UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

Advanced face and
gesture recognition
for visual HMI

Tesis Doctoral

Tomás Mantecón del Valle
Ingeniero de Telecomunicación
2018

DEPARTAMENTO DE SEÑALES,
SISTEMAS Y RADIOCOMUNICACIONES

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

**Advanced face and
gesture recognition
for visual HMI**

Tesis Doctoral

Autor:

Tomás Mantecón del Valle

Ingeniero de Telecomunicación

por la Universidad Politécnica de Madrid

Directores:

Fernando Jaureguizar Núñez

Doctor Ingeniero de Telecomunicación

por la Universidad Politécnica de Madrid

Carlos Roberto del Blanco Adán

Doctor Ingeniero de Telecomunicación

por la Universidad Politécnica de Madrid

TESIS DOCTORAL

**Advanced face and
gesture recognition
for visual HMI**

Autor: Tomás Mantecón del Valle

Directores: Fernando Jaureguizar Núñez y Carlos Roberto del Blanco Adán

Tribunal nombrado por el Sr. Rector Magnífico de la Universidad Politécnica de Madrid, el día de de 2018.

Presidente: D.

Vocal: D.

Vocal: D.

Vocal: D.

Secretario: D.

Realizado el acto de defensa y lectura de la Tesis el día de de 2018 en

Calificación:

EL PRESIDENTE

LOS VOCALES

EL SECRETARIO

Agradecimientos

En primer lugar me gustaría agradecer a mis dos directores de tesis: Fernando y Carlos Roberto por todo el esfuerzo y tiempo que me han dedicado, ayuda que ha contribuido en gran medida a que el trabajo de estos últimos años haya terminado en esta tesis. También agradecer todo su apoyo en la relación con Airbus facilitando la integración del trabajo de la tesis en el demostrador. En especial agradecer a Fernando toda la ayuda administrativa que me ha facilitado el papeleo relacionado con los diversos trámites del doctorado.

Agradecer toda la ayuda recibida de los distintos miembros, que forman o han formado parte del GTI. Dar las gracias a Narciso por permitirme ser parte de dicho grupo. Por supuesto, agradecer a mis tutores en el proyecto fin de carrera: Luis y Massi por introducirme en el mundo de la investigación en general y el tratamiento de imágenes en particular. También por enseñarme y guiarme en la elaboración del trabajo lo cual me ha permitido seguir en este mundillo y este grupo de investigación. A Susana por hacer más llevaderos los viajes a Getafe y por intercambiar opiniones sobre las integraciones en el demostrador haciendo más ameno el trabajo. A Dani por enseñarme tantas cosas de programación y soportar todas mis tildes y eñes. A Carlos Roberto, César, y Fili porque, aunque les decía que lo de correr era de cobardes, me acabaron convirtiendo en un cobarde más. A Ana por compartir buenas y malas experiencias con el “machine learning” y el reconocimiento de gestos. Y al resto de miembros del GTI: Rafa, Pablo, Raúl, Jesús, José Luis, Sergio, Virginia, Adrián, Carlos Cuevas, Francisco, Julián, Esther, Nacho, Richard, Andrés, Alberto, Javi, Carmen, Marie, Dani Corregidor, Carlos Carmona, Marta, Lara, y tantos otros que han hecho más ameno el paso por el GTI. Agradecer a todos ellos su paciencia para ayudarme a generar las distintas bases de datos.

Por otro lado, agradecer también la ayuda recibida por parte de los miembros del proyecto SAVIER, tanto por parte de la gente propia de AIRBUS: a Juanjo por el interés de mi problema y la propuesta de caminos y soluciones al mismo. A José y Gemma por la unificación del trabajo de las distintas tesis. Y también a Sofía por su ayuda en la integración en el demostrador. Por otro lado, también me gustaría agradecer la ayuda recibida por parte del resto de doctorandos del programa por hacer posible que este trabajo llegue a su fin y haber

conseguido la integración conjunta en el demostrador.

Muchas gracias también a todos mis amigos tanto del colegio mayor, de la universidad, de León, de la estancia, . . . por el apoyo recibido durante estos años y por la paciencia que han tenido tanto en los buenos como en los malos momentos.

Gracias a mis padres por haber permitido que viniera a estudiar a Madrid y así acabar haciendo la tesis, gracias por todo su apoyo y ánimos durante todo este tiempo.

Resumen

Durante los últimos años se han propuesto soluciones que permiten una interacción hombre-máquina más natural e intuitiva gracias a la aparición de nuevos dispositivos que complementan o sustituyen a los teclados y ratones. Tales dispositivos basan la interacción en el uso de distintas partes del cuerpo humano para que ésta sea lo más parecida a una comunicación entre humanos, haciendo uso, por ejemplo, de las manos o de la voz. Dentro de los trabajos que utilizan gestos de las manos, los hay que proponen el uso de información visual adquirida por cámaras para el reconocimiento gestual. Estas soluciones introducen una ventaja frente a sistemas basados en sensores inerciales, al ser sistemas no intrusivos para el usuario. También es de gran importancia contar con nuevos sistemas que autentifiquen el acceso a estas interfaces, sustituyendo las claves de paso introducidas por teclado. Para ello, se puede utilizar el reconocimiento por huella dactilar, reconocimiento por iris, o el reconocimiento facial. El aumento del número de cámaras, tanto en entornos de vigilancia como embebidas en los propios dispositivos (por ejemplo, teléfonos móviles, tabletas, TV, etc.), ha despertado un gran interés en los sistemas de reconocimiento facial, ya que permiten la identificación del usuario sin necesidad de sensores adicionales.

A lo largo de esta tesis se presentan diversas soluciones con el objetivo de resolver tanto el reconocimiento facial como el reconocimiento de gestos de las manos, ambos basados en el uso de diversas fuentes de información visual. En cuanto a los sistemas de reconocimiento facial, se presentan tres sistemas cuya principal aportación consiste en el diseño de descriptores especialmente adaptados a la extracción de características relevantes de la cara de las personas usando imágenes de profundidad de alta resolución. Las soluciones presentadas permiten el reconocimiento facial desde diferentes perspectivas de la cara, a diferencia de la mayoría de sistemas existentes que solo admiten caras frontales. Además, el uso de imágenes de profundidad dificulta la suplantación de la identidad, siendo necesario un modelo tridimensional de la cara de la persona para su hipotética suplantación. Con el objetivo de validar los sistemas implementados, se han elaborado dos bases de datos, disponibles públicamente, que son pioneras en el campo del reconocimiento facial con imágenes de profundidad de alta resolución.

En cuanto al reconocimiento de gestos, se presentan diversas soluciones para gestos estáticos y gestos dinámicos. Similarmente a los sistemas previos de reconocimiento facial, se han propuesto nuevos descriptores altamente discriminativos diseñados para información de profundidad. Además, se han aplicado técnicas de reducción de la dimensionalidad para reducir sus requisitos de memoria y favorecer el funcionamiento en tiempo real. Dichos sistemas se han integrado en un demostrador de la empresa Airbus como parte del proyecto SAVIER. Este demostrador implementa un interfaz hombre-máquina basado en gestos para una estación de control de tierra de vehículos aéreos no tripulados. Para evaluar el rendimiento del sistema se han creado nuevas bases de datos, disponibles públicamente, compuestas por información de profundidad e infrarroja.

