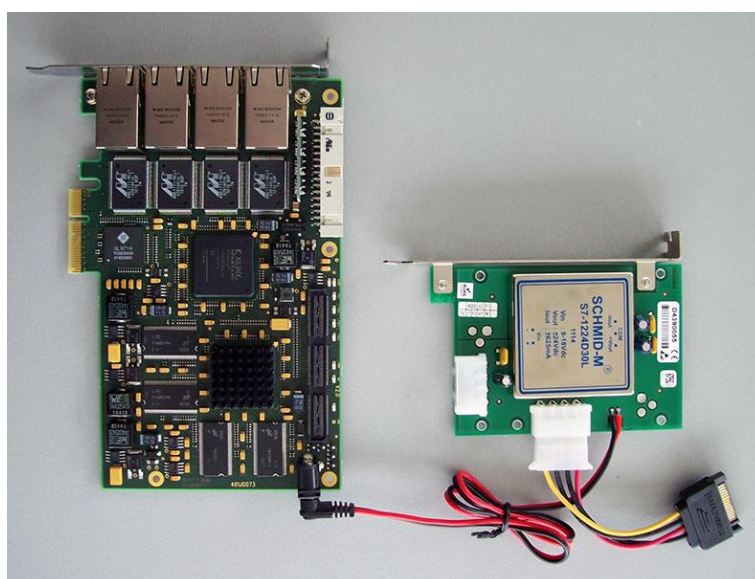


Figura 2. MEIV VQ4-GPoE connesso al modulo di alimentazione.

Le caratteristiche del frame grabber microEnable IV VQ4-GE sono elencate nella Tabella 2. Come di può notare il frame grabber consente l’interfacciamento con quattro telecamere GigE, ciò può risultare utile per una possibile futura espansione dall’attuale configurazione ad un set up che comprenda 2 coppie stereo.

Caratteristiche del dispositivo	
Processore	Vision Processor, Spartan3 FPGA XC3S4000
Memoria a bordo del dispositivo	512 MByte DDR-RAM
Modulo di interfaccia I/O	Trigger/GPIO-IF (Opto Trigger, TTL Trigger)
Numero di Porte GigE	4
PC Bus Interface	PCI Express x4 (Gen1), DMA900

Tabella 2. Caratteristiche del dispositivo microEnable IV VQ4-GE.

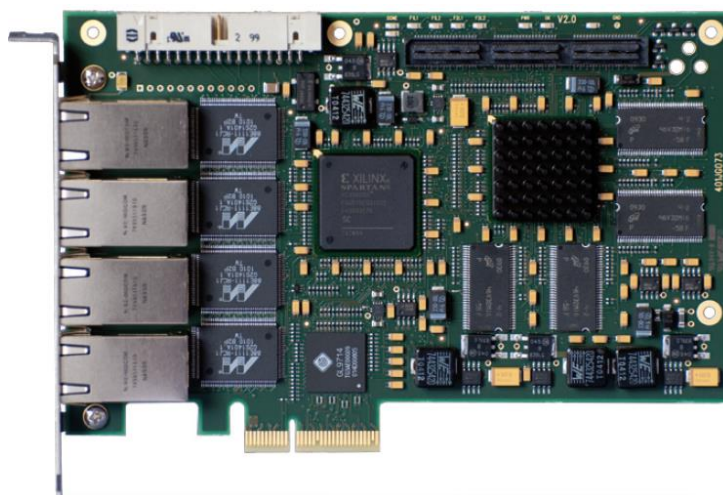


Figura 3. Video processor MEIV-VQ4GE della Silicon Software.

Si può concludere che il frame grabber microEnable IV VQ4-GE della Silicon Software, verifica tutti i requisiti del set-up scelto per l'SDVA in quanto è basato sullo standard hardware:GigE Vision, ha un numero di porte GigE maggiore di due e monta a bordo come FPGA programmabile lo Spartan3 della Xilinx (14).

Device	System Gates	Equivalent Logic Cells ⁽¹⁾	CLB Array (One CLB = Four Slices)			Distributed RAM Bits (K=1024)	Block RAM Bits (K=1024)	Dedicated Multipliers	DCMs	Max. User I/O	Maximum Differential I/O Pairs
			Rows	Columns	Total CLBs						
XC3S50 ⁽²⁾	50K	1,728	16	12	192	12K	72K	4	2	124	56
XC3S200 ⁽²⁾	200K	4,320	24	20	480	30K	216K	12	4	173	76
XC3S400 ⁽²⁾	400K	8,064	32	28	896	56K	288K	16	4	264	116
XC3S1000 ⁽²⁾	1M	17,280	48	40	1,920	120K	432K	24	4	391	175
XC3S1500	1.5M	29,952	64	52	3,328	208K	576K	32	4	487	221
XC3S2000	2M	46,080	80	64	5,120	320K	720K	40	4	565	270
XC3S4000	4M	62,208	96	72	6,912	432K	1,728K	96	4	633	300
XC3S5000	5M	74,880	104	80	8,320	520K	1,872K	104	4	633	300

Notes:

1. Logic Cell = 4-input Look-Up Table (LUT) plus a 'D' flip-flop. "Equivalent Logic Cells" equals "Total CLBs" x 8 Logic Cells/CLB x 1.125 effectiveness.
2. These devices are available in Xilinx Automotive versions as described in [DS314](#): Spartan-3 Automotive XA FPGA Family.

Tabella 3. Sommario degli attributi dei dispositivi della famiglia Spartan-3 FPGA. In verde è evidenziato l'FPGA montato sul frame grabber microEnable IV VQ4-GE.

2. Ottimizzazione di algoritmi di analisi tesi alla ricostruzione 3D della gestualità o del moto del player del serious games, partendo da un flusso video stereo @50fps, perché essi vengano mappati in hardware

Il workflow che si è deciso di adottare per la ricostruzione 3D del moto del player del serious game è quello che fa uso di markers. Poiché tra le operazioni candidate ad essere eseguite sulla piattaforma FPGA, vi sono la sogliatura del frame e le operazioni di estrazione delle features degli oggetti segmentati (l'area, la bounding box, il centro di massa ecc) si è deciso di non utilizzare dei markers grafici, ma bensì dei markers di tipo spot (led attivi o riflettori passivi).

Per ridurre l'operazione di segmentazione dei markers ad una semplice sogliatura, è necessario aumentare il contrasto tra marcatore e sfondo. Per i sistemi passivi si possono utilizzare dei markers di materiale fortemente retroriflettente e una fonte di illuminazione collocata in modo solidale ad ogni telecamera. Per quelli attivi, si può far uso di marcatori LED impulsati in modo sincrono con lo shutter delle telecamere digitali. In entrambi i tipi di sistemi, per ridurre l'interferenza di sorgenti luminose terse, si possono dotare

le telecamere di filtri ottici centrati sulla lunghezza d'onda della sorgente di illuminazione (in caso di marker attivi) o di emissione (marker attivi) LED.

Una differenza significativa tra i sistemi è che i marcatori attivi possono essere identificati in modo univoco grazie alla modulazione dei loro impulsi, mentre i sistemi passivi devono dedurre l'identità del marker dall'osservazione continua di questi ultimi. Entrambi i tipi di sistemi ottici generano in modo affidabile la posizione esatta dei marcatori.

Nella fattispecie, consultandosi con il partner SENSICHIPS, responsabile della progettazione e realizzazione del guanto aptico, si è optato per marcatori passivi, piuttosto che per quelli attivi, per sgravare il guanto stesso da problematiche legate all'alimentazione dei LED.

2.1 Componenti Hardware dell'SDVA

Alla luce delle precedenti considerazioni, il SDVA che si è deciso di realizzare sarà basato sulla componentistica di seguito descritta.

2.1.1 Markers

Come detto si opterà per marcatori passivi, per non gravare il sensore aptico con problematiche legate all'alimentazione di marcatori attivi. Nello specifico, si sono scelti dei marcatori sferici rivestiti da Reflective Tape "3M 7610" del diametro di $\frac{1}{4}$ di pollice (6.4 mm), che consentono il fissaggio sul guanto aptico, attraverso opportune viti, grazie ad una predisposizione filettata, come visibile in Figura 4



Figura 4. Marcatore passivo dotato di rivestimento riflettente 3M 7610 e filettatura interna per il fissaggio.

2.1.2 Illuminatori

Avendo optato per un approccio basato su marcatori passivi, è opportuno demandare l'illuminazione a sorgenti terze rispetto ai marcatori stessi, affinché essi siano comunque ben nitidi nelle immagini da elaborare.

Allo scopo di non creare disturbo al player e rendere il più neutro possibile il SDVA, si è considerata una lunghezza d'onda nella banda IR, e specificatamente quella ad 850 nm. Si è pertanto scelto di adoperare gli illuminatori ad anello "OverDrive Ring Light 850nm IR - EZ Mount Ring Light" della Smart Vision Light (Figura 5).



Figura 5. OverDrive Ring Light 850nm IR - EZ Mount Ring Light.

2.1.3 Filtri

Allo scopo di ridurre l'interferenza con sorgenti di luminosità terze, rendendo stabile il SDVA rispetto alla luminosità dell'ambiente, si doteranno le telecamere di filtri centrati su λ pari a 850 nm, come i BN850-27 della MIDWEST OPTICAL, che ha una trasmissione a 850 nm superiore all'85%, come risulta dalla caratteristica riportata in Figura 6

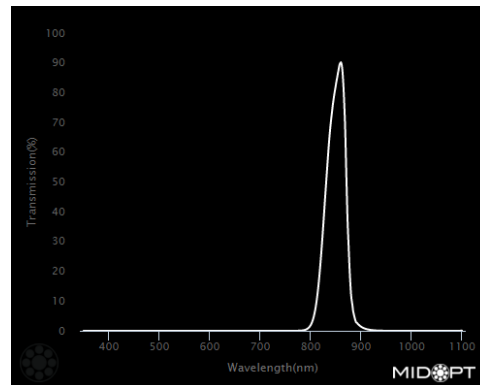


Figura 6. Caratteristica “Trasmissione=f(λ)” del filtro BN850-27.

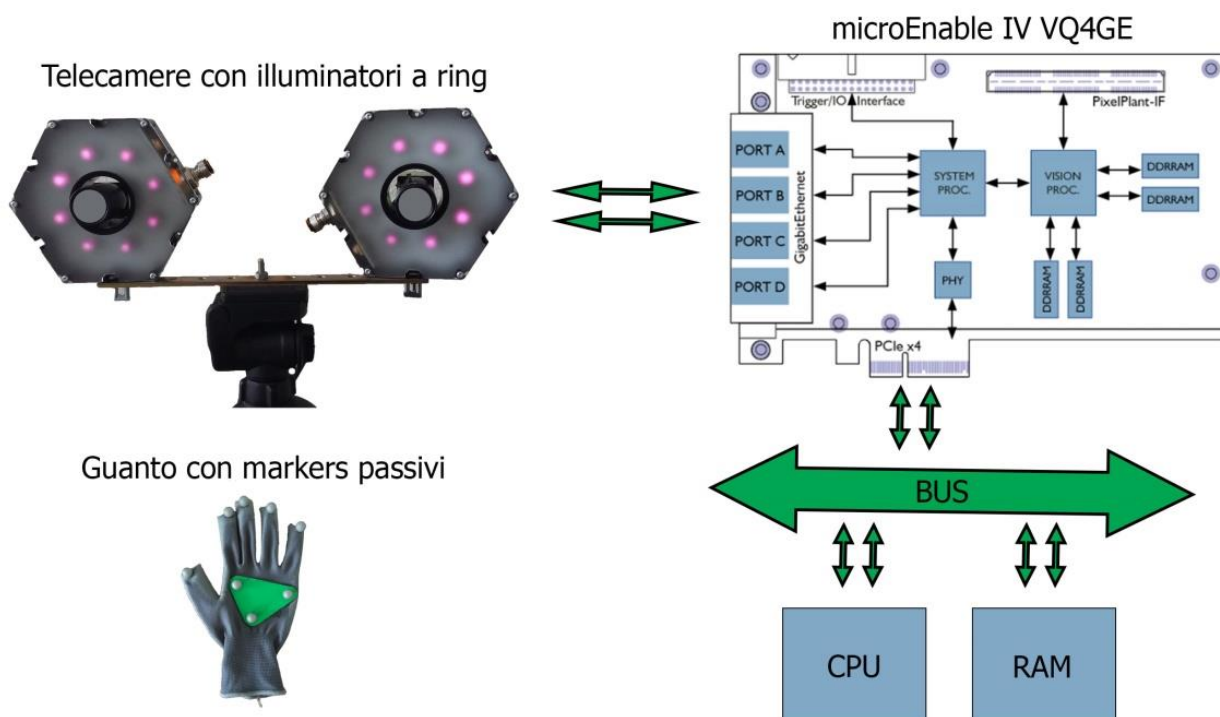
3. Progettazione di un set up di calcolo basato su hardware configurabile

In base alle considerazioni fatte nei punti A e B, viene definita l'architettura del SDVA e viene individuato il flow chart concettuale dell'algoritmo di visione.

3.1 Architettura del Sistema Di Visione Artificiale

Il SDVA prevede un hardware composto da una coppia di telecamere stereo GigE, munite ciascuna di un illuminatore IR ad anello, connesse alla scheda microEnable IV VQ4GE mediante le porte GigabitEthernet. Sul frame grabber è presente un FPGA programmabile (Spartan3 FPGA XC3S4000) che effettuerà una parte dell'elaborazione a bordo della scheda. Il frame grabber è collegato ad un PC Desktop tramite PCIe x4.

La Figura 7 mostra una rappresentazione grafica dell'architettura descritta.



3.2 Flow chart concettuale dell'algoritmo di visione

I dati provenienti dalle videocamere vengono in prima battuta processati dall'FPGA opportunamente programmato, per poi essere riversati in RAM. Un'applicazione software elabora ulteriormente questi dati al fine di ottenere le informazioni utili per il serious game.

In Figura 8 viene mostrato il flow chart concettuale dell'algoritmo di visione.

Come già anticipato precedentemente, per ridurre l'operazione di segmentazione dei markers ad una semplice sogliatura, è necessario aumentare il contrasto tra marcatore e sfondo, perciò sarà fondamentale impostare opportunamente l'otturatore delle telecamere e il loro tempo di esposizione in modo tale che nella scena siano visibili solo i marcatori passivi.

È importante sottolineare che la velocità del processo di analisi dei blob non è fissa e dipende dal contenuto dell'immagine e potrebbe rallentare l'applicazione se le immagini in ingresso sono molto rumorose. La riduzione di velocità dovrebbe essere limitata e irrilevante nelle applicazioni reali, tuttavia se si presenterà tale problema, sarà opportuno introdurre un blocco di pre-processing per velocizzare tale operazione.