Abstract

In the last few years, many solutions have been proposed to allow a more natural and intuitive human-machine interaction thanks to the advent of new devices that improve the quality of interaction of keyboards and mouses. Different systems have been designed that make use of different human parts to offer a human-machine interaction as similar as possible to the interaction between humans, using hands or voice. Of special interest are the systems based on hand gestures and visual information, since they are non-intrusive (no sensor is wore by the user) unlike other alternatives as inertial sensors. On the other hand, new authentication systems for these mechanisms of interaction are required in substitution of passwords introduced by keyboard, such as fingerprint recognition, iris identification, or face recognition. The increase of the number of cameras in surveillance environments and embedded in electronic devices (mobiles, tablets, TVs, etc.), has awakened interest in face recognition system based on visual imagery, since no additional sensor is required for the authentication.

This thesis proposes new solutions to solve both face and hand gesture recognition using visual information. With respect to the face recognition systems, three solutions based on the design of feature descriptors adapted to the characteristics of the human face using high-resolution depth-images have been proposed. They allow the face recognition from different perspectives, unlike most of existing works that only accept frontal faces. Depth information makes more difficult the identity theft as a 3D model of the face would be needed for the identification. Two new databases have been created, and made publicly available, to properly evaluate the system, since no high-resolution image databases of faces are available.

With respect to hand gesture recognition, novel solutions are proposed to recognize both static and dynamic hand gestures, which include new descriptors specially designed for depth information that are highly discriminative. These descriptors have been combined with dimensionality reduction techniques to reduce the memory requirements and favor the operation in real time. The proposed systems have been integrated within an Airbus demonstrator as part of the project SAVIER. The demonstrator implements a hand gesture human-machine interaction for a ground control station that commands unmanned aerial

vehicles. New databases have been created, and made publicly available, composed by depth and infrared imagery to properly evaluate the system performance.

Nomenclature

ANN	Artificial Neural Network
ASH-LQDP	Array of Spatial Histograms of Local Quantized Depth Patterns
ASL	American Sign Language
ASTH-LQDP	Array of Spatio-Temporal Histograms of Local Quantized Depth Patterns
AUC	Area Under the Curve
BAG-D3P	Bag of Dense Derivative Depth Patterns
BLSTM-NN	Bidirectional Long Short-Term Memory Neural Networks
BSIF	Binarized Statistical Image Features
CHMM	Coupled Hidden Markov Model
CIPBR	Convex Invariant Position Based on RANSAC
CM	Confusion Matrix
CNN	Convolutional Neural Network
CS	Compressive Sensing
CSCW	Computer Supported Collaborative Work
DBN	Deep Belief Network
depthGestRecog	Depth based Gesture Recognition Database
DLQP	Depth Local Quantized Pattern
DS-LQP	Depth Spatiogram of Local Quantized Patterns

DSS	Data Distribution Service
DTW	Dynamic Time Warping
FPR	False Positive Rate
fps	frames per second
GCS	Ground Control Station
HCNF	Hidden Conditional Neural Field
HMI	Human Machine Interaction
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
HOOF	Histograms of Oriented Optical Flow
HOT	Histogram of Oriented Trajectories
HRRFaceD	High Resolution Range based Face Database
HSV	hue, saturation, value color model
IKF	Interval Kalman Filter
IPF	Improved Particle Filter
k-NN	k Nearest Neighbor
KBLBP	Kernel Based Local Binary Pattern
LBP	Local Binary Pattern
leapGestRecog	Hand Gesture Recognition Database Using Leap Motion
leapHandGestUAV	Leap Motion Hand Gesture dataset for UAV manage
LGBP	Local Gabor Binary Pattern
LPQ	Local Phase Quantization
LQDP	Local Quantized Depth Patterns
LRRFaceD	Low Resolution Range based Face Database

LSM	Least Square Matching
MALE	Medium-Altitude Long Endurance
MLP	Multi Layer Perceptron
MRF	Markov Random Field
MultiModHandGestRec	Multi-modal Leap Motion dataset for Hand Gesture Recognition
NIRS	near infrared spectroscopy
NN	Neural Networks
PCA	Principal Component Analysis
RANSAC	Random Sample Consensus
RBFNN	Radial Basis Function Neural Network
RIP	Restricted Isometry Property
RNN	Recurrent Neural Networks
ROC	Receiver Operating Characteristics
RPCA	Recursive Principle Components
RSS	Reduced Shape Signature
SAVIER	Situational Awareness Virtual EnviRonment
sEMG	surface electromyography
SIFT	Scale Invariant Feature Transform
SLSP	Semi-Local Structure Patterns
SMRF	Supapixel-Markov Random Field
SURF	Speeded Up Robust Features
SVD	Singular Value Decomposition
SVM	Support Vector Machine
ToF	Time of Flight

TPR	True Positive Rate
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Objectives	2
1.3	Thesis structure	3
1.4	Main contributions of the thesis	4
2	Context	7
2.1	SAVIER demonstrator	7
2.2	Visual information	7
2.2.1	Depth imagery	8
2.2.2	Infrared imagery	9
2.3	Object recognition process	10
2.3.1	Detection	10
2.3.2	Tracking	10
2.3.3	Recognition	11
2.3.3.1	Feature descriptor techniques	12
2.3.3.2	Classification techniques	12
3	State of the art	15
3.1	Introduction	15
3.2	Acquisition devices	15
3.3	Face recognition	19
3.4	Hand gesture recognition	23
3.4.1	Introduction	23
3.4.2	Vision-based solutions	24
3.4.3	Skeletal-based solutions	32
3.4.4	Other devices solutions	36
3.5	Face and hand gesture recognition	37

3.6	Conclusions	39
4	Vision-based Human-Machine Interaction	41
4.1	Introduction	41
4.2	Main system description	41
4.3	Face recognition module	42
4.3.1	Depth Local Quantized Pattern (DLQP)	42
4.3.1.1	Quantification of local depth differences	43
4.3.1.2	Computation of histograms of quantized codes	44
4.3.2	Depth Spatiogram of Local Quantized Patterns (DS-LQP)	45
4.3.2.1	Computation of spatiograms of bins of quantized codes	45
4.3.3	Bag of Dense Derivative Depth Patterns (BAG-D3P)	47
4.3.3.1	Dense spatial derivatives	47
4.3.3.2	Non-uniform quantization and codification	48
4.3.3.3	Multi-bag of words processing	48
4.3.3.4	Spatial block division	49
4.3.4	SVM classification process	50
4.4	Hand gesture recognition module	50
4.4.1	High discriminative scenario	50
4.4.1.1	Static gesture recognition	51
4.4.1.2	Dynamic gesture recognition	54
4.4.2	Real-time scenario	57
4.4.2.1	Real-time hand gesture recognition system using HOG and LBP descriptors with an SVM classifier	58
4.4.2.2	NN and HMM real-time solution based on the combination of HOG and LBP descriptors	63
4.5	Conclusions	66
5	Databases	67
5.1	Introduction	67
5.2	Face databases	67
5.2.1	High Resolution Range based Face Database (HRRFaceD)	68
5.2.2	Low Resolution Range based Face Database (LRRFaceD)	68
5.3	Hand databases	71
5.3.1	Depth based Gesture Recognition Database (depthGestRecog)	71
5.3.2	Hand Gesture Recognition Database Using Leap Motion (leapGestRecog)	73

CONTENTS

5.3.3	Multi-modal dataset for Hand Gesture Recognition (MultiModHandGestRec)	74
5.3.4	Leap Motion Hand Gesture dataset for UAV management (leapHandGestUAV)	76
5.4	Data augmentation	78
5.5	Conclusions	79
6	Results	81
6.1	Introduction	81
6.2	Metrics	81
6.3	Face recognition	83
6.4	Hand gesture recognition	93
6.4.1	Static hand gestures	93
6.4.1.1	Static hand gesture recognition using infrared imagery	94
6.4.1.2	Static hand gesture recognition using depth imagery	96
6.4.2	Dynamic hand gestures	99
6.4.2.1	Dynamic hand gesture recognition using depth information .	102
6.4.2.2	Dynamic hand gesture recognition using Leap Motion infor- mation	105
6.5	Integration into SAVIER project	108
6.6	Conclusions	115
7	Conclusions and future work	117
7.1	Conclusions	117
7.2	Future work	118
A	Leap Motion hand reflectance characterization	121
	Bibliography	123

List of Figures

2.1	Depth map difference between Creative Sens3D and Kinect v2.	8
2.2	Point cloud difference between Creative Sens3D and Kinect v2.	9
2.3	Leap Motion image.	9
2.4	Sliding window example. Extracted from [10].	11
2.5	Computation of the LBP feature descriptor.	12
2.6	Computation of the HOG feature descriptor.	13
2.7	Sample of the general HMM architecture.	14
3.1	Kinect 360.	16
3.2	Kinect one.	17
3.3	Asus Xtion Pro Live.	17
3.4	Creative Sens3D.	18
3.5	Leap Motion.	18
3.6	Face identification and face recognition.	19
3.7	Example of an LBP-based facial shape index representation. Extracted from [44].	20
3.8	Illustration of the improved view-based facial detector extracted from [56]. . .	21
3.9	Some key application for vision-based hand gesture recognition, extracted from [61].	23
3.10	A typical application scenario: The driver interacts with the infotainment system. The ToF-sensor captures the sensitive VOI marked in red. Extracted from [69].	25
3.11	(Top) A user doing the take-off gesture with a badminton racquet as the take-off pad. (Bottom left) Making a raise gesture. (Bottom right) Making a come gesture. Extracted from [76].	26
3.12	Non-contact mouse system setup.	27
3.13	An example of a hand model on the left, and two different scenario applications of the work presented in [93].	28
3.14	CAD Interaction Laboratory demonstration. Extracted from [126].	32

3.15	A user is writing text in 3D space using his/her fingertip over the Leap Motion sensor.	33
3.16	Acquisition setup extracted from [136].	34
3.17	The gesture recognition setup using a magnetic 3D position tracker extracted from [143].	36
3.18	Description of the functional components of GoldFinger smart glove extracted from [147].	37
3.19	Face-hand feature detection, extracted from [156].	38
4.1	Blocks diagram of the proposed system.	42
4.2	DLQP descriptor.	43
4.3	DLQP-based feature descriptor combined with the division approach.	44
4.4	Stages of the DS-LQP descriptor.	45
4.5	DSLQP-based feature descriptor computation.	46
4.6	H_esp histogram computation for an image region.	47
4.7	Block diagram of BAG-D3P feature descriptor.	47
4.8	First stage of Bag-D3P computation: dense spatial derivatives.	48
4.9	Third stage of Bag-D3P computation: multi-bag of words concatenating $N_h = 12$ histograms for chunks of $N_{div} = 9$ bits.	49
4.10	Block diagram of the DLQP-based hand pose recognition system.	51
4.11	Block diagram of the BAG-D3P-based hand pose recognition system.	52
4.12	Example of a CS process in which x is the input and sparse vector, ϕ is the transform matrix and y the output vector.	53
4.13	Block diagram of the ASTH-LQDP-based hand pose recognition system.	54
4.14	Illustration of the four stages/levels for the computation of the ASTH-LQDP descriptor along with the main tasks performed in everyone.	55
4.15	Neighborhood level computation.	55
4.16	Illustration of the block and image levels.	56
4.17	Computation of the ASTH-LQDP feature vector at the video sequence level.	57
4.18	Block diagram of the proposed system.	58
4.19	Image segmentation with different thresholds.	59
4.20	Block division and HOG and LBP histograms.	61
4.21	Example of the hand pose classification strategy using four hand poses and three different clusters (K-means strategy).	62
4.22	Example of the dynamic hand gesture recognition.	63
4.23	Block diagram of the proposed solution.	64
4.24	Structure of the proposed Neural Network architecture.	65

LIST OF FIGURES

5.1	Three sets of depth maps for different subjects from the HRRFaceD database.	69
5.2	Three sets of depth maps for different subjects from the LRRFaceD database.	70
5.3	Dynamic hand gestures of the depthGestRecog database.	72
5.4	Static hand gestures of the depthGestRecog database.	72
5.5	Database samples of the ten different gestures performed by different subjects.	73
5.6	Hand poses of the database.	75
5.7	Dynamic hand gestures of the database.	76
5.8	Hand poses for the gestures to command the UAV.	77
5.9	Hand poses for the gestures to interact with the payload of the UAV.	78
6.1	ROC curves for each classifier of the BAG-D3P algorithm using the LRRFaceD database.	85
6.2	ROC curves for each classifier of the BAG-D3P algorithm using the HRRFaceD database.	89
6.3	Samples of each gesture performed by the five subjects of the ASL Finger Spelling Database [72].	97
6.4	ROC curves for each classifier of the BAG-D3P+CS+SVM algorithm using the ASL Finger Spelling Database.	101
6.5	Evolution of precision, recall and F-score values with respect to the number of K-means clusters.	107
6.6	Block diagram of the connection between the real-time hand gesture recognition system and the GCS demonstrator.	110
6.7	Leap Motion sensor position over the GCS.	111
6.8	ROC curves for each classifier of the Real-Time-NN algorithm using the leapHandGestUAV database.	112
6.9	Take off.	113
6.10	Go up.	113
6.11	Go down.	113
6.12	Go right.	113
6.13	Go left.	114
6.14	Zoom in.	114
6.15	Zoom out.	114
6.16	Camera pan right.	114
6.17	Camera pan left.	114
6.18	Camera tilt up.	115
6.19	Camera tilt down.	115
6.20	Demonstrator environment.	115

A.1 Mean and variance intensity value of the hand regions with respect to the distance to the sensor.	122
--	-----