Per la ricostruzione 3D delle coordinate dei markers è necessario risolvere il problema delle corrispondenze stereo, per semplificare tale problema è possibile rendere le linee epipolari delle due immagini parallele (le telecamere vengono ruotate virtualmente per rendere complanari i loro piani immagine). A tal fine è quindi necessario effettuare un remapping delle immagini provenienti dalle telecamere, poiché la struttura dell'FPGA non si presta particolarmente a tale scopo si è deciso di modificare il flusso dell'algoritmo e "rettificare" solo le coordinate 2D dei markers invece che l'intera immagine.

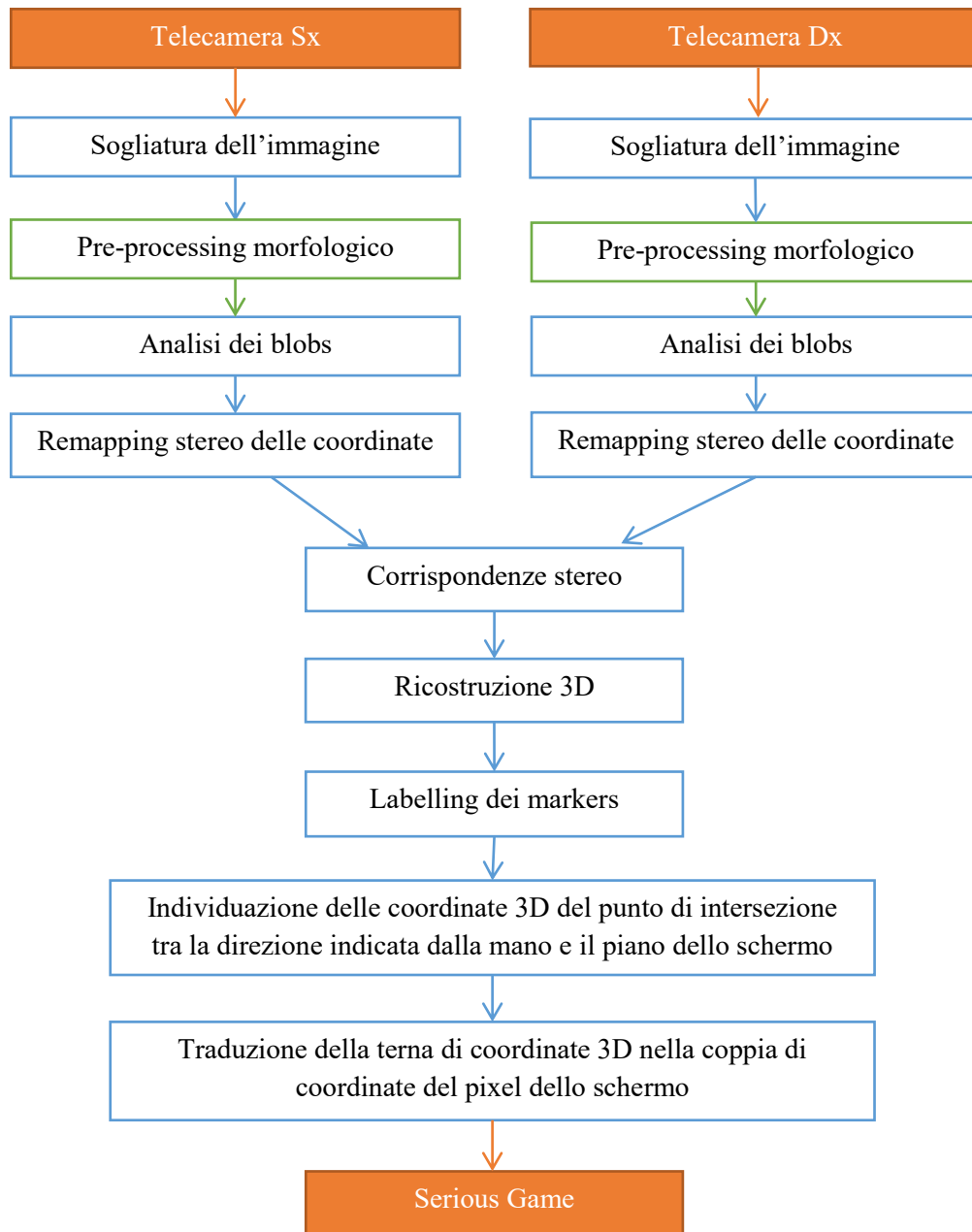


Figura 8. Flow chart concettuale dell'algoritmo di visione.

4. Implementazione del set up individuato nel capitolo 3 sulla piattaforma individuata nel capitolo 1

Come anticipato nel capitolo 1, la scheda microEnable IV VQ4GE appartiene alla serie V dei frame grabber della Silicon Software, e consente di programmare l'FPGA mediante il software Visual Applet. Questo tool mappa gli algoritmi di elaborazione dell'immagine come un flow charts grafico e mette a disposizione un'estesa libreria di operatori. Consente inoltre, prima della sintesi del progetto, una simulazione dei risultati visuali, con la precisione del bit, in ogni punto del modello del flusso dati. Il tool di sintesi dell'FPGA è integrato in Visual Applet e produce una applet hardware che andrà poi scaricata sulla scheda microEnable IV VQ4GE.

In **Errore**. L'origine riferimento non è stata trovata. viene mostrato il flusso di lavoro relativo all'implementazione dell'algoritmo di visione sulla scheda FPGA.

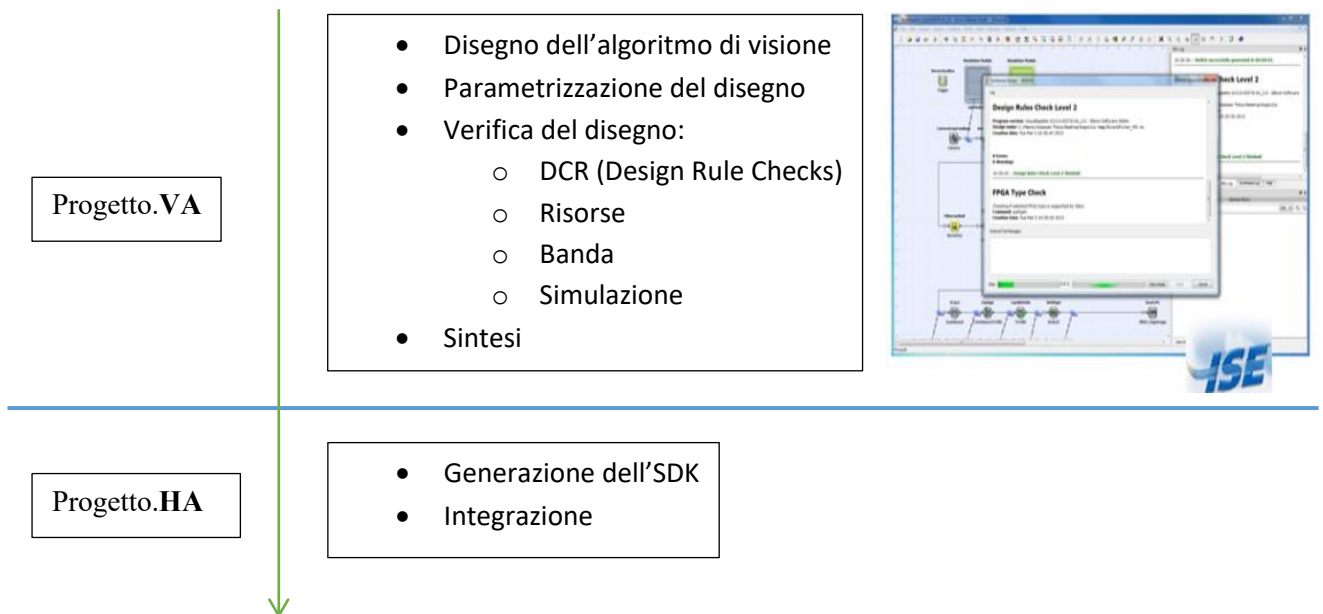


Figura 9. Diagramma a blocchi del flusso di lavoro.

4.1 Mappatura hardware dell'algoritmo di visione

Poiché la prima parte dell'elaborazione è identica sulle due immagini stereo, l'applet è stata strutturata in due processi identici. La Figura 11 mostra come l'algoritmo di visione è stato mappato in hardware per uno dei processi.

Il disegno è stato realizzato tenendo conto che le telecamere da utilizzare per il SDVA sono le GO-5000-PGE, della casa costruttrice JAI (Figura 10). Tali telecamere hanno una risoluzione di 5MPixel (2560 x 2048), ma verranno configurate per operare con una ROI 1920x1080, che garantisce sia adeguata risoluzione, che un throughput di 56 fps (quindi superiore a quello di 50 fps fissato come target). Nello schematico è stato quindi necessario dimensionare il link di acquisizione tenendo conto delle dimensioni del sensore della telecamera ed introdurre un blocco per la selezione della ROI.



Figura 10. Telecamera JAI GO-5000-PGE.

L'applet disegnata effettua la sogliatura dell'immagine e l'analisi dei blob e trasferisce al PC host non solo il risultato di tale analisi (blocco DMA_Center_Of_Gravity), ma anche l'immagine originale (blocco DMA) e quella sogliata (DMA_Thr), in modo tale da rendere possibile il debug una volta che l'applet sarà caricata sull'FPGA.

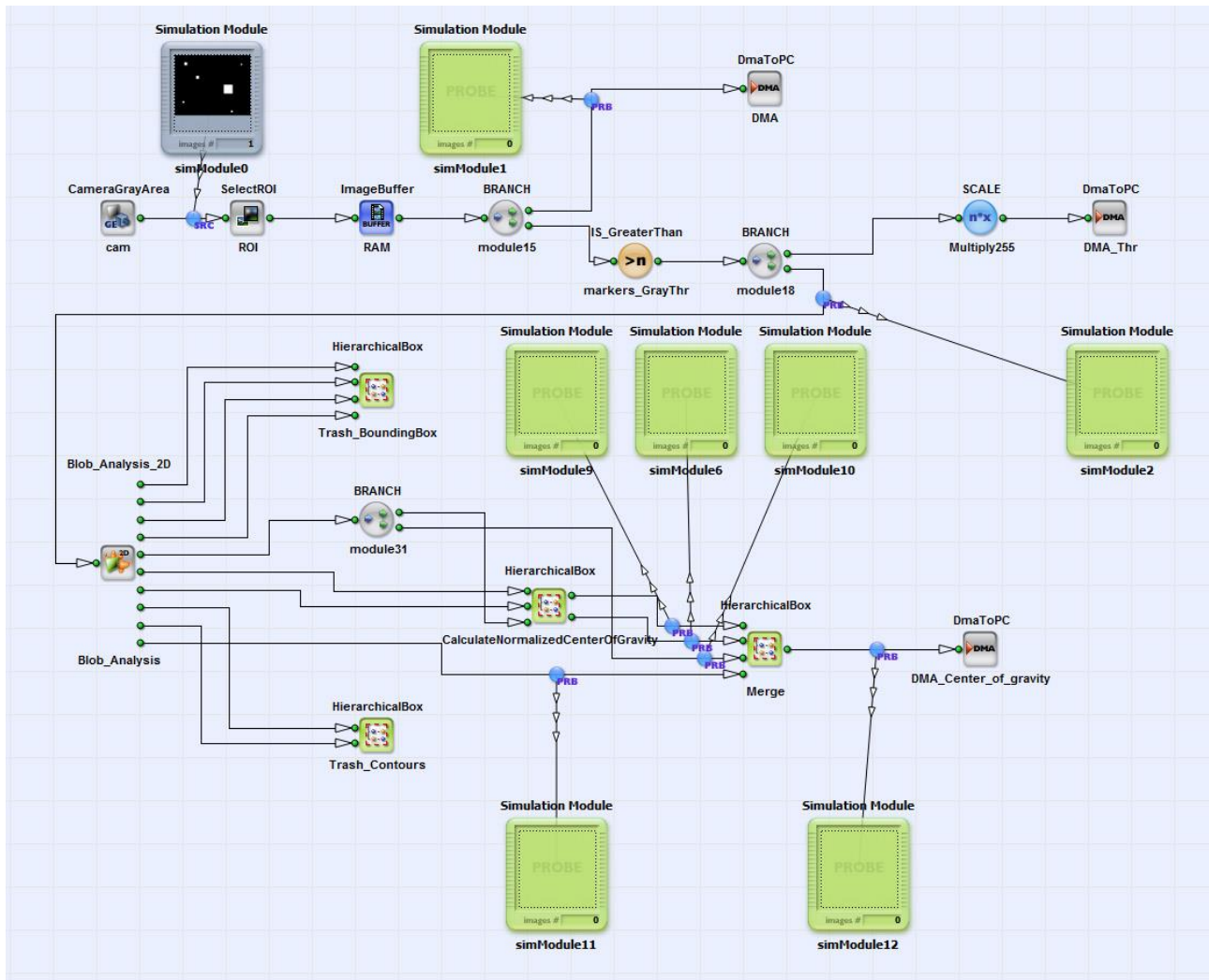


Figura 11. Progetto Visual Applet dell'algoritmo di visione relativo al Processo 0.

L'analisi dei blobs restituisce le coordinate dei centri di massa e le aree degli oggetti trovati nell'immagine binaria. Tali risultati vengono combinati insieme in un unico link di 32 bit con un parallelismo di 3bit. La Figura 12 mostra come i primi 32bit contengono l'informazione del centro di massa, dove la coordinata x è mappata nei 16 bit più bassi e quella y sui 16 bit più alti, i secondi 23 bit trasportano l'informazione relativa all'area dei blob e gli ultimi 32 sono relativi ai flags di errore del processo di analisi.

Per rendere l'applicazione configurabile a run time i parametri del modulo di threshold e del modulo di selezione della ROI sono stati impostati come dinamici.

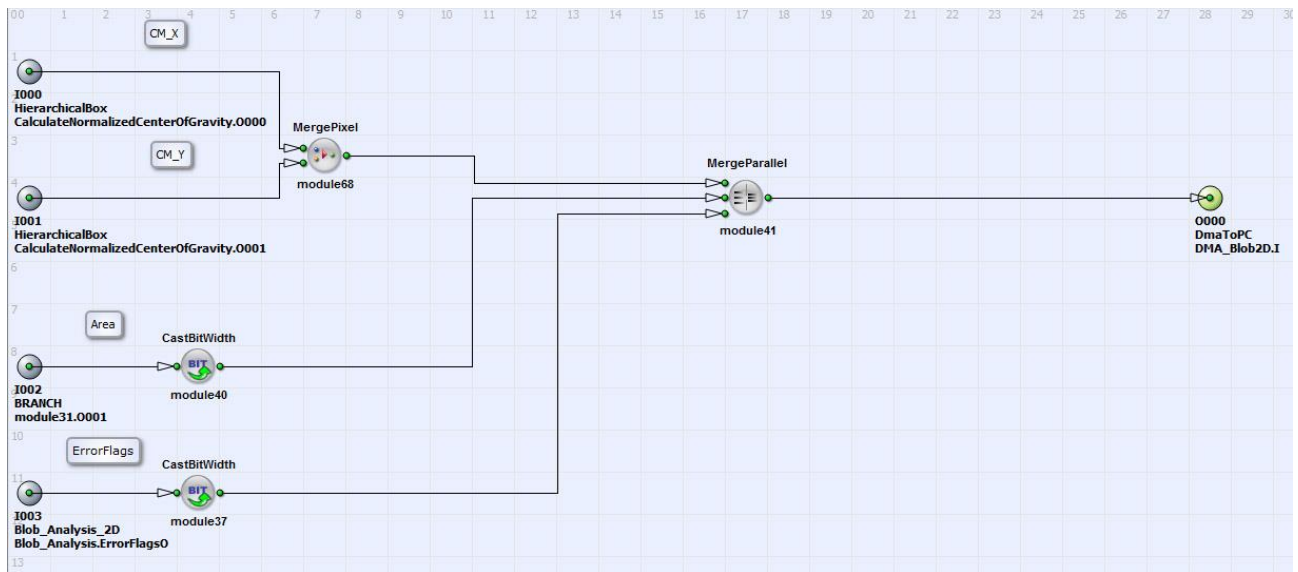


Figura 12. Disegno relativo al blocco Merge del Processo 0.