List of Tables

3.1	Details of different state-of-the-art solutions for face recognition.	22
3.2	Details of different state-of-the-art solutions for static hand gesture recognition solutions using visual information.	30
3.3	Details of different state-of-the-art solutions for dynamic hand gesture recognition solutions using visual information.	31
3.4	Details of different state-of-the-art solutions for hand gesture recognition using skeletal information.	35
5.1	Summary of the created databases.	67
6.1	Summary of the databases used to test different face recognition solutions. . .	83
6.2	Precision, recall, and F-score results of the DLQP, DS-LQP, and BAG-D3P algorithms using the LRRFaceD database.	84
6.3	Comparison of F-score values of the proposed descriptors (DLQP, DS-LQP, and BAG-D3P) with other state-of-the-art solutions: SIFT [52], and LBP [57] using the LRRFaceD database.	86
6.4	Precision, recall, and F-score results of the DLQP, DS-LQP, and BAG-D3P solutions applied over the HRRFaceD database.	87
6.5	Comparison in F-score value of the proposed descriptors (DLQP, DS-LQP, and BAG-D3P) with other state-of-the-art solutions: SIFT [52], and LBP [57] using the HRRFaceD database.	90
6.6	Mean accuracy results using the Biwi database.	92
6.7	Mean accuracy results using the IIIIT-D RGB-D database.	92
6.8	Mean accuracy results using the EURECOM database.	92
6.9	Summary of the databases uses to test different static hand recognition solutions.	93
6.10	Summary of the databases uses to test different dynamic hand recognition solutions.	93

6.11	Confusion Matrix with results of testing the BAG-D3P-based feature descriptor in combination with a CS solution over the leapGestRecog database. . . .	95
6.12	Precision, recall, and F-score values for the BAG-D3P+CS+SVM algorithm and the work presented by Huang in [52] using the leapGestRecog database. .	95
6.13	Confusion matrix obtained of testing the BAG-D3P-based feature descriptor in combination with a CS solution over the ASL Finger Spelling Database. . .	98
6.14	Comparison of precision, recall, and F-score values of testing different algorithms: DSQP-SVM, DSQP-CS-SVM and Pugeault [72] over the ASL Finger Spelling Database.	100
6.15	F-score values of the compared algorithms tested over the depthGestRecog database.	103
6.16	Values of M , N , and K for different values of R and P	104
6.17	F-score results for the proposed solutions over the depthGestRecog database.	104
6.18	Hand gesture recognition performance comparison of F-score value for the proposed algorithms and state-of-the-art works over the MultiModHandGestRec database.	106
6.19	Time performance of the Real-Time-SVM and the Real-Time-NN solutions. .	108
6.20	Hand gesture recognition performance comparison on F-score value of the real-time solutions.	110

Chapter 1

Introduction

1.1 Motivation

Nowadays, many are the works that have proposed several approaches to create new ways of solving the Human Machine Interaction (HMI) problem in a variety of environments, trying to create a more natural and intuitive interaction. Fields like medicine, scholar environments, entertainment, military, etc. are looking for new ways of interfacing their systems. In the military field, there is an increasing interest in managing one or more Unmanned Aerial Vehicles (UAVs) to satisfy the new demands of the UAV market. For this purpose, a human controller should efficiently interact with a Ground Control Station (GCS) to pilot and manage one or more UAVs simultaneously. In addition, such as interface must alleviate the workload of the human controller, who checks the information related to the UAV status and supervises a wide variety of information derived from the payload. Security is another important concern in military applications, and therefore it is mandatory to provide new ways of identification to allow the UAV control to specific users via biometric information.

Different works have proposed the use of various human body parts to perform the interaction: the whole body through human-pose recognition, eye-tracking, face recognition, voice recognition, and hand-gesture recognition. Especially hand-gesture interfaces have generated large expectations for their potential to alleviate the restrictive level of interaction of typical devices such as the keyboard or the mouse. Hand gestures allow to perform the interaction in a more natural and intuitive way. The kind of input information is another key part in HMI applications. Existing approaches can be divided into two main groups: the ones which use visual information, and those which make use of haptic devices.

Works dealing with visual information have the advantage that the user does not have to wear anything to perform the interaction. Early works proposed the use of basic color clues to easily detect the hand performing the gesture but, nowadays different kinds of visual

information are used: color, depth, infrared information, or a combination of them. In those works, depth information has becoming popular as it is more robust to indoor illumination problems, and it also provides more relevant information. One of the most popular devices is the Kinect v2 which provides depth information with high resolution, typically used for both hand and face recognition thanks to the provided skeleton information of the human body. Some other devices are specially designed for hand gesture interaction, such as the case of the Leap Motion that provides skeletal information, in addition to near-infrared imagery. The use of this kind of imagery has the advantage that the clutter is really low, allowing to obtain better recognition results and favoring a real-time solution.

On the other hand, an HMI solution for a GCS must secure the interaction system, allowing the access to certain people. Instead of using keyboards, mice, and other usual devices to enter a security password, new biometric identification procedures are recommended, such as fingerprint, eye, and face verification.

Specially interesting are the face recognition systems. These have been growing along the last years, given the fact that many devices have attached a camera. Even it is relatively common nowadays that they use the face to unlock them. Similarly to the field of hand gesture recognition, early face recognition works have exclusively used color information. However, the literature has proven that systems based only on color imagery can be easily hacked. For this reason, face recognition systems have evolved to use multimodal information: color plus depth imagery for example. This information is more robust, complex, and difficult to hack, increasing its security.

A key advantage of using visual information is that the same device can be used for both identification and to HMI, reducing the cost and the complexity of the system.

1.2 Objectives

The work of this thesis is part of the Airbus' Situational Awareness Virtual EnviRonment (SAVIER) project, which main aim is to develop technologies for the new generation of GCS to manage UAVs. This project has a great variety of objectives, one of them is to allow a more natural and intuitive way to interact between the operator placed at the GCS and the UAV. This thesis is focused on this topic, a hand gesture recognition system to be integrated within an Airbus demonstrator, along with the results of other theses of the project SAVIER. Apart from facilitating an easier interaction, the developed system should be robust and work on real-time. In this scenario, real-time performance refers to offer a flow of hand gesture recognition that enables a fluid interaction between the GCS and the human operator, minimizing any kind of delay. This means a time period of some milliseconds.

A secondary goal of this thesis is to design a face recognition system to control who is

able to perform the interaction for security purposes. In this line, a global system solving both problems, hand gesture and face recognition have been designed, which uses visual information in a touch-less environment.

At the beginning of the project SAVIER, the most restrictive objective was the robustness of the system, without computational cost or real time restrictions. For this reason, depth information and very discriminant and complex descriptors were used and developed to obtain as much discriminative information as possible. Therefore, the designed descriptors were highly discriminative, but their computation and the related classification processes were also quite time demanding. However, the Airbus policy changed during the second period of this thesis, requiring solutions with a reduced computational cost to operate in real-time. This fact motivated the decision of exchange the use of generic depth information to the specific near-infrared one provided by the Leap Motion, since the clutter information in those images is really small reducing the time and complexity needed to make the recognition process.

With respect to the hand gesture recognition, the viability of different approaches has been studied by using depth and infrared information, proposing new robust feature descriptors. With the aim of reaching real-time performance, different approaches to reduce the information have been considered. Finally, a system trading off the robustness and the real-time performance has been designed, and integrated within the SAVIER demonstrator.

In the case of face recognition, robustness prevails over real-time requirements in order to verify the identity of a person who is trying to make the interaction. For this reason, high-resolution depth information is used along with specific feature descriptors to characterize with great detail human faces.

1.3 Thesis structure

In addition to this introductory chapter, this thesis is structured in other six chapters, whose contents are briefly outlined next.