La Tabella 4 mostra come le risorse dell’FPGA a bordo del frame grabber sono sufficienti per l’implementazione dell’algoritmo di visione. In Figura 13 è possibile vedere in dettaglio le risorse utilizzate per ogni modulo del disegno. Si può osservare come i moduli più onerosi, dal punto di vista delle risorse, siano i blocchi della divisione utilizzati per normalizzare le coordinate del centro di massa. Tali blocchi però sono indispensabili per il corretto funzionamento dell’algoritmo e quindi non possono essere eliminati o sostituiti in nessun modo.

Risorse	Contatore	Livello di riempimento
Blocchi RAM	42	~ 43%
LookupTables	39975	~ 72%
Flip Flops	36222	~ 65%

Tabella 4. Tabella sintetica dell’utilizzo delle risorse dell’FPGA

Module	Type	LUT	RAM LUT	Flip Flops	Block RAM	Embedded ALU
Applet	microEnable IV VQ4-GE	39975	10546	36222	42	0
Environment/Infrastructure		27249	9718	21302	26	0
Process0	Process	6363	414	7460	8	0
Blob_Analysis	Blob_Analysis_2D	2515	142	2174	7	0
CalculateNormaliz...	HierarchicalBox	2162	0	2838	0	0
module32	BRANCH	0	0	0	0	0
module33	DIV	1146	0	1495	0	0
module34	DIV	1016	0	1343	0	0
module35	Trash	0	0	0	0	0
module36	Trash	0	0	0	0	0
module64	CastBitWidth	0	0	0	0	0
module65	CastBitWidth	0	0	0	0	0
cam	CameraGrayArea	168	272	265	0	0
DMA	DmaToPC	113	0	337	0	0
DMA_Blob2D	DmaToPC	199	0	397	0	0
DMA_Thr	DmaToPC	113	0	337	0	0
markers_GrayThr	IS_GreaterThan	36	0	0	0	0
Merge	HierarchicalBox	0	0	0	0	0
module37	CastBitWidth	0	0	0	0	0
module40	CastBitWidth	0	0	0	0	0
module41	MergeParallel	0	0	0	0	0
module68	MergePixel	0	0	0	0	0
module15	BRANCH	0	0	0	0	0
module18	BRANCH	0	0	0	0	0
module31	BRANCH	0	0	0	0	0
Multiply255	SCALE	32	0	0	0	0
RAM	ImageBuffer	822	0	666	1	0
ROI	SelectROI	202	0	299	0	0
Trash_BoundingBox	HierarchicalBox	0	0	0	0	0
Trash_X0	Trash	0	0	0	0	0
Trash_X1	Trash	0	0	0	0	0
Trash_Y0	Trash	0	0	0	0	0
Trash_Y1	Trash	0	0	0	0	0
Trash_Contours	HierarchicalBox	0	0	0	0	0
Trash_Contour...	Trash	0	0	0	0	0
Trash_Contour...	Trash	0	0	0	0	0
Process1	Process	6363	414	7460	8	0

Figura 13. Risorse FPGA utilizzate dall'algoritmo di visione.

4.2 Test e debug del modulo FPGA embedded

È stata effettuata un'analisi funzionale per testare il comportamento dell'implementazione. Si è utilizzata, come ingresso, l'immagine binaria di Figura 14 e mediante la sonda posta sul link del DMA_Center_og_gravity (Figura 11) si è verificata la correttezza dei risultati (Figura 15).

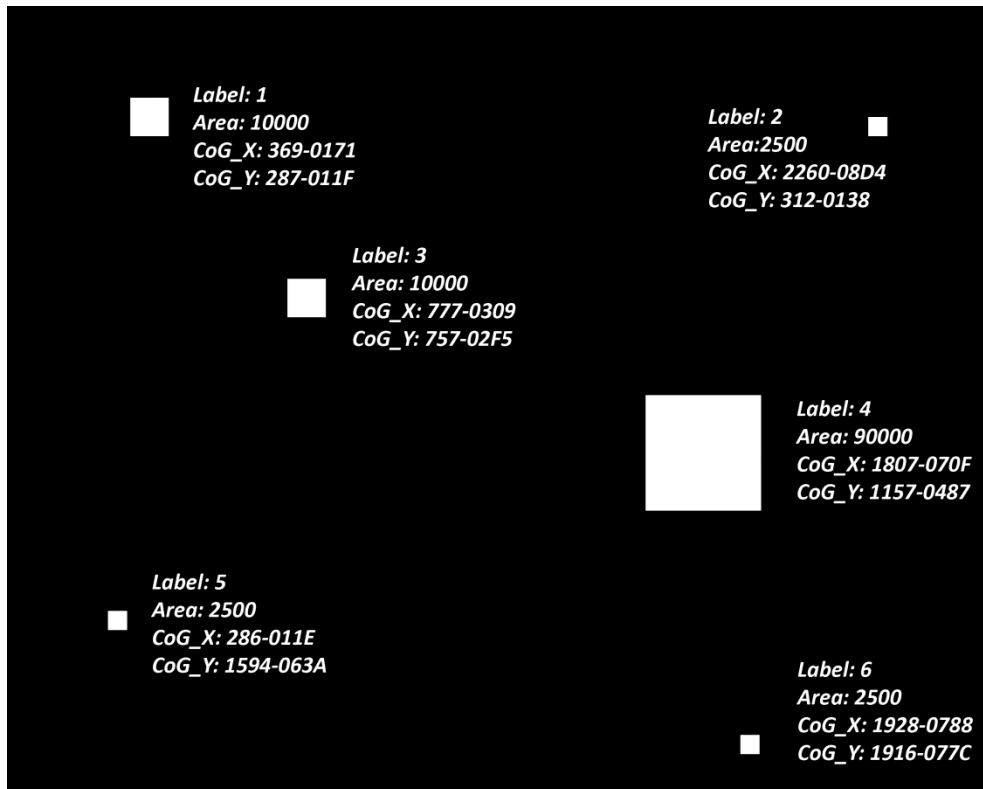


Figura 14. Immagine binaria utilizzata per il test funzionale dell'applet.

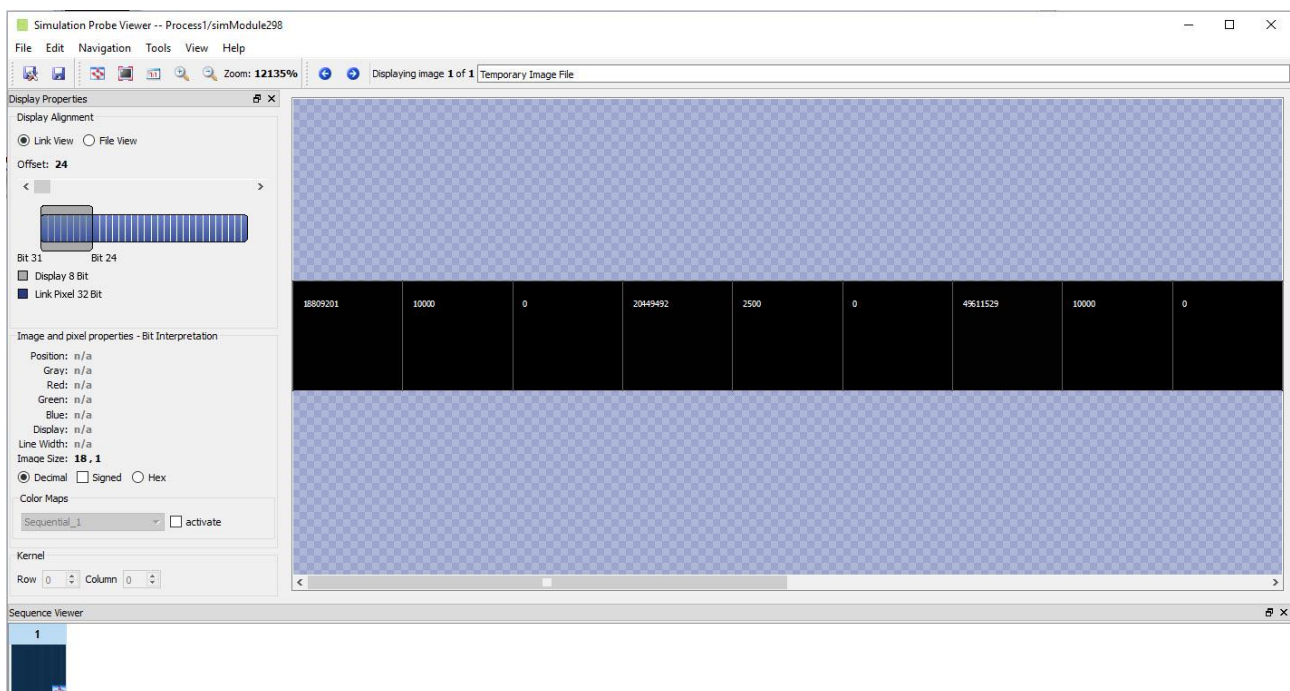


Figura 15. Risultati dell'analisi dei blob. È possibile leggere il centro di massa, l'area e il Byte relativo agli errori dell'analisi, delle prime tre regioni dell'immagine in Figura 14.

Tenendo conto che il SDVA dovrà acquisire una ROI di (1920x1080) ad un frame rate di 50 fps e che la frequenza di lavoro dell'FPGA è fissa e pari a 62.5 MHz, si è effettuata l'analisi di banda si è visto che nell'applicazione creata non ci sono colli di bottiglia (Figura 16).

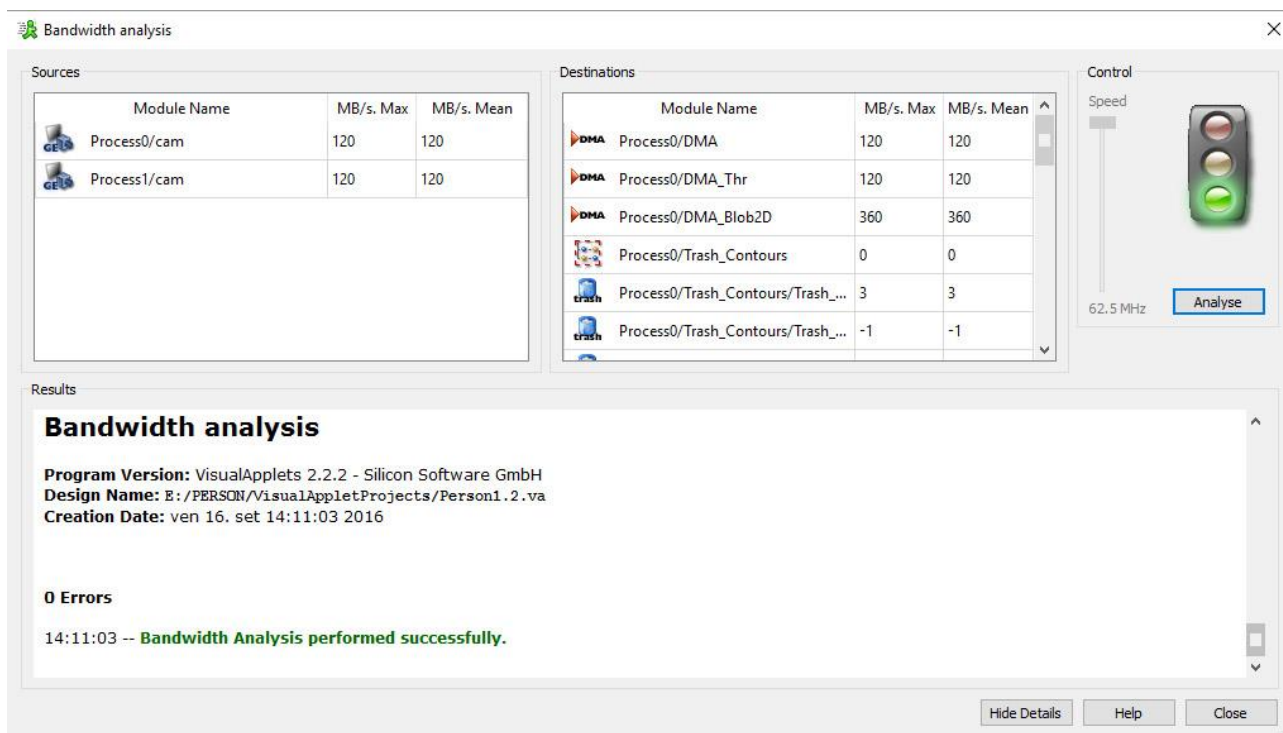


Figura 16 Risultato dell'analisi di banda dell'Applet Person1.3 ottenuto impostando come ROI la dimensione massima del frame 2560*2048

4.3 Sintesi e download

Dallo stesso ambiente di Visual Applet è stato lanciato il processo di sintesi. Tale processo ha generato il file hardware binario (HAP) solo dopo aver verificato che il placement e il routing rispettassero i vincoli temporali.

Per abilitare l'algoritmo di visione sul frame grabber è necessario caricare l'applet generata sulla scheda microEnable IV VQ4GE mediante il tool MicroDisplay della Silicon Software.

5. Integrazione del modulo implementato con il set up di visione artificiale

Per integrare il modulo implementato nel set up di visione artificiale, è stato necessario sincronizzare le due telecamere ed implementare un interfaccia software per l'applet.

5.1 Sincronizzazione delle telecamere

La sincronizzazione delle telecamere è indispensabile per una corretta ricostruzione 3D a partire da un sistema di telecamere stereo.

A tal fine si è scelto di utilizzare una telecamera come master, che acquisisce in modalità free running, e l'altra come slave, che utilizza come trigger esterno per l'acquisizione, lo stesso trigger della telecamera master.

Tenendo conto dell'interfaccia ottica della telecamera JAI GO-5000-PGE (Figura 17, Figura 18), dell'assegnamento dei pin del connettore Hirose 6 (Figura 19) e dei controlli di acquisizione, è stato realizzato un circuito per la sincronizzazione della coppia di telecamere stereo. Di seguito viene illustrato sia il modulo hardware realizzato, che le impostazioni software da applicare alle due telecamere per la corretta sincronizzazione (15).

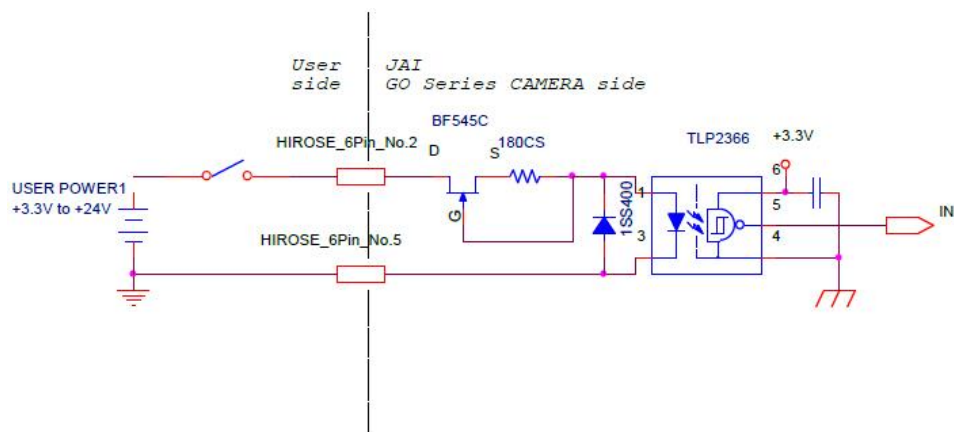


Figura 17 Diagramma della parte esterna del circuito di ingresso

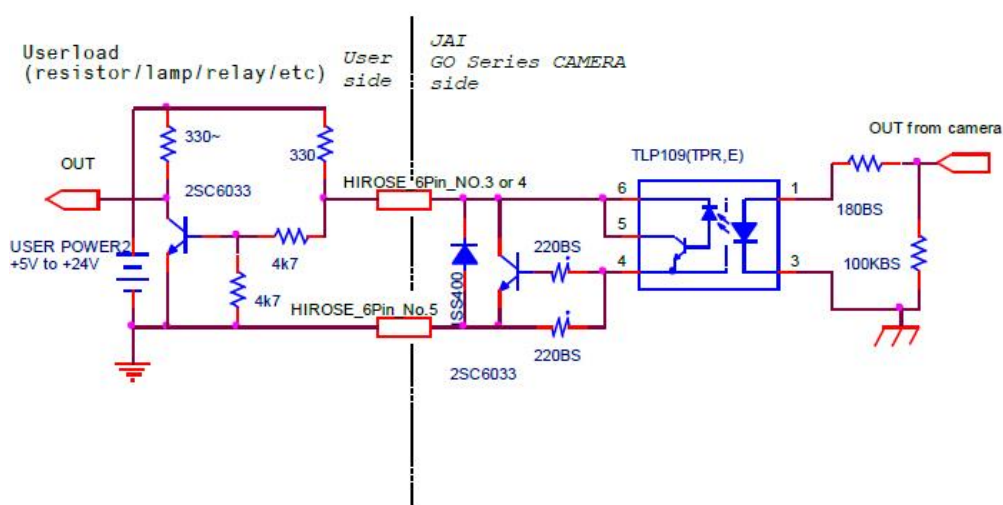
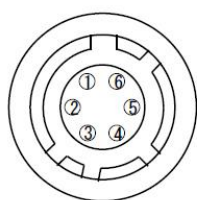


Figura 18 Diagramma della parte esterna del circuito di uscita



Pin no.	I/O	Signal	Remarks
1		DC in	+12V ~ +24V
2	I	Opto in1	GPIO 5
3	O	Opto out1	GPIO 1
4	O	Opto out2	GPIO 2
5		Opto Common	
6		GND	

Figura 19 Vista dall'esterno del connettore Hirose a 6 pin e tabella dell'assegnamento dei pin

5.1.1 Modulo Hardware

In Figura 20 è mostrato lo schematico del circuito realizzato per l'alimentazione e la sincronizzazione della coppia di telecamere stereo. Il trigger della telecamera master (segnale Exposure Active) è messo in output, sui pin 3, ed inviato in input alla slave, sui pin 2.

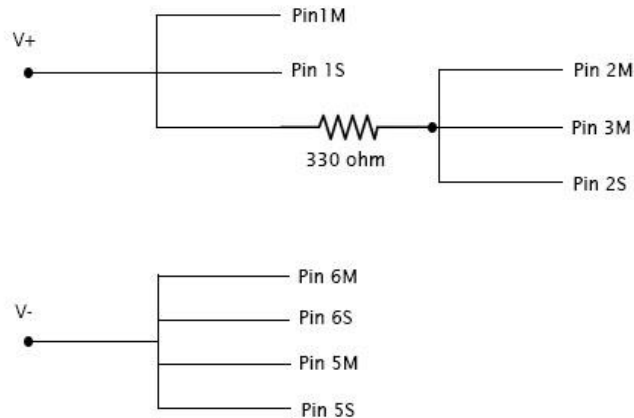


Figura 20 Circuito per la sincronizzazione delle telecamere

5.1.2 Impostazioni software

La Tabella 5 e la Tabella 6, mostrano come configurare la coppia di telecamere stereo per la corretta sincronizzazione.

MASTER		
Acquisition Controll	Trigger Mode(Trigger Selector)	Off
	Exposure Mode	Timed
	Exposure Time	et
	Frame Rate	fps
Digital IO Controll	Line Selector	Line 1 Opt 1 Out
	Line Source	Exposure Active

Tabella 5 Configurazione della telecamera master

SLAVE		
Acquisition Controll	Trigger Mode(Trigger Selector)	On
	Exposure Mode	Timed
	Exposure Time	et
	Trigger Source(Trigger Selector)	Line 5 OPT IN 1

Tabella 6 Configurazione della telecamera slave

5.1.3 Valutazioni sul modulo di sincronizzazione implementato

Il Grafico 1 mostra la differenza tra l'istante di acquisizione della telecamera master e quello della telecamera slave, in un ciclo di acquisizione di 1 minuto e 47 secondi. Tale differenza risulta essere in media di 0,01ms con un deviazione standard di 0,02ms. Dato che il periodo di acquisizione è di 20ms, si può considerare che i frames delle due telecamere siano stati sincronizzati correttamente.

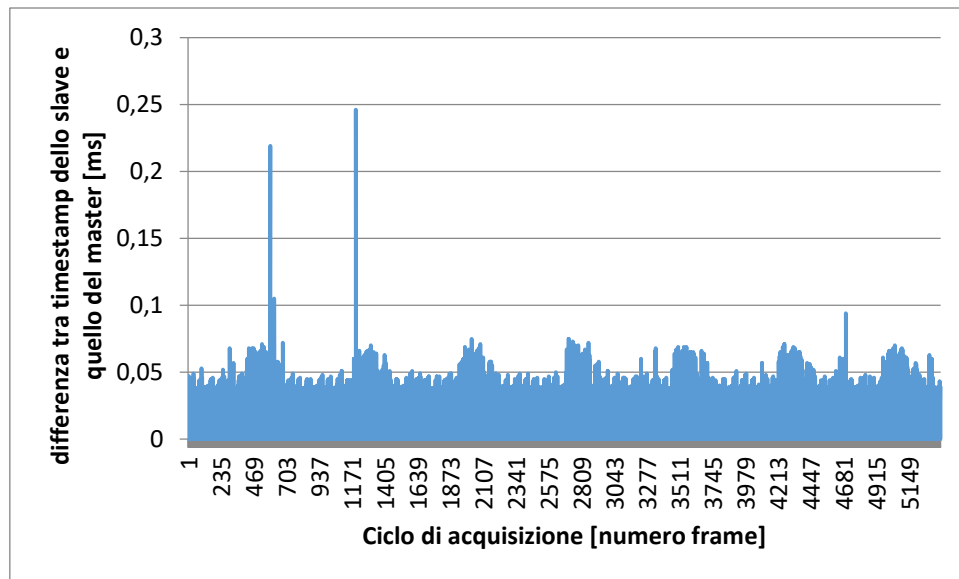


Grafico 1 Differenza tra l'istante di acquisizione della telecamera master e della slave

5.2 Interfaccia Software

La Figura 21 descrive l'architettura del sistema di visione artificiale.

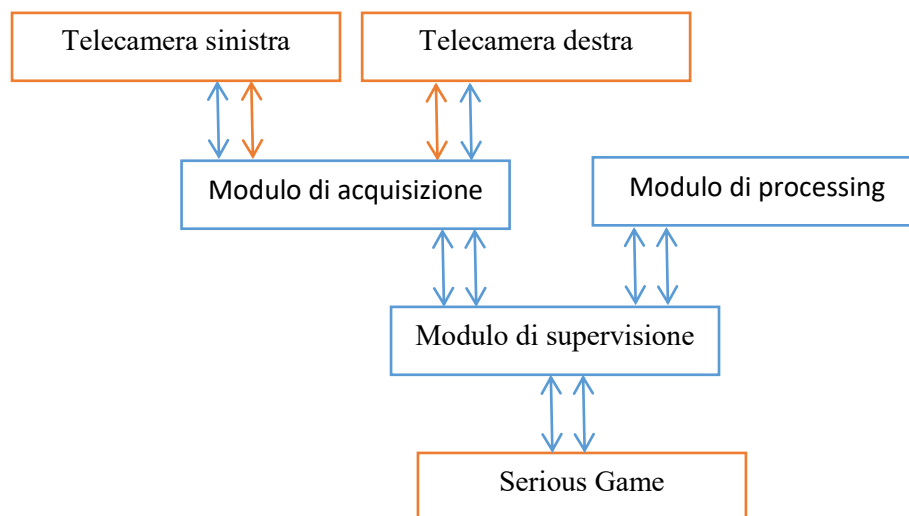


Figura 21 Componenti del Sistema Di Visione Artificiale

Per integrare l'applet Person1.3, è stato necessario sviluppare un nuovo modulo di acquisizione in grado di interfacciarsi con il frame grabber, invece che direttamente con la coppia di telecamere stereo, e con il modulo di supervisione. Anche il modulo di processing ha subito delle modifiche in quanto, invece di ricevere, come input, i frames acquisiti, riceverà direttamente le coordinate dei centri di massa dei markers.

5.2.1 Modulo di acquisizione

Il modulo di acquisizione implementato, è costituito dalle classi C++:

- GbEStereo
- Blob2D

5.2.1.1 Classe GbEStereo

La classe GbEStereo costituisce il cuore del modulo di acquisizione, in quanto si interfaccia sia con il frame grabber, mediante la funzioni della libreria Runtime della Silicon Software, che con il modulo di supervisione.

La Figura 22 descrive la struttura della classe GbEStereo.

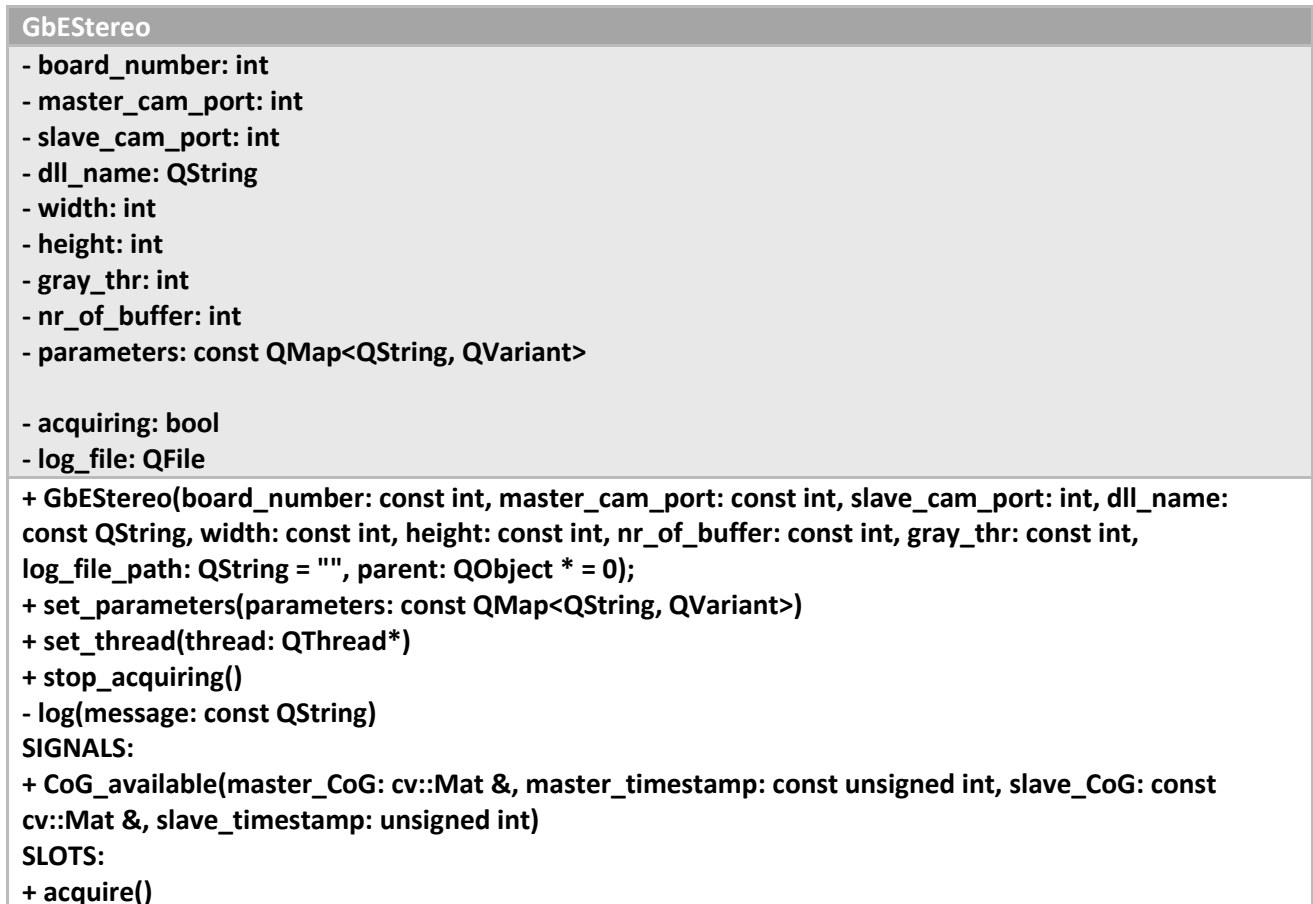


Figura 22 Diagramma delle classi di GbEStereo

Per poter utilizzare il modulo di acquisizione è necessario:

- Instanziare e inizializzare un oggetto della classe GbEStereo
- Impostare i parametri di acquisizione
- Associare un thread all'oggetto istanziato
- Avviare il thread, che a sua volta avvierà l'acquisizione
- Interrompere l'acquisizione mediante il metodo stop_acquiring()

L'inizializzazione dell'oggetto della classe GbEStereo avviene passando al costruttore i seguenti parametri:

- board_number: indice del frame grabber da cui effettuare l'acquisizione
- master_cam_port: indice della porta GigE della telecamera master
- slave_cam_port: indice della porta GigE della telecamera slave
- dll_name: nome dell'applet da caricare sul frame grabber ("Person1.3")
- width: larghezza del frame da acquisire
- height: altezza del frame da acquisire
- gray_thr: livello di grigio con cui sogliare l'immagine
- nr_of_buffer: dimensione del buffer circolare di immagini

Gli unici parametri dell'acquisizione che sarà necessario impostare sono:

- il frame rate
- il tempo di esposizione

Questi parametri vanno invece impostati con il metodo `set_parameters(const QMap<QString, QVariant> parameters)`.