In Chapter 2, a thesis context is provided giving information about the SAVIER demonstrator, the sources of visual information used along this work, and introducing some algorithms in which the solutions presented along the thesis are based on.

In Chapter 3, different state-of-the-art techniques are analyzed. First, devices managing different kind of visual information, such as color, depth or infrared, are introduced. Then, techniques proposing solutions for face recognition by using depth or color information are presented. Finally, some techniques trying to perform an HMI via hand gesture recognition are described.

In Chapter 4, a description of the proposed solutions for both HMI and face recognition

system are described, considering different kinds of visual imagery and descriptors.

In Chapter 5, a description of different databases to validate the system is performed. They comprise two depth-based face databases, a depth-based database for dynamic hand gesture recognition, and three different infrared-based databases for both static and dynamic hand gesture recognition.

In Chapter 6, exhaustive results for the developed systems are presented. Additionally, comparisons between the proposed systems and other state-of-the-art solutions are described.

Finally, Chapter 7 includes the conclusions of this thesis.

1.4 Main contributions of the thesis

The contents of this thesis have been published in several international peer-reviewed conferences and journals, and different databases have been generated and made them publicly available. Next, a summary of the main contributions is presented.

- Face recognition
 - A depth-based face recognition algorithm, specially adapted to high range resolution data acquired by the new Kinect v2 device, was presented in [1]. A novel descriptor called Depth Local Quantized Pattern descriptor has been designed to make use of the extended range resolution of the new sensor. In addition, a new depth-based face database acquired by the new Kinect v2 device has been created and made it publicly available.
 - A novel and robust biometric face identification algorithm for access control applications was proposed in [2]. The key contribution is the design of a high discriminative feature descriptor for depth imagery, called Depth Spatiogram of Local Quantized Patterns.
 - A novel biometric face recognition algorithm using depth cameras was proposed in [3]. The key contribution is the design of a novel and highly discriminative face image descriptor called Bag of Dense Derivative Depth Patterns. The proposed solution can recognize people faces from a wide range of poses, not only frontal ones, increasing its applicability to real situations.
- Hand gesture recognition
 - A hand gesture recognition solution based on depth imagery was proposed in [4] to enable an interaction between human operators and Unmanned Aerial Vehicles. The key contribution is the design of a spatio-temporal depth descriptor

specially designed to efficiently work with depth video sequences. Other major consideration is the hand sign database developed and made it publicly available.

- A novel and robust vision-based human-machine interface system to naturally interact with computers/smart devices was proposed in [5]. The key contribution is the introduction of a Compressive Sensing technique to largely reduce the dimensionality of highly discriminative feature descriptors, which originally have an excessive and inoperative high dimension to be applied to a Support Vector Machine based classifier.
- A hand gesture-based recognition system is presented in [6] with the aim of recognizing finger-spelling using the American Sign Language. The main novelty is the introduction of a Compressive Sensing step to reduce the dimension of a depth-based feature descriptor, called Depth Spatiograms of Quantized Patterns, which is very discriminative, but also too large for its practical application.
- A hand gesture recognition system using near-infrared imagery, acquired by a Leap Motion device, was proposed in [7]. The recognition system directly characterizes the hand gesture by computing a global image descriptor, called Depth Spatiogram of Local Quantized Patterns, without any hand segmentation stage. To deal with the high dimensionality of the image descriptor, a Compressive Sensing framework is applied. Also, a new hand-based near-infrared database has been created.

Chapter 2

Context

This chapter introduces different topics to provide an overview about the context of this thesis, a brief presentation of the SAVIER demonstrator (Section 2.1), information about the characteristics of depth and infrared imagery (Section 2.2), and an overview of a general object detector solution (Section 2.3).

2.1 SAVIER demonstrator

The work of this thesis is part of the project SAVIER, whose main objective is the development of new solutions to manage UAVs from a GCS. All the research work has been tested in a GCS demonstrator at Airbus facilities in Getafe. This demonstrator simulates the same systems existing in a conventional GCS.

More specifically, this thesis develops new HMI algorithms to interact with an UAV with the purpose of substituting or complementing the use of the keyboard (to set the flight parameters of the UAV) or a joystick (to manage the payload of the UAV). The HMI needs to be designed according to the specifications of the demonstrator: real-time performance, a complete set of actions to be interfaced, and the adoption of specific communication protocols.

2.2 Visual information

Many are the commercial devices that provide different kind of visual information. The most usual, and easy to obtain types of visual information understood as matrices of values, or images, are: visible information (obtained by RGB cameras or by monochrome ones), depth information (obtained by stereo cameras, or by depth sensors), and infrared information (with different working ranges). In this thesis, depth information acquired by depth sensors

Figure 2.1: Depth map difference between Creative Sens3D and Kinect v2.

and infrared images acquired by the Leap Motion device are used, which are described in the following sections.

2.2.1 Depth imagery

To acquire depth images, or depth maps, either a stereo par of RGB cameras or an active depth sensor is used. In both cases, a gray scaled image is obtained whose pixel value encodes the distance of an object to the sensor. Depth maps acquired using a stereo par have problems related to illumination changes and can work indoor and outdoors. On the other hand, depth sensors are quite robust to illumination variations, but they can only work in indoor scenarios. Two are the technologies used by depth sensors. One of them projects a known pattern, and depending on the pattern deformation over the real scene, an estimation of the distance is computed. The other technology, called Time of Flight (ToF), measures the time of a pulse of light traveling between the depth sensor and the scene objects in order to estimate the depth.

The precision of the depth maps has increased considerably along the past years. An evolution can be seen in Figure 2.1, where two depth maps of the same scene acquired by two sensors (Creative Sens3D [8] and Kinect v2 [9]) are shown. In Figure 2.1a, the depth map obtained by the Creative Sens3D is quite noisy: pixels that are at the same distance does not have the same intensity value, borders are not well defined, and for some body parts, like the hair, the sensor is not able to estimate a depth value. In contrast, depth information acquired by the Kinect v2 is less noisy (see Figure 2.1b), borders are well defined, and the hair depth values are well defined. Depth information obtained by the Kinect v2 has more detail in the face of the subject than the one obtained by the Creative Sens3D as it can be seen in Figure 2.2.

Figure 2.2: Point cloud difference between Creative Senz3D and Kinect v2.

Figure 2.3: Leap Motion image.

2.2.2 Infrared imagery

Infrared imagery covers a wide range of the electromagnetic spectrum (approximately from 720 to 1000 nanometers), although depending on the application domain, different ranges of spectrum are used. In our case, the Leap Motion device obtains infrared images in a range centered in 850 nanometers. Near infrared imagery is quite similar to a gray-scale image in the visible range, although less texture is usually preserved. Leap Motion images have a special characteristic which makes them really useful for hand gesture recognition: closer objects have much higher intensity value than background objects. An example of such images can be seen in Figure 2.3, where the hand (closer object) has a higher intensity value than other scene objects.

2.3 Object recognition process

A general recognition process can be divided into three main stages: detection, tracking, and recognition. Detection is the stage that locates inside the image the object to be recognized, the face or the hand in our case. Tracking is the stage that locates and preserves the identity of objects over the time, i.e. describes the object movement and its trajectory. Finally, recognition is the stage that infers the object category. In the following subsections, a selection of the most representative techniques and approaches are presented for every stage.