Ogni volta che una coppia di immagini stereo viene acquisita e le coordinate dei markers vengono calcolati, l'oggetto della classe `GbEStereo` emette il segnale `CoG_available`, passando come parametri:

- un reference alla matrice delle coordinate dei markers della telecamera master (`master_CoG`)
- il time stamp, in microsecondi, corrispondente al frame acquisito della telecamera master (`master_timestamp`)
- un reference alla matrice delle coordinate dei markers della telecamera slave (`slave_CoG`)
- il time stamp, in microsecondi, corrispondente al frame acquisito della telecamera slave (`slave_timestamp`)

Di seguito viene mostrato il codice di esempio per l'utilizzo del modulo di acquisizione:

```

//! Instanzia e inizializza un oggetto della classe GbEStereo
GbEStereo gbe_stereo( 0, PORT_A, PORT_B, "Person1_3.hap", 1920, 1080, 16, 80,
"log_gbestereo.txt" );
//! Instanzia un thread da associare all'oggetto gbe_stereo
QThread gbe_stereo_thread;
//! Imposta i parametri di acquisizione
QMap<QString, QVariant> parameters;
parameters["AcquisitionFrameRate"] = 56;
parameters["ExposureTime"] = 1 * 1e+3;
gbe_stereo.set_parameters( parameters );

//! Associa il thread all'oggetto GbEStereo
gbe_stereo.set_thread( &gbe_stereo_thread );
gbe_stereo.moveToThread( &gbe_stereo_thread );

//! Avvia l'acquisizione
gbe_stereo_thread.start( QThread::TimeCriticalPriority );

//! Interrompe l'acquisizione
gbe_stereo.stop_acquiring();

```

5.2.1.2 Classe *Blob2D*

La classe `Blob2D`, è invece utilizzata dalla classe `GbEStereo`, per la lettura dei risultati dell'analisi dei blob.

La struttura della classe è mostrata in Figura 23.

Blob2D
- <code>m_noOfObjects</code>: unsigned int - <code>m_Features</code>: BlobStruct* - <code>CoG_vect</code>: cv::Mat
+ <code>ReadRawData(p_blobData: unsigned char*, p_noOfObjects: unsigned int): int</code> + <code>GetNumberOfObjects(): unsigned int</code> + <code>GetArea(objectIndex: unsigned int): unsigned int</code> + <code>GetCoGX(objectIndex: unsigned int): double</code> + <code>GetCoGY(objectIndex: unsigned int): double</code> + <code>GetErrorflags(objectIndex: unsigned int): unsigned int</code> + <code>get_CoG_vect():cv::Mat</code>

Figura 23 Diagramma delle classi di `Blob2D`

Gli attributi privati sono i seguenti:

- `m_noOfObjects`: numero delle regioni trovate nell'immagine
- `m_Features`: vettore di elementi `BlobStruct`, che memorizzano i risultati dell'analisi dei blob
- `CoG_vect`: matrice $1 \times N$, contenente i centri di massa di tutte le regioni dell'immagine

L'elemento `BlobStruct` contenete i risultati dell'analisi dei blob relativi ad una singola regione, ed in particolare:

- L'area (Area)
- La coordinata x del centro di massa (CoGX)
- La coordinata y del centro di massa (CoGY)
- Il byte (Errorflags) relativo ai flags di errore, che indicano se si è verificato un overflow del numero di blob¹

La lettura dei risultati della blob analisi viene effettuata mediante il metodo `ReadRowData`, al quale è necessario passare:

- Il puntatore al buffer di memoria dei risultati della blob analisi (`p_blobData`)
- Il numero di regioni che sono state trovate nell'immagine (`p_noOfObjects`)

La classe `Blob2D`, mette inoltre a disposizione una serie di funzioni in grado di restituire, per ciascuna regione, l'area e il centro di massa.

Il metodo `get_CoG_vect()` restituisce invece una matrice $1 \times N$, dove N è pari al numero di regioni trovate nell'immagine. Ciascun elemento della matrice è un vettore di due double, contenente la coordinata x e la coordinata y del centro di massa.

Di seguito viene mostrato un frammento di codice della classe `GbEStereo` in cui vengono utilizzati due oggetti della classe `Blob2D`:

```
#!/ Recupera i risultati della blob analisi della telecamera master
Blob2D blob_master;
unsigned char* iPtr_CoG_master = (unsigned char*) Fg_getImagePtrEx(fg,
last_CoG_nr_master, 1, p_mem_CoG_master);
int DMAlength_CoG_master = last_CoG_nr_master;
Fg_getParameter (fg, FG_TRANSFER_LEN, &DMAlength_CoG_master, 1);
#!/ Legge i risultati della blob analisi dal buffer
blob_master.ReadRawData(iPtr_CoG_master, DMAlength_CoG_master / 12); //12
Bytes per Object

#!/ Recupera i risultati della blob analisi della telecamera slave
Blob2D blob_slave;
unsigned char* iPtr_CoG_slave = (unsigned char*) Fg_getImagePtrEx(fg,
last_CoG_nr_slave, 3, p_mem_CoG_slave);
int DMAlength_CoG_slave = last_CoG_nr_slave;
g_getParameter (fg, FG_TRANSFER_LEN, &DMAlength_CoG_slave, 3);
#!/ Legge i risultati della blob analisi dal buffer
blob_slave.ReadRawData(iPtr_CoG_slave, DMAlength_CoG_slave / 12); //12 Bytes
per Object
```

5.2.2 Modulo di processing

¹ 128 è il numero massimo di blobs impostato nell'Applet Person 1.3

In questa nuova struttura del SDVA, il modulo di processing è stato notevolmente snellito, in quanto una tra le operazioni più onerose dal punto di vista computazionale, il calcolo dei centri di massa delle regioni, è stato spostato sulla piattaforma FPGA.

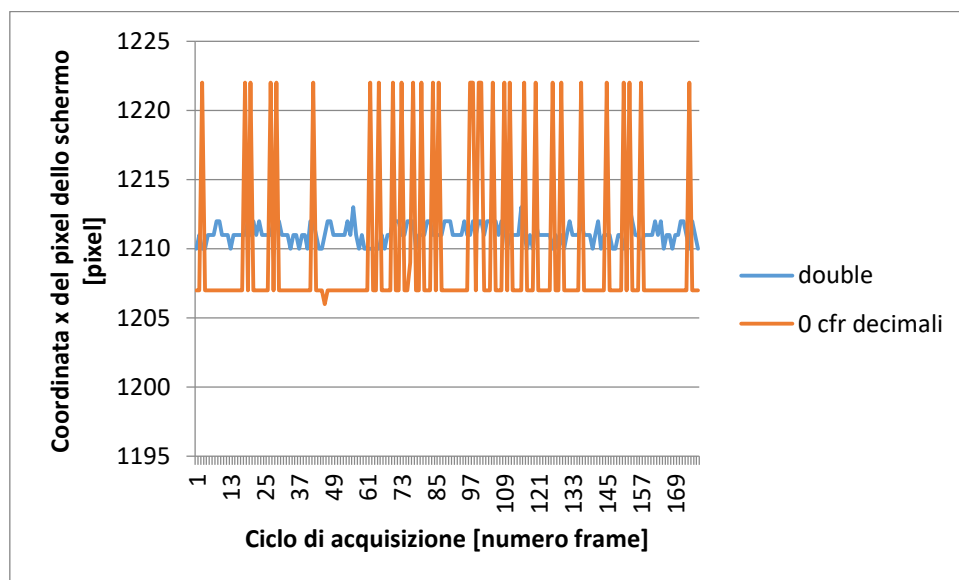
Il modulo di processing, invece di processare le immagini acquisite, si ritrova a dover processare solo le coordinate dei markers della scena. Questo implica un'ulteriore velocizzazione del sistema, in quanto l'operazione di remapping sulle coordinate dei 16 markers (8 markers per ciascun²), è molto meno onerosa rispetto al remapping dei frames sinistro e destro, ciascuno costituito da 2 MPixels³.

6 Test e debug del sistema integrato

Il sistema integrato di parte hardware ed interfaccia software è stato testato e ha rivelato dei problemi di instabilità del pixel selezionato, nella fase di puntamento dello schermo. Tale problema è stato risolto andando ad apportare una modifica al progetto della parte hardware del SDVA. La nuova versione del sistema è stata testata e sono state valutate le prestazioni dal punto di vista computazionale.

6.1 Problema dell'instabilità del puntamento

Indicando un punto fisso sullo schermo, per un intervallo di tempo, il punto del monitor che il SDVA andava a selezionare non era costante nell'intero periodo, ma oscillava attorno ad un punto medio fisso. Si è ipotizzato che tale problema fosse legato al fatto che il calcolo dei centri di massa dei markers, effettuato sulla scheda FPGA, restituisse un intero e non un double, come nel caso del SDVA sviluppato per lavorare esclusivamente con la CPU e la GPU. Per verificare la validità di tale ipotesi, è stato modificato il modulo di processing del SDVA, pensato per lavorare senza frame grabber, in modo tale che arrotondasse all'intero i valori dei centri di massa dei markers. Si è riscontrato che, con questa modifica, si verificava lo stesso tipo di variabilità del pixel puntato. Il Grafico 2 e il Grafico 3 mostrano i risultati di tale test. Come è possibile osservare, nel caso di troncamento all'intero dei centri di massa, la propagazione di tale errore al calcolo del pixel puntato è notevole, in quanto si hanno delle oscillazioni che arrivano fino a differenze di 16 pixel tra due frames consecutivi, mentre nel caso in cui non viene fatto nessun troncamento si può avere al massimo una variazione di 2-3 pixels.



² Il modello di mano scelto, per il SDVA, prevede 8 markers

³ La ROI di acquisizione dell'SDVA è di 1920x1080

Grafico 2 Grafico della coordinata x del pixel selezionato dal SDVA al variare del tempo, mentre il pixel puntato dall'utente rimane costante per 24 secondi

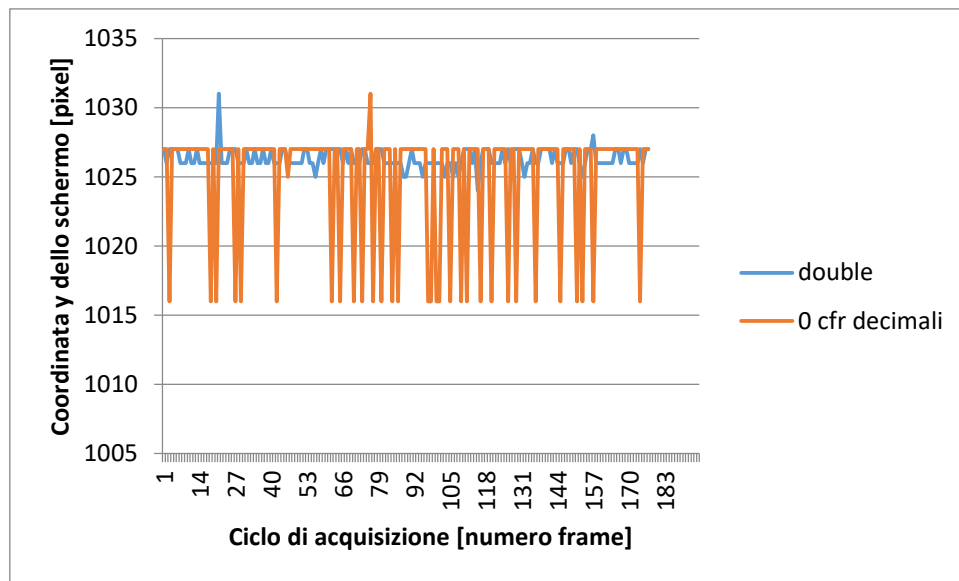


Grafico 3 Grafico della coordinata y del pixel selezionato dal SDVA al variare del tempo, mentre il pixel puntato dall'utente rimane costante per 24 secondi

Per capire il numero di cifre decimali sufficienti ad eliminare il problema dell'instabilità del puntamento, è stato effettuato un ulteriore test, sfruttando sempre il modulo di processing del SDVA pensato per lavorare senza frame grabber. Il Grafico 4e il Grafico 5 mostrano i risultati di tale test.