2.3.1 Detection

The first task of an object recognition algorithm is to localize the object in the image. This is a challenging task due to several facts: appearance changes, different perspectives, scale variations, background clutter, high degrees of freedom of articulated objects (such as a hand).

Many solutions have been proposed to solve those problems. Some of them are based on the sliding window strategy, which tries to localize the object of interest by moving one window over the image with certain overlapping. As the size of the object within the image is unknown, this process is done over different image scales (see Figure 2.4). This strategy has high computational cost, as for each window, it is necessary to check if the object of interest is present inside. For that reason, some algorithms try to use additional clues or information. For example, regarding the recognition of body parts, the skin color is traditionally used. However, those techniques fail in the presence of illumination changes, and also by the fact that there is no a prototype color skin. Finally, other techniques that use depth images assume that the objects of interest are the closer ones to the camera, specially for face or hand gesture recognition. However, this heuristic is not always true.

2.3.2 Tracking

The tracking process for hand gestures is quite challenging because the hand movement can be very fast and the appearance can change largely between consecutive time steps. Different solutions have been proposed in the literature to perform the hand tracking process.

Template-based methods search a new object instance at the spatial vicinity given a template as reference that contains the object to be tracked. A slow motion of the hand or a high frame rate acquisition is needed for a satisfactory performance. The correlation-based feature tracking algorithm presented in [11], assumes that the movement of the hand is limited, and then the search for the new hand is made in a small neighborhood of the

Figure 2.4: Sliding window example. Extracted from [10].

previous object location. In [12], they use a correlation-based solution to obtain gesture patterns using a static camera. Contour tracking uses deformable contours or snakes to track regions over consecutive frames. For example, a finger spelling detector is developed using this technique in [13] .

Other solutions are based on probabilistic methods. The Kalman filter is a popular technique that allows a real-time tracking performance. Other methods allow to perform the process in case of occlusions [14], or if more than one object has to be tracked [15]. In other cases, multiple cameras are used [16]. Particle filtering approaches, represent the probability distribution of object locations by a set of particles. In [17] it is proposed to reduce the dimensionality of particle filtering models to simulate the hand anatomy.

Finally, another tracking solution is CamShift, based on the mean shift algorithm. This uses a kernel-based appearance model to represent the targets. One proposal for hand tracking based on the use of CamShift uses a real-time detector to take into account the velocity of the hand [18].

2.3.3 Recognition

Object recognition can be divided in two stages: a feature extractor that obtains discriminative information from the object, and a classification stage that identifies the object. Along this section, the most popular algorithms are described.

Figure 2.5: Computation of the LBP feature descriptor.

2.3.3.1 Feature descriptor techniques

Local Binary Pattern The Local Binary Pattern (LBP) [19] is a feature descriptor that, for each pixel of the image, computes the difference between the intensity value of a pixel and the intensity value of its neighborhood, by default 8 neighbors (see Figure 2.5). A thresholding process over the obtained difference values is performed quantifying the differences with 1 bit, whose value is 1 if the difference is positive and 0 if it is negative. Once the quantization is computed, the resulting binary number from all the neighborhood values is converted into a decimal number, which is in turn used to build a histogram by considering all the decimal number codes (called Local Binary Patterns), computed for each pixel of the image. As the neighborhood has 8 pixels, and 1 bit is used to quantify the sign value, the length of the histogram is $2^8 = 256$ bins. The overall process is shown in Figure 2.5. Different modifications of the original LBP have been proposed [19] changing the size of the neighborhood, adding temporal information, making it rotational invariant, etc.

Histogram of Oriented Gradients The Histogram of Oriented Gradients (HOG) [20] is based on the computation of the distribution of the directions of the gradients in an image. The algorithm has different stages. The first one is the gradient computation for which a filtering process is needed. For that purpose, the following filter kernels are used: $[-1, 0, 1]$ and $[-1, 0, 1]^T$. The objective is to capture the contour, silhouette and some texture information. Then, in a second stage, the gradient image is divided into cells of 8×8 pixels, and for each of them a histogram of 9 bins is computed, where each bin corresponds to certain angle of the gradients. A third stage computes a block normalization in which the histograms of 4 adjacent cells are normalized to obtain a new histogram of 36 bins for a block of 16×16 pixels. The final stage concatenates all the histograms of every block to obtain the final feature descriptor. The entire process is summed up in Figure 2.6.

2.3.3.2 Classification techniques

Support Vector Machine Support Vector Machine (SVM) [21] is a supervised algorithm that is used for classification. The input data of the algorithm are vectors in an n-dimensional

Figure 2.6: Computation of the HOG feature descriptor.

space. In a two-class scenario, an SVM obtains a hyper-plane able to differentiate the vector descriptor of the two classes. But for many cases, it is not easy to find a hyper-plane to separate the data in the original n -dimensional space. For those cases the SVM has a technique called the kernel trick, which are functions that takes the input n -dimensional space to a higher dimensional space, in which the input data is separable. Generally, the problem is more complex as it is necessary to differentiate among more than two classes. In these cases, a multi-class SVM is needed. To solve that problem, the usual approach is to reduce the multi-class problem into multiple binary classification problems. This can be done by a one-versus-all strategy, in which one binary classifier is trained for each class, or by a one-versus-one strategy in which for each class one binary classifier is trained with each one of the other classes.

MultiLayer Perceptron A MultiLayer Perceptron (MLP) [22] is a Neural Network (NN) that at least has three layers of nodes, the first layer is the input one which, has the same number of nodes as the number of elements of the input feature descriptor. The last one is the output one and has as many nodes as classes has the database. The number of intermediate layers, and their number of neurons changes from one to another application, and is a design parameter. Since MLPs are fully connected networks, each node in one layer is connected to every node of the following layer. Every neuron has associated an activation function which maps the weighted inputs to the output of each of them. The usual activation functions are the sigmoids or the hyperbolic tangent. In the learning process, the weight of each neuron changes until the net is adapted to the training feature descriptors. This is usually carried

Figure 2.7: Sample of the general HMM architecture.

out by a backpropagation algorithm which is a generalization of the least mean squares algorithm in the linear perceptron. For that process, an initial weight is associated to each connection between every pair of neurons, then, the training samples are passed through the network to obtain an error value. This error value is then backpropagated through the network, and the weights change to minimize the error value.

Hidden Markov Model A Hidden Markov Model (HMM) [23] is a statistical Markov model with hidden states which are unobserved. This kind of model is used to solve problems of recognition of sequences or time series, and it has been widely used in speech and gesture recognition applications.

The general architecture of a standard HMM can be seen in Figure 2.7, where each oval represents a random variable. The variable $x(t)$ is the hidden state at time t while the variable $y(t)$ is the observation at time t . Arrows in Figure 2.7 indicate conditional dependencies, so it can be observed how the value of the hidden variable $x(t)$ at time t only depends on the value of the hidden variable at time $t - 1$. This is similar to the observed variable $y(t)$ that just depends on the value of the hidden variable at the same time ($x(t)$). The states of an HMM define the possible value that a hidden variable can take at a certain time among N possible values. The observations, i.e. the possible values of the observed variables, can be either discrete or continuous. In case of discrete observe variables, they can take a value among M possibilities, which are the emission values.