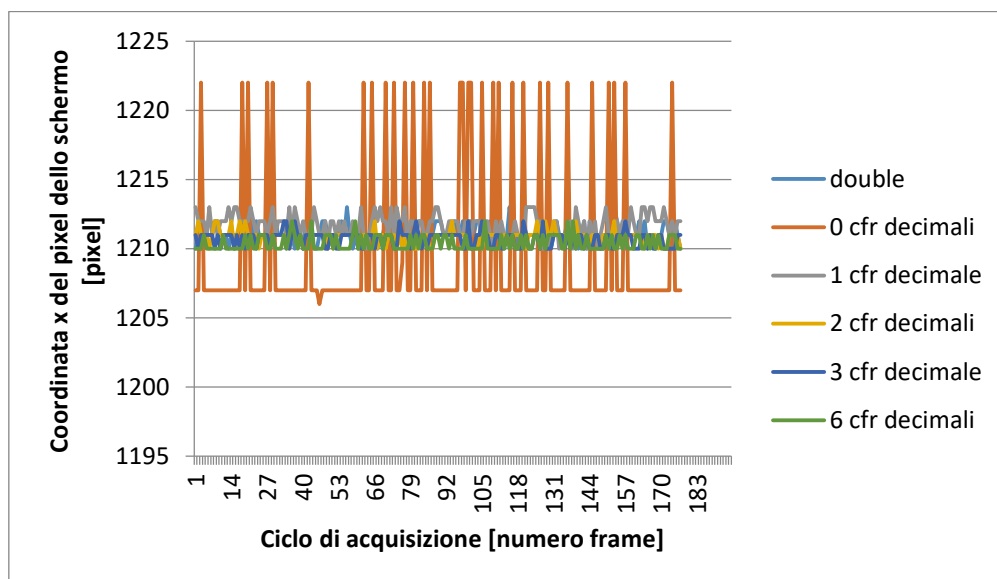


Grafico 4 Grafico della coordinata x del pixel selezionato dal SDVA al variare del tempo, mentre il pixel puntato dall'utente rimane costante per 24 secondi. Relativo al troncamento dei centri di massa dei markers rispettivamente a 0, 1, 2, 3 e 6 cifre decimali

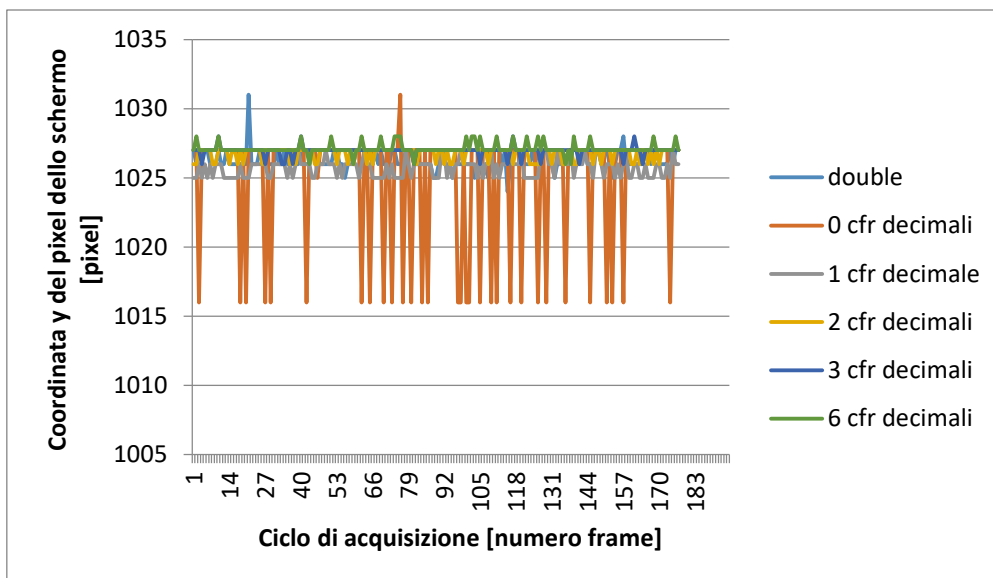


Grafico 5 Grafico della coordinata y del pixel selezionato dal SDVA al variare del tempo, mentre il pixel puntato dall'utente rimane costante per 24 secondi. Relativo al troncamento dei centri di massa dei markers rispettivamente a 0, 1, 2, 3 e 6 cifre decimali

Come è possibile osservare più chiaramente nel Grafico 6 e nel Grafico 7 un troncamento alla prima cifra decimale è sufficiente ad eliminare il problema dell'instabilità del puntamento, dovuta alla propagazione dell'errore.

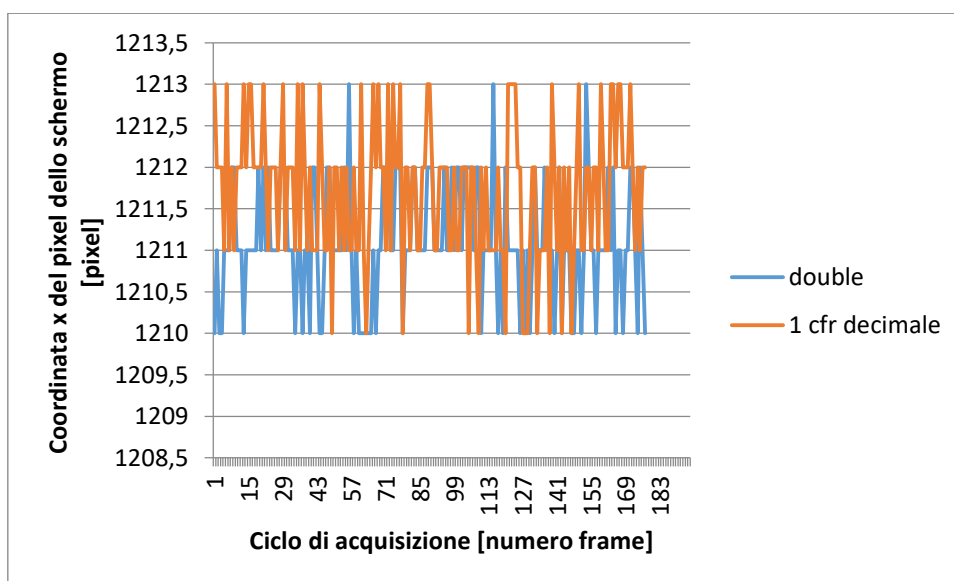


Grafico 6 Grafico della coordinata x del pixel selezionato dal SDVA al variare del tempo, mentre il pixel puntato dall'utente rimane costante per 24 secondi. Relativo al troncamento a 0 cifre decimali dei centri di massa dei markers

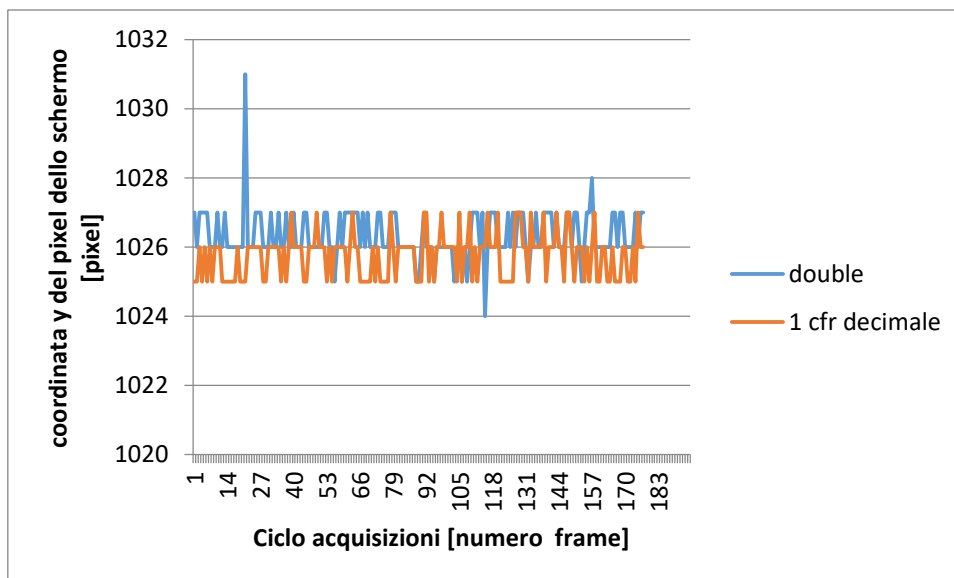


Grafico 7 Grafico della coordinata y del pixel selezionato dal SDVA al variare del tempo, mentre il pixel puntato dall'utente rimane costante per 24 secondi. Relativo al troncamento a 0 cifre decimali dei centri di massa dei markers

6.2 Soluzione all'instabilità del puntamento

Come abbiamo visto, per risolvere il problema dell'instabilità del pixel puntato è sufficiente troncare il centro di massa del pixel alla prima cifra decimale. Ciò semplifica notevolmente le modifiche da apportare al progetto hardware e software, in quanto la struttura del link dei risultati dell'analisi dei blob, rimane invariata. Infatti, tenendo conto che la ROI di acquisizione è 1920x1080 e che vogliamo approssimare alla prima cifra decimale, per la codifica delle coordinate dei centri di massa, sono ancora sufficienti 16 bit.

La Figura 24 mostra il nuovo blocco per la normalizzazione delle coordinate dei centri di massa.

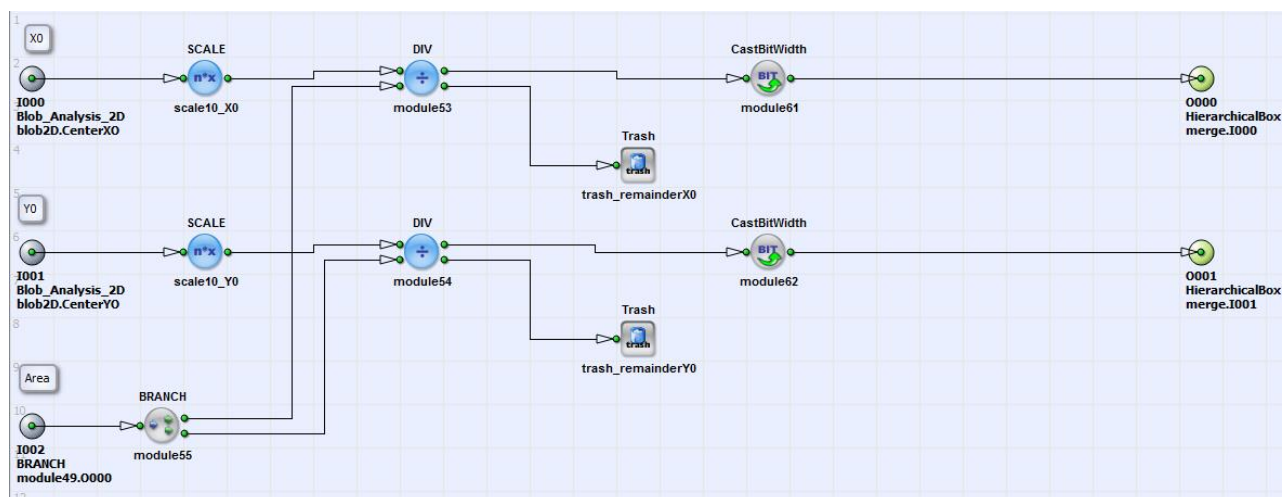


Figura 24 Disegno relativo al blocco `calculate_normalized_CoG`

Questa modifica però, porta ad un aumento delle risorse FPGA necessarie, come è possibile vedere nella Tabella 7. Più si ampliano le risorse richieste più critica diventa la fase di sintesi del progetto, ed in particolare il placement, il routing e il check del rispetto dei vincoli di tempo. Per ridurre tale criticità, e visto che una fase di test è già stata effettuata, si è ridotto il numero di DMA utilizzati dall'applet. Si è preferito lasciare il DMA relativo alle immagini acquisite per garantire un minimo debug, ma si è eliminato il

DMA relativo al risultato della sogliatura del frame acquisito. La Tabella 8 mostra le risorse utilizzate nel caso di questa nuova applet, “Person1.6”, che utilizza solo 4 DMA.

Risorse	Contatore	Livello di riempimento
Blocchi RAM	42	~ 43%
LookupTables	41215	~ 74%
Flip Flops	37666	~ 68%

Tabella 7 Tabella sintetica dell'utilizzo delle risorse dell'FPGA nel progetto che prevede 6 DMA

Risorse	Contatore	Livello di riempimento
Blocchi RAM	34	~ 35%
LookupTables	33544	~ 60%
Flip Flops	31884	~ 57%

Tabella 8 Tabella sintetica dell'utilizzo delle risorse dell'FPGA nel progetto che prevede 4 DMA

In Figura 25 è possibile vedere le schematico del Processo 0 relativo alla nuova applet Person1.6. Il blocco merge, come abbiamo anticipato, è rimasto invariato (Figura 12).

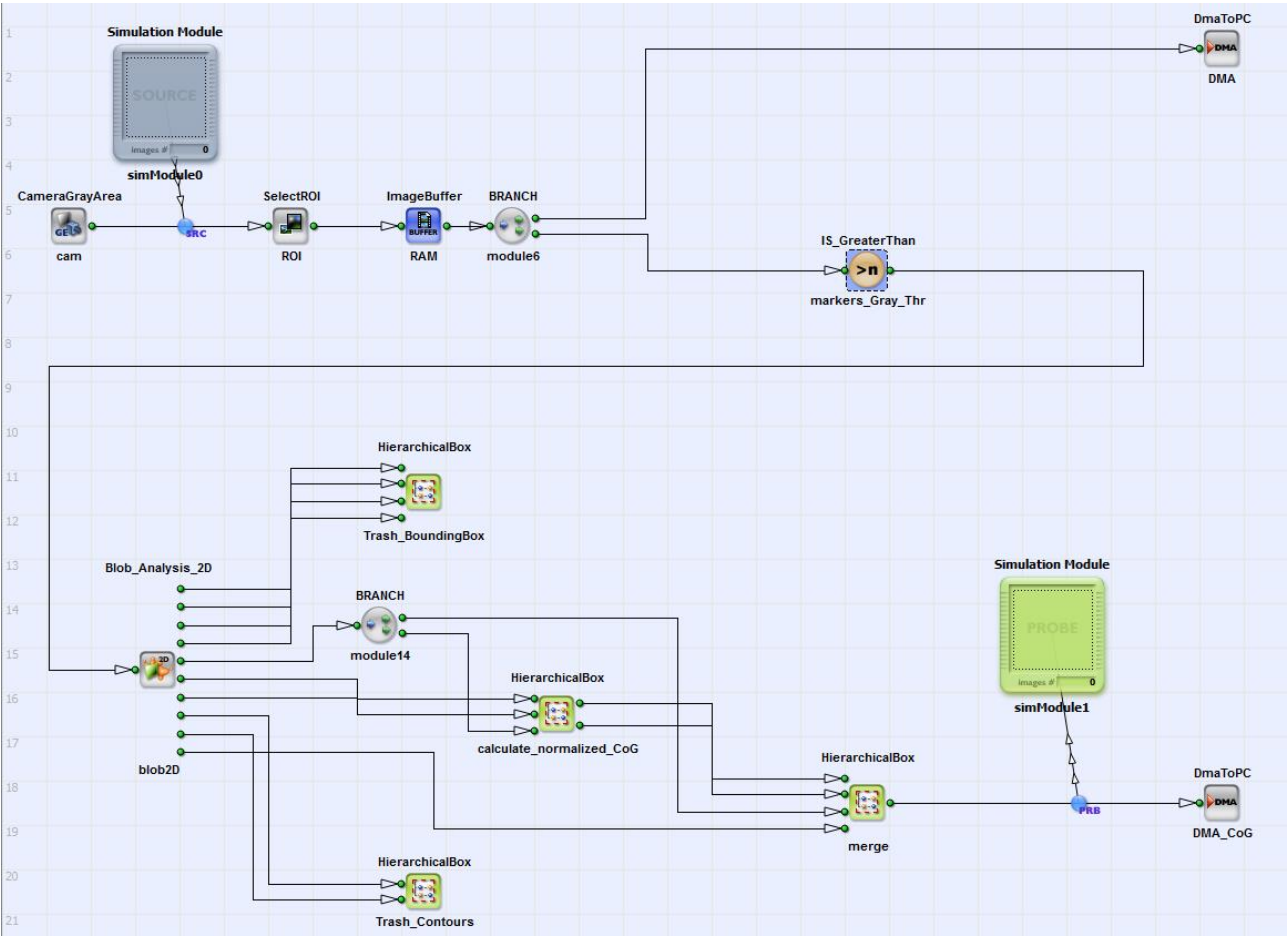


Figura 25 Progetto Visual Applet dell’algoritmo di visione relativo al Processo 0

La Tabella 9 mostra la corrispondenza tra i moduli del progetto e le risorse dell’FPGA, mentre la Figura 26 descrive in dettaglio le risorse FPGA utilizzate da ciascun singolo modulo.

Module Name	Resource Name	Resource Index
Process0/cam	CAM	0
Process0/DMA	DMA	0
Process0/RAM	RAM	0
Process0/DMA_CoG	DMA	1
Process1/DMA	DMA	2
Process1/cam	CAM	1
Process1/RAM	RAM	1
Process1/DMA_CoG	DMA	3

Tabella 9 Tabella degli indici delle risorse utilizzate

Module	Type	LUT	RAM LUT	Flip Flops	Block RAM	Embedded ALU
Applet	microEnable IV VQ4-GE	33544	8214	31884	34	0
Environment/Infrastructure		19874	7386	16386	18	0
Process0	Process	6835	414	7749	8	0
blob2D	Blob_Analysis_2D	2512	142	2169	7	0
calculate_normalized_CoG	HierarchicalBox	2782	0	3478	0	0
module16	DIV	1430	0	1823	0	0
module17	DIV	1284	0	1655	0	0
module18	BRANCH	0	0	0	0	0
module21	CastBitWidth	0	0	0	0	0
module22	CastBitWidth	0	0	0	0	0
scale10_X0	SCALE	35	0	0	0	0
scale10_Y0	SCALE	33	0	0	0	0
trash_resto_X0	Trash	0	0	0	0	0
trash_restoY0	Trash	0	0	0	0	0
cam	CameraGrayArea	168	272	265	0	0
DMA	DmaToPC	113	0	337	0	0
DMA_CoG	DmaToPC	199	0	397	0	0
markers_Gray_Thr	IS_GreaterThan	36	0	0	0	0
merge	HierarchicalBox	0	0	0	0	0
module29	MergeParallel	0	0	0	0	0
module30	MergePixel	0	0	0	0	0
module31	CastBitWidth	0	0	0	0	0
module32	CastBitWidth	0	0	0	0	0
module14	BRANCH	0	0	0	0	0
module6	BRANCH	0	0	0	0	0
RAM	ImageBuffer	822	0	666	1	0
ROI	SelectROI	202	0	299	0	0
Trash_BoundingBox	HierarchicalBox	0	0	0	0	0
trash_X0	Trash	0	0	0	0	0
trash_X1	Trash	0	0	0	0	0
trash_Y0	Trash	0	0	0	0	0
trash_Y1	Trash	0	0	0	0	0
Trash_Contours	HierarchicalBox	0	0	0	0	0
trash_contour_dia	Trash	0	0	0	0	0
trash_contour_ortho	Trash	0	0	0	0	0
Process1	Process	6835	414	7749	8	0

Figura 26 Risorse FPGA utilizzate dall'applet Person1.6

6.3 Valutazione prestazioni

Prima dell'analisi delle performance del sistema SDVA basato sull'FPGA, vengono brevemente sintetizzati i risultati ottenuti con l'utilizzo della GPU.

Come è possibile osservare dalla Figura 27, l'impiego della GPU aveva portato il tempo di elaborazione da 19,01 ms e deviazione standard di 10,05 ms (caso di solo utilizzo della CPU), a 10,99 ms e deviazione standard pari a 0,89 ms (per la Tesla K40).

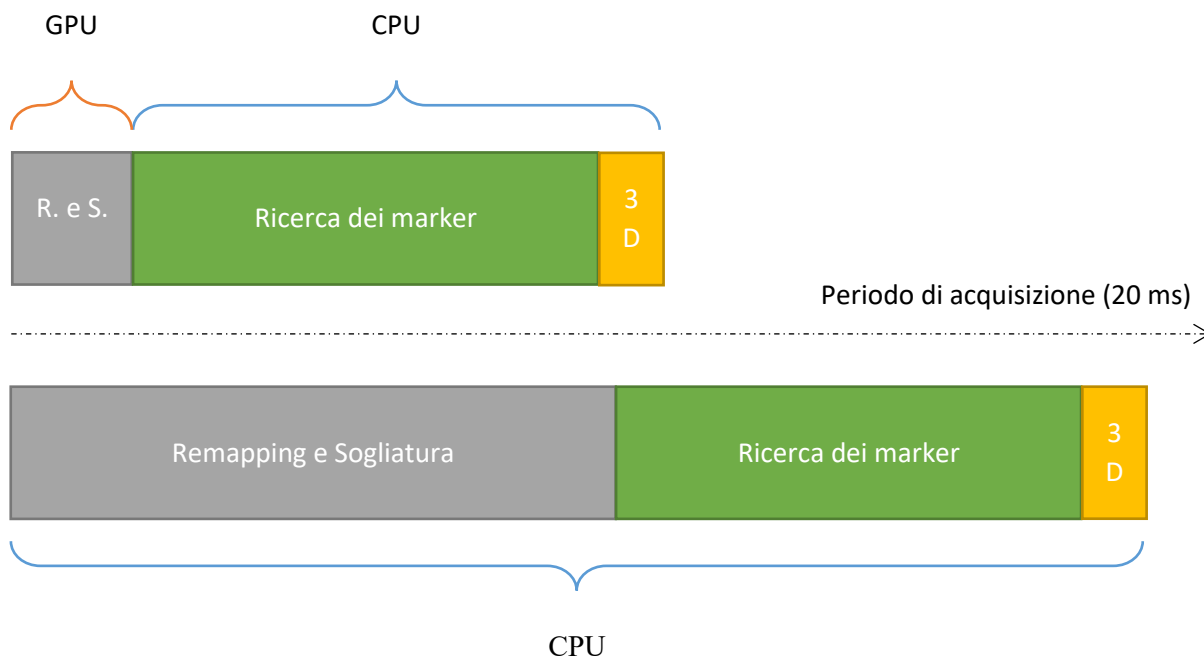


Figura 27 Confronto dei tempi di elaborazione

Con l'introduzione dell'FPGA, l'operazione di remapping dell'intera immagine viene eliminata, in quanto il remapping verrà effettuato solo sulle coordinate 2D dei markers, e la ricerca dei markers è demandata alla scheda hardware. In questo modo il carico della CPU è notevolmente ridotto, in quanto corrisponderà al solo blocco giallo della Figura 27.

Il Grafico 8 mostra i tempi di elaborazione della CPU in un ciclo di acquisizione di un minuto e 57 secondi. In questo modo il tempo di elaborazione della CPU è stato ridotto a 0,003ms con una deviazione standard di 0,129ms.

Tenendo conto che, se l'elaborazione di una coppia di immagini richiede più tempo del periodo di acquisizione il supervisore evita di elaborare la coppia appena acquisita, e che il carico computazionale della CPU si è drasticamente ridotto, ci si aspetta che la percentuale delle coppie di immagini non elaborate sia ridotta rispetto al caso della GPU. Infatti come è possibile notare dal Grafico 8, vengono elaborate tutte le coppie di immagini, mentre con la GPU la percentuale delle coppie non elaborate era di 0,55% di quelle acquisite. La riduzione del carico computazionale a livello della CPU, è vantaggiosa anche perché consente di sfruttare, per altri compiti, le risorse che prima erano impiegate dal modulo di processing.

Il tempo che la scheda FPGA impiega per effettuare la ricerca dei markers, non è possibile quantificarlo. Si può solo verificare che non vengano persi frames. Il Grafico 9 mostra come tra ogni frame e quello precedente trascorra sempre un intervallo di tempo di 20 ms, e che quindi nessun frame venga perso. Un test da poter effettuare è quello di aumentare il frame rate dell'acquisizione per vedere quando il frame grabber comincia a perdere frames, ma dato che la ROI di acquisizione deve essere di 1920x1080, il

massimo frame rate a cui si può arrivare è di 56fps. Il Grafico 10 mostra come anche nel caso di un frame rate di 56 fps non viene persa nessuna coppia di frames. Da queste osservazioni si può dedurre che il blocco dell’analisi dei blob non rallenta l’applicazione, ed è per questo che non è stato introdotto il blocco dei pre-processing morfologico nell’applet (Figura 8).

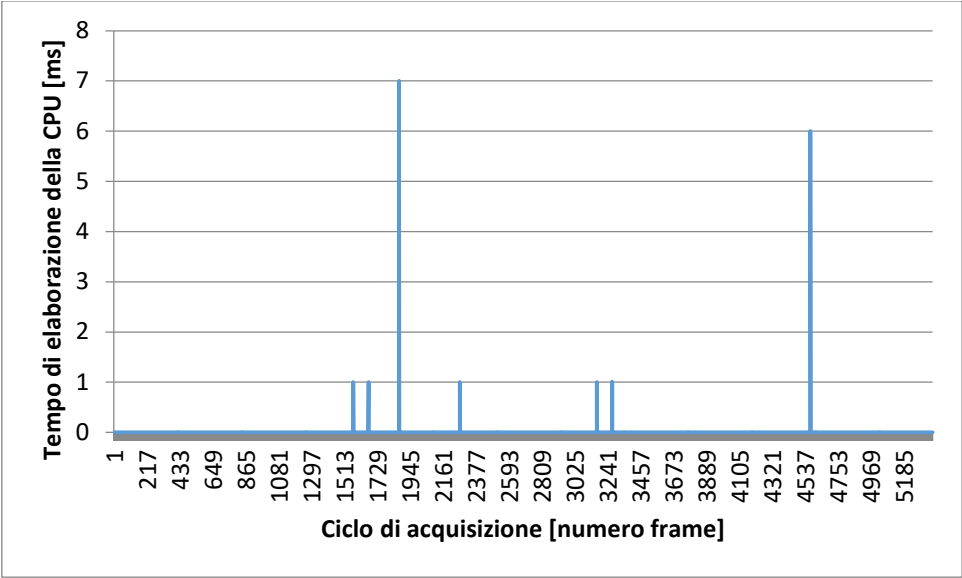


Grafico 8 Tempi di elaborazione della CPU del SDVA con frame grabber

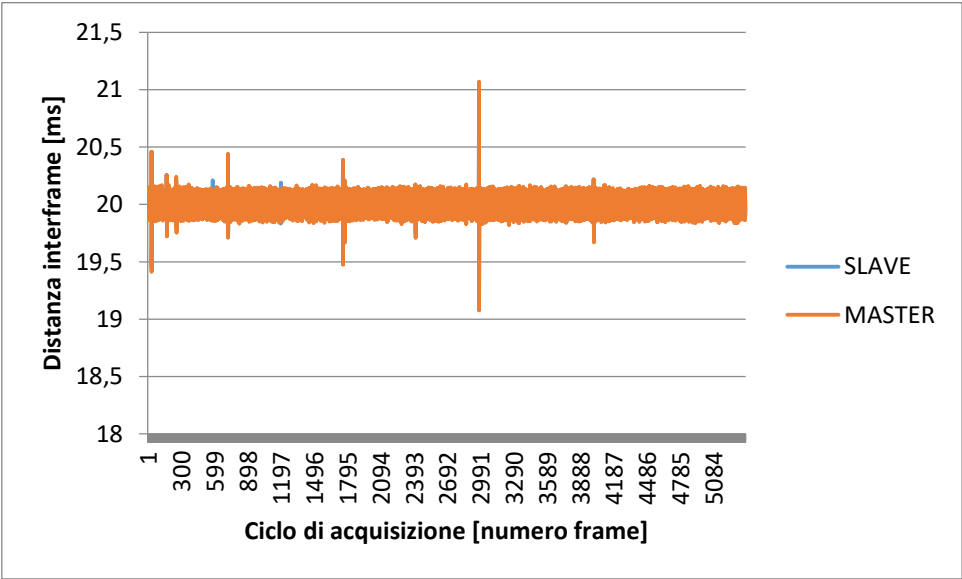


Grafico 9 Distanza interframe del SDVA con frame grabber, nel caso di frame rate di 50fps



Grafico 10 Distanza interframe del SDVA con frame grabber, nel caso di frame rate di 56fps

7 Applet Person1.6

Vengono illustrate qui di seguito le informazioni dettagliate dell'applet creata.

7.1 Caratteristiche

L'Applet "Person1.6" non supporta l'aggregazione dei link (LAG: Link Aggregation) e consente di utilizzare fino a due telecamere. Le caratteristiche dell'applet sono disponibili solo sulle prime due porte del frame grabber (PORT_A, PORT_B). Le telecamere devono essere matriciali a scala di grigio, di tipo GigE, con una profondità del pixel di 8 bit e una risoluzione massima di 2560(H) x 2048(V). L'applet processa i dati con una profondità di 8 bit. Le immagini acquisite vengono memorizzate nella memoria del frame grabber. Si può selezionare una regione di interesse (ROI) su cui effettuare l'elaborazione. Inoltre si può configurare il valore di grigio con cui sogliare l'immagine.

L'immagine originale, e i risultati della blob analisi sono trasmessi tramite canali DMA ad alta velocità. I formati di uscita dei pixel dei DMA sono 8 bit per pixel, per il primo, e 32 bit per pixel per il secondo.

Caratteristiche	Proprietà dell'Applet
Nome dell'Applet	Person1.6
Frame Grabber	microEnable IV VQ4GE/-GPoE
Numero di telecamere	2
Tipo di sensore	Matriciale
Formato della telecamera	Scala di grigio
Profondità di elaborazione in bit	8 bit
Dimensioni massime dell'immagine	2560*2048
Formato di output dei dati – DMA	8 bit scala di grigio
Formato di output dei dati – DMA_CoG	32 bit

Tabella 10 Sommario delle caratteristiche dell'applet Person1.3

7.2 Parametri

Per il corretto funzionamento dell'Applet è necessario configurare correttamente la ROI e il valore di grigio da utilizzare per la sogliatura dell'immagine.

Proprietà	Valore		
Nome	Device1_Process0_ROI_X_Length	Device1_Process0_ROI_Y_Length	Device1_Process0_markers_GrayThr_Number
Tipo	uint	uint	int
Politica di accesso	Lettura/Scrittura	Lettura/Scrittura	Lettura/Scrittura
Valori consentiti	Minimo 8 Massimo 2560	Minimo 2 Massimo 2084	Minimo 0 Massimo 255
Valore di default	1920	1080	80

Tabella 11 Proprietà dei parametri del Process0

Proprietà	Valore		
Nome	Device1_Process1_ROI_X_Length	Device1_Process1_ROI_Y_Length	Device1_Process1_markers_GrayThr_Number
Tipo	uint	uint	Int
Politica di accesso	Lettura/Scrittura	Lettura/Scrittura	Lettura/Scrittura
Valori consentiti	Minimo 8 Massimo 2560	Minimo 2 Massimo 2084	Minimo 0 Massimo 255
Valore di default	1920	1080	80

Tabella 12 Proprietà dei parametri del Process1

7.3 Banda

La seguente tabella elenca le massime bande per l'Applet Person1.3.

Descrizione	Banda
Banda media per telecamera	120 MPixel/s
Banda del DMA	120 MPixel/s (dipende dalla scheda madre del PC)
Banda del DMA_CenterOfGravity	360 MPixel/s (dipende dalla scheda madre del PC)

Tabella 13 Banda di Person1.3

La banda media per telecamera descrive la massima banda media permessa per ciascuna telecamera dell'interfaccia GigE.

È importante tener presente che l'impostazione di una ROI può diminuire significativamente la banda richiesta tra la telecamera e il frame grabber, ma affinché ciò si verifichi è necessario impostare la dimensione desiderata della ROI direttamente sulla telecamera, oltre che nell'applet.