HMMs are characterized by three different kind of parameters: start probabilities which indicate the probability that a hidden variable takes certain value at the beginning, the transition probabilities that define the probability of transition of a hidden variable between two different states from time t to time $t + 1$, and the emission probabilities which indicates the probability that the observed variable took a certain value with respect to the value, at the same time, of the hidden variable.

The objective of an HMM is to find the best set of state transitions and emissions probabilities given one or a set of output sequences. Parameters are usually estimated by deriving the maximum likelihood given the set of output sequences. This value is usually derived by using the Baum-Welch algorithm [24].

Chapter 3

State of the art

3.1 Introduction

Traditionally, the interaction between humans and machines has been performed by using different kinds of peripherals such as keyboards, mouses, joysticks, etc. In many practical environments, the simultaneous use of several of them is needed to perform a complex interaction. However, they have some limitations, specially with modern devices, such as touch-screens, mobiles, autonomous robots, and virtual or augmented reality tools. For this reason, during the last years, a lot of work has been done in the field of HMI to introduce new solutions. Therefore, new ways of interaction should be designed to establish an efficient, natural, friendly, and intuitive communication between humans and machines. And thus, taking advantage of the overwhelming amount of information that the human being can access nowadays.

Along this chapter, different solutions are presented for HMI that can be grouped in two application fields: face recognition for the identification of the person who performs the interaction, and hand gesture recognition for the own HMI. The following subsections describe the state-of-the-art of these fields. First, section 3.2 introduces new devices that can acquire different sources of information. Then, a review of different works about face recognition is described in Section 3.3. Section 3.4 deals with hand gesture recognition related works. And finally, Section 3.5 deals with state-of-the-art works solving both face and hand gesture recognition problems.

3.2 Acquisition devices

During the last years, a big group of solutions for gesture recognition have exclusively used color imagery because of the low cost and easy integration of such information. Stereo

imaging has been also considered, but to a lesser extent because the higher computational burden. Depth imagery has been another alternative widely used in the last years due to the decreasing price of depth cameras. However, the resolution of the provided depth images is significantly smaller than the one provided by color sensors. Remarkably, a new generation of low-cost devices that can acquire multimodal information have fostered new algorithms to enhance the HMI. These devices typically provide color information in combination with depth, infrared, or skeletal one. Another advantage of these devices is that the factory default calibration among the different sources of information they combine, avoiding an explicit and complex calibration process by the user. Some examples are the Microsoft Kinect for Xbox 360 (also called Kinect v1) [25], the Microsoft Kinect for Xbox One (called Kinect v2) [9], the Asus Xtion Pro Live [26], the Creative Senz3D [8], and the Leap Motion [27].

Figure 3.1: Kinect 360.

The Kinect v1 was the first multimodal low-cost device (see Figure 3.1). It has a color sensor which captures images with a resolution of 640×480 pixels at 30 frames per second (fps). To obtain the depth information, an infrared projector is used which generates a light pattern that is projected on the scene. This light pattern is received by an infrared sensor, and the distortion with the original projected pattern is used to compute the depth between nearby objects and the device. With those values a depth map of the scene can be constructed. The depth resolution is higher in a range between 70 centimeters and 6 meters, and the spatial resolution is 640×480 pixels with a frame rate of 30 fps. This sensor also provides general skeletal information of the entire body, but not specific one related to the hand or fingers. Different HMI applications have been presented along the literature based on this device. In [28], a solution to interact with the Windows operating system is proposed by using depth information. Another hand gesture recognition solution for sign language recognition is presented in [29]. Other solution for interacting with 2D and 3D medical images is presented in [30].

The Kinect v2 device (see Figure 3.2) provides color imagery with higher resolution than the Kinect v1, 1920×1080 pixels, at 30 fps. The depth information can be also acquired with more precision, but within a smaller range, between 50 centimeters and 4.5 meters.

Figure 3.2: Kinect one.

The Kinect v2 uses a different technology, Time of Flight (ToF), which is more accurate. The provided depth images have a resolution of 512×424 pixels at 30 fps. It also provides skeletal information of the entire body with more joints than the one provided by Kinect v1, and also more details of the hand and the fingers. In the field of HMI, different solutions have been proposed using this device. In [31] a driver who interacts with different car subsystems is presented allowing to recognize different gestures and numbers performed by both hands. In [32], a gesture-based recognition system is used to control the movements of a quadrotor. Also, an interactive solution which combines hand and face recognition is presented in [33] for the interaction with a computer.

Figure 3.3: Asus Xtion Pro Live.

The Xtion Pro Live (see Figure 3.3) gives color information with a resolution of 640×480 pixels at 30 fps, and depth imagery with a resolution of 640×480 pixels at the same frame rate. The depth information is obtained in a range between 50 centimeters and 3.5 meters using the structured light technology, similarly to the Kinect v1. Different solutions have been presented in the literature using this device for the field of HMI. For example, in [34] a gesture recognition system which allows a robot to recognize a gesture performed by a human that is in front of it is presented. In [35], an interaction environment is presented, in which visual information is used for gesture recognition and interaction. Finally, an augmented reality interaction environment that makes use of the Xtion Pro Live is presented in [36].

The Creative Senz3D device (see Figure 3.4) provides color information with a resolution of 1280×720 pixels at 30 fps, and depth information with a resolution of 320×240 pixels at 30 fps. The active depth range is between 15 centimeters and 1 meter. As in the case of Kinect v2, the depth information is acquired by using the ToF technology. This device also

Figure 3.4: Creative Sens3D.

provides additional information about the hand location, but not skeletal information. HMI applications using this sensor include rehabilitation solutions as the one presented in [37], which measures the active range of the hand motion. Other rehabilitation solution introduces a system to improve the movement of the upper extremity [38]. An interactive system using the detection and tracking of the fingertip is presented in [39]. Finally, a solution that is able to recognize the Vietnamese sign language is presented in [40].

Figure 3.5: Leap Motion.

The Leap Motion is another device specially designed for HMI applications (see Figure 3.5). It has been conceived for close interaction, similarly to the Creative Sens3D, and uses the motion of the hands in a contactless way. The device is able to estimate the hand skeleton for certain hand poses, and performs the tracking of the skeleton model. The Leap Motion is an active sensor equipped with three LEDs that emit infrared light at a wavelength of 850 nanometers, and two near-infrared cameras sensible to the reflected infrared light of those objects that are close to the sensor. Therefore, only objects close enough to the sensor (around 50 centimeters) can reflect a representative amount of light to be sensed by the infrared cameras. Moreover, background objects that are usually further are (almost) no sensed by the Leap Motion. Thus, most of the clutter is discarded. The Leap Motion SDK provides the hand skeleton and the infrared imagery of the scene. The skeletal information consists on the position and velocity of the fingertips, the palm center, the wrist position, the length of the fingers, and the location of the intermediate joints of the fingers.

Figure 3.6: Face identification and face recognition.