7.4 Requisiti

Vengono elencati di seguito i requisiti software, hardware, e la licenze richiesta dal frame grabber.

7.4.1 Requisiti Software

Per eseguire questa applet è richiesta l'installazione del pacchetto Runtime della Silicon Software.

7.4.2 Requisiti Hardware

Per eseguire l'Applet Person1.6 è richiesto come frame grabber la scheda microEnable IV VQ4GE/-GPoE della Silicon Software.

7.4.3 Licenza

Per quest'applet è richiesta una licenza per la libreria Blob1D/2D di Visual Applet.

7.5 Interfaccia delle telecamere

L'Applet Person1.6 supporta delle telecamere GigE Vision. Il frame grabber microEnable IV VQ4Ge/-GPoE ha 4 porte GigE {0, 1, 2, 3}. L'Applet supporta solo 2 telecamere, che vanno connesse alle prime 2 porte.

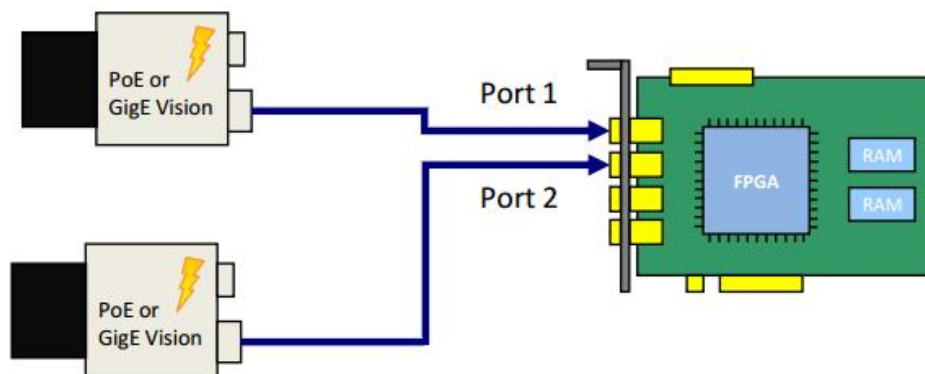


Figura 28 Interfaccia delle telecamere e Setup dei cavi

7.6 Trasferimento delle immagini e dei dati alla memoria del PC

Il trasferimento dei dati dal frame grabber al PC è di tipo DMA (Direct Memory Access). In quest'applet esistono 4 canali DMA. Due canali per ciascuna telecamera.

Per la telecamera collegata alla porta 0, gli indici dei DMA sono i seguenti

- 0 per il DMA che trasferisce l'immagine acquisita
- 1 per il DMA che trasferisce i dati dell'analisi delle regioni

Per la telecamera collegata alla porta 1, gli indici dei DMA sono invece

- 2 per il DMA che trasferisce l'immagine acquisita
- 3 per il DMA che trasferisce i dati dell'analisi delle regioni

8 GbECameraSDK

Di seguito, vengono illustrate in modo sintetico le informazioni relative al pacchetto Software dell'SDVA.

8.1 Struttura

Come abbiamo visto il pacchetto Software sviluppato è costituito da 3 moduli:

- Modulo di acquisizione
- Modulo di processing

- Modulo di supervisione

I files di progetto relativi al modulo di acquisizione sono:

- blob2d.h, blob2d.cpp, errorcodes.h
- gbestereo.h, gbestereo.cpp

Quelli relativi al modulo di processing sono:

- hungarianalg.h, hungarianhalg.cpp
- matchstereopoints.h, matchstereopoints.cpp
- markersdetector.h, markersdetector.cpp
- stereoprocessing.h, stereoprocessing.cpp

Ed infine quelli relativi al modulo di supervisione sono:

- monitor.h, monitor.cpp, monitor.ui

8.2 Requisiti

Per la compilazione del modulo software del SDVA è necessario aver installato la libreria di elaborazione di immagini OpenCV (16) e il pacchetto Runtime della Silicon Software (17).

Per l'esecuzione sarà necessario che sulla work station sia presente la scheda microEnable IV VQ4GE della Silicon Software.

8.3 Interfaccia delle telecamere

Il modulo software prevede che la telecamera sinistra sia la telecamera master, nella generazione del segnale di trigger, e che venga collegata alla porta 1 del frame grabber.

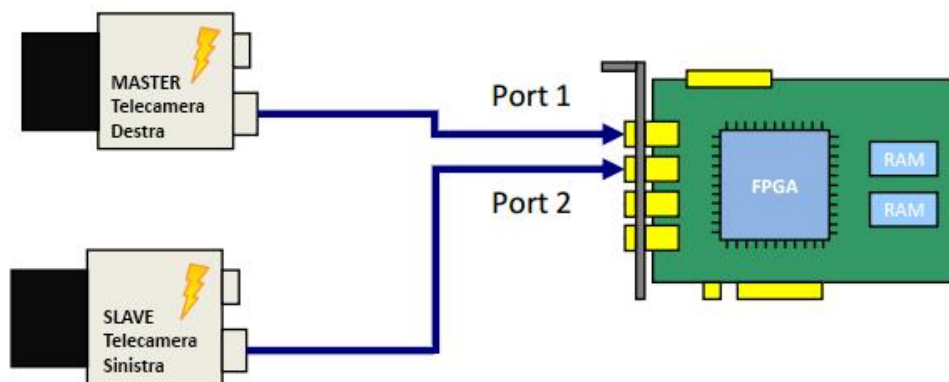


Figura 29 Interfaccia delle telecamere e Setup dei cavi

Appendice 1 – Struttura del Log del modulo di processing

Si mostra di seguito la struttura dei file Log prodotti dal modulo di processing. I file completi, di 5375 righe non sono riportati in forma stampata per economia di carta, ma sono inclusi come allegato.

I log riportano (in un ordine che dipende dall'elaborazione):

- La coppia di immagini elaborate con relativi timestamp
- La coppia di immagini elaborate con tempo impiegato dalla CPU per l'elaborazione dei centri di massa dei marckers (remapping e, ricostruzione 3D)
- Le coppie di immagini non elaborate (Watcher busy)

Processed 1 slave_timestamp 24007241 master_timestamp 24007209
 Processed 1 CPU_Processing_ms 0
 Processed 2 slave_timestamp 24027237 master_timestamp 24027211
 Processed 2 CPU_Processing_ms 0
 Processed 3 slave_timestamp 24047349 master_timestamp 24047349
 Processed 3 CPU_Processing_ms 0
 Processed 4 slave_timestamp 24067347 master_timestamp 24067347
 Processed 4 CPU_Processing_ms 0
 Processed 5 slave_timestamp 24087257 master_timestamp 24087210
 Processed 5 CPU_Processing_ms 0
 Processed 6 slave_timestamp 24107347 master_timestamp 24107347
 Processed 6 CPU_Processing_ms 0
 Processed 7 slave_timestamp 24127334 master_timestamp 24127334
 Processed 7 CPU_Processing_ms 0
 Processed 8 slave_timestamp 24147253 master_timestamp 24147215
 Processed 8 CPU_Processing_ms 0

Appendice 2 – Struttura del Log del modulo di acquisizione

Si mostra di seguito la struttura dei file Log prodotti dal modulo di acquisizione. I file completi, di 5375 righe non sono riportati in forma stampata per economia di carta, ma sono inclusi come allegato.

I log riportano:

- Il numero del frame acquisito dalla telecamera slave
- Il time stamp del buffer delle immagini della telecamera slave
- Il time stamp del buffer dei risultati dell'analisi dei blob della telecamera slave
- Il numero del frame acquisito dalla telecamera master
- Il time stamp del buffer delle immagini della telecamera master
- Il time stamp del buffer dei risultati dell'analisi dei blob della telecamera master

s_no 1 s_ts 24007241 s_CoG_ts 24007241 m_no 1 m_ts 24007209 m_CoG_ts 24007209
 s_no 2 s_ts 24027237 s_CoG_ts 24027237 m_no 2 m_ts 24027211 m_CoG_ts 24027211
 s_no 3 s_ts 24047349 s_CoG_ts 24047349 m_no 3 m_ts 24047349 m_CoG_ts 24047349
 s_no 4 s_ts 24067347 s_CoG_ts 24067347 m_no 4 m_ts 24067347 m_CoG_ts 24067347
 s_no 5 s_ts 24087257 s_CoG_ts 24087257 m_no 5 m_ts 24087210 m_CoG_ts 24087210
 s_no 6 s_ts 24107347 s_CoG_ts 24107347 m_no 6 m_ts 24107347 m_CoG_ts 24107347
 s_no 7 s_ts 24127334 s_CoG_ts 24127334 m_no 7 m_ts 24127334 m_CoG_ts 24127334

s_no 8 s_ts 24147253 s_CoG_ts 24147253 m_no 8 m_ts 24147215 m_CoG_ts 24147215
s_no 9 s_ts 24167343 s_CoG_ts 24167343 m_no 9 m_ts 24167343 m_CoG_ts 24167343
s_no 10 s_ts 24187334 s_CoG_ts 24187334 m_no 10 m_ts 24187334 m_CoG_ts 24187334
s_no 11 s_ts 24207240 s_CoG_ts 24207240 m_no 11 m_ts 24207214 m_CoG_ts 24207214
s_no 12 s_ts 24227346 s_CoG_ts 24227346 m_no 12 m_ts 24227346 m_CoG_ts 24227346
s_no 13 s_ts 24247261 s_CoG_ts 24247261 m_no 13 m_ts 24247222 m_CoG_ts 24247222
s_no 14 s_ts 24267318 s_CoG_ts 24267318 m_no 14 m_ts 24267318 m_CoG_ts 24267318
s_no 15 s_ts 24287317 s_CoG_ts 24287317 m_no 15 m_ts 24287317 m_CoG_ts 24287317
s_no 16 s_ts 24307257 s_CoG_ts 24307257 m_no 16 m_ts 24307212 m_CoG_ts 24307212

Appendice 3 – Struttura del Log del test di instabilità del puntamento

Si mostra di seguito la struttura dei file Log prodotti, dal modulo di processing del SDVA senza frame grabber, per il test dell'instabilità del puntamento. Non sono riportati in forma stampata per economia di carta, ma sono inclusi come allegato.

I log riportano:

- La coordinata x del pixel puntato sullo schermo
- La coordinata y del pixel puntato sullo schermo

Pixel 1210 1027

Pixel 1211 1026

Pixel 1210 1027

Pixel 1210 1027

Pixel 1211 1027

Pixel 1211 1027

Pixel 1211 1026

Pixel 1212 1026

Pixel 1212 1026

Pixel 1211 1027

Pixel 1211 1026

Pixel 1211 1026

Pixel 1210 1027

Pixel 1211 1026

Pixel 1211 1026

Pixel 1211 1026

Allegati del modulo hardware

Si allegano il file del progetto Visual Applet (.va), l'applet hardware da caricare sul frame grabber (.hap) e il file di log relativo al processo di sintesi dell'applet (.html).

- Person1.6.va
- Person1.6.hap
- Person1.6.html

Allegati del pacchetto software

Si allega la cartella GbECamera_SDK contenente:

- il file di progetto GbECamera_SDK.pro
- la cartella Headers
- la cartella Sources
- la cartella Forms

La cartella Headers contiene i seguenti files:

- blob2D.h
- errorcodes.h
- gbestereo.h
- HungarianAlg.h
- markersdetector.h
- matchstereopoints.h
- monitor.h
- stereoprocessing.h

La cartella Sources contiene i seguenti files:

- blob2D.cpp
- gbestereo.cpp
- HungarianAlg.cpp
- markersdetector.cpp
- matchstereopoints.cpp
- monitor.cpp
- stereoprocessing.cpp

La cartella Forms contiene il file monitor.ui

Allegati Log modulo processing

Si allegano i files:

- log_stereoprocessing_50fps.txt
- log_stereoprocessing_56fps.txt

Allegati Log modulo acquisizione

Si allegano i files:

- log_gbestereo_performance_50fps.txt

- log_gbestereo_performance_56fps.txt

Allegati Log test dell'instabilità del puntamento

Si allegano i files:

- TestInstabilità_double.txt
- TestInstabilità_0cifre_decimali.txt
- TestInstabilità_1cifre_decimali.txt
- TestInstabilità_2cifre_decimali.txt
- TestInstabilità_3cifre_decimali.txt
- TestInstabilità_6cifre_decimali.txt

Bibliografia

- [1. **emva**. Global Machin Vision Interface Standards. *emva*. [Online] <http://www.emva.org/wp-content/uploads/FSF-Vision-Standards-Brochure-A4-screen.pdf>.
- [2. **Matrox**. VisionProcessor. *Matrox*. [Online] http://www.matrox.com/imaging/en/products/vision_processors/.
- [3. **Gidel**. ProcFG. *Gidel*. [Online] <http://gidel.com/ProcFG.html>.
- [4. **National Instruments**. Elaborazione immagini FPGA su scheda. *National Instruments*. [Online] <http://sine.ni.com/nips/cds/view/p/lang/it/nid/210037>.
- [5. **SiliconSoftware**. Frame Grabbers. *SiliconSoftware*. [Online] <https://silicon.software/products/framegrabber/>.
- [6. **Silicon Software**. microEnable 5 ironman Product Series. *SiliconSoftware*. [Online] <https://silicon.software/products/framegrabber/me5-ironman/>.
- [7. —. microEnable 5 marathon Product Series. *SiliconSoftware*. [Online] <https://silicon.software/products/framegrabber/me5-marathon/>.
- [8. —. LightBridge Product Series. *SiliconSoftware*. [Online] <https://silicon.software/products/framegrabber/lightbridge/>.
- [9. —. microEnable IV Product Series. *SiliconSoftware*. [Online] <https://silicon.software/products/framegrabber/microenable4/>.
- [10. —. microEnable IV AQ4-GE. *Silicon Software*. [Online] https://silicon.software/pdf-generator/?product_id=712.
- [11. —. microEnable IV AQ4-GPoE. *Silicon Software*. [Online] https://silicon.software/pdf-generator/?product_id=715.
- [12. —. microEnable IV VQ4-GE. *SiliconSoftware*. [Online] https://silicon.software/pdf-generator/?product_id=718.
- [13. —. microEnable IV VQ4-GPoE. *Silicon Software*. [Online] https://silicon.software/pdf-generator/?product_id=721.

14. **Xilinx.** Spartan-3 FPGA Family Data Sheet. *Xilinx*. [Online]
http://www.xilinx.com/support/documentation/data_sheets/ds099.pdf.
15. **JAI.** Manuale Utente GO-5000-PGE. *JAI*. [Online] <http://www.jai.com/en/products/go-5000-pge>.
16. OpenCV (Open Source Computer Vision). [Online] <http://opencv.org/>.
17. **Silicon Software.** Installazione Completa. *Silicon Software*. [Online] <https://silicon.software/know-how/all-dls/>.
18. —. Frame Grabbers. *Silicon Software*. [Online] <https://silicon.software/products/framegrabber/>.