Regarding the near infrared imagery, the Leap Motion provides two streams of images, one per camera, with a resolution of 640×240 pixels at a variable frame rate between 30 and 200 fps. The images are affected by a huge distortion due to the wide-angle lenses of the cameras. Most of the research activity associated with the Leap Motion has been focused on the use of the hand skeletal information, without using the near-infrared imagery due to its distortion. Some works propose to use the trajectory of different hand parts [41], using the palm trajectory to recognize on-air written numbers. In [42], a system to recognize the signature for checking the identity of a person is proposed. In some cases, the proposed solution uses more than one Leap Motion, combining their information to be more robust to skeletal estimation problems [43].

3.3 Face recognition

The solutions presented in the literature try to solve two main problems which are shown in Figure 3.6. Some works try to recognize the person from a group of them and other works perform an identification task checking that the face of a person really belongs to that person. Face recognition and identification systems have many practical applications, such as people awareness in controlled environments, entrance allowance in security areas, identification of missing/searched people, and HMI. For this reason, a wide range of works have been published in the area of computer vision with the aim of recognizing people faces using different kinds of solutions.

Figure 3.7: Example of an LBP-based facial shape index representation. Extracted from [44].

Many of the proposed solutions use color information to solve both problems: face recognition, and face identification. They have to deal with many problems related with color imagery such as illumination variations, changes in the scale, perspective, etc. Another relevant challenge for security scenarios is to prevent the hacking of the system by using a picture of the subject. To alleviate these problems, specially the last one, many solutions have proposed to incorporate depth or 3D information. Those sources of information are more difficult to fake, since a 3D model of the face of a person would be required, rather than a picture of the face. This section describes different systems for face recognition by using visual information.

Most of the existing face recognition and identification algorithms use color imagery as input data, which is usually processed by a machine learning framework. Since color information is strongly dependent on the illumination conditions, a feature extraction stage is needed to obtain a feature vector that robustly represents the face information. Some of the most popular feature extraction techniques used in face recognition are: the Scale Invariant Feature Transform (SIFT) [45], the Histogram of Oriented Gradients (HOG) [46], the Local Binary Pattern (LBP) [47], the Principal Component Analysis (PCA) [48], the Local Gabor Binary Patterns (LGBP) [49], the Local Phase Quantization (LQP) [50], and the Binarized Statistical Image Features (BSIF) [51]. Of special consideration is the LBP descriptor and its numerous variants [52], which have proven to be highly robust to even non-linear illumination variations. These image feature descriptors are used in combination with different classification techniques for the purpose of face recognition, such as Support Vector Machine (SVM) [47], AdaBoost [46], or Neural Networks (NN) [53]. All these works have common strategies to be robust to changing illumination conditions. However, the performance of a color-based recognition algorithm is still greatly affected by them. This fact compromises the use of such systems for highly demanding applications, such as security

Figure 3.8: Illustration of the improved view-based facial detector extracted from [56].

and defense.

Depth imagery-based solutions have emerged in the last years to solve some of the problems posed by color imagery. The first advantage of depth imagery is to be almost immune to variations of the illumination conditions in indoor environments (they can work even in the absence of illumination). Some of the first approaches making use of 3D information usually use the same descriptors and classification techniques as the ones applied to color information. In [54], an LBP variation for 3D face recognition is proposed, and in [44] an LBP-based solution for the recognition of facial expressions using 3D curvature information is used (see Figure 3.7). Nonetheless, these works have two main problems. The first one is that they propose slight variations of existing feature descriptors designed for color imagery, and therefore they do not fully exploit the specific characteristics of depth information. The second one is the restricted spatial and depth resolution of depth sensors. To alleviate these problems, some works have made use of pre-computed 3D face models to be more robust and accurate in the recognition task [55]. But, they have the disadvantage that the process of obtaining 3D face models is costly and complex. There are also some proposals specific to depth imagery [54] that combines depth and normal facial information. However, all the previous approaches have in common that do not exploit all the characteristics of the depth imagery, potentially losing discriminative capability for the face recognition task.

To compensate the relatively low resolution of depth imagery, some works combine color and depth information. However, the use of depth information is merely secondary, just as a complement of the color-based recognition [56] [57] (see Figure 3.8), precisely due to the imbalanced situation between the resolution of both types of imagery. Although some of them have been specifically tested for face detection, such as the Kernel Based Local Binary Pattern (KBLBP) [58] and the Semi-Local Structure Patterns (SLSP) [59], they do not explicitly exploit the 3D structure information of a human face. In [60] color and depth imagery are combined using HOG descriptors for a user access application.

Table 3.1 summarizes different state-of-the-art algorithms for face recognition indicating

Solutions	Algorithm	Database	Imagery	Accuracy
[45]	SIFT + Naive Bayes Classifier	FG NET	Color	60%
[46]	HOG + AdaBoost	MIT + CMU	Color	95.9%
		Face Detection Data Set and Benchmark	Color	78.9%
[47]	LBP + SVM	FERET	Color	95%
		LWF	Color	78%
[48]	PCA	Not public	Color	96%
[49]	LGBP	AR	Color	96%
		FERET	Color	94%
[50]	LPQ + Nearest Neighborh	CMU PIE	Color	98.6%
		FRGC	Color	74.5%
[51]	BSIF	FRGC	Color	75%
[47]	DCT + LBP + SVM	FERET	Color	95%
		LFW	Color	78%
[53]	LBP + NN	FERET	Color	92.8%
[54]	LBP + Nearest Neighborh	BJUT-3D	3D	92.2%
		FRGC v2.0	3D	94.9%
[44]	CLBP + Chi-square	Bosphorus	3D	80.3%
[55]	Curvelet local features	FRGC v2.0	3D	94.4%
[56]	Active Shape Model	CMU-PIE	Color + 3D	85.1%
		FERET	Color + 3D	98.5%
		CAS-PEAL	Color + 3D	95.7%
[57]	Extended LBP pyramid	ORL	Color	87.7%
[58]	Kernel based LBP + SVM	ORL	Color	86%
[59]	SLSP + Adaboost	MIT+CMU	Color	91.1%
		Fddb	Color	76.9%
[60]	HOG	Not public	Color + 3D	75%

Table 3.1: Details of different state-of-the-art solutions for face recognition.

the used algorithm, the database, the kind of input visual information and the mean accuracy of the proposed solution.

3.4 Hand gesture recognition

3.4.1 Introduction

Figure 3.9: Some key application for vision-based hand gesture recognition, extracted from. a) Desktop computing interaction using hand gestures. b) Augmented reality based interaction using hand gestures. c) A robotic learning using hand gestures. d) Virtual reality based gaming using hand gestures. e) Hand gesture for interaction with large screen displays. f) Surgeons interacting with computing devices using hand gestures in medical environment.

Along the last years, with the advent of the devices presented in Section 3.2, many are the works which have presented HMI solutions in a variety of scenarios. They implement hand gesture recognition systems based on either visual or skeletal information. Some applications can be seen in Figure 3.9. Most of them use visual information, which is more complex, but also more discriminative. Moreover, the precision of the skeletal information can be quite noisy and not really accurate. The main problem related to hand gesture recognition by using either skeletal or visual information is the large variance in appearance of the different gestures that are needed to be recognized. Other problems more related with the use of visual information are the camera view point, the large number of degrees of freedom of the

hand movements, different scales, and different resolutions of the visual information [61]. As an alternative to the use of visual sensors, there are works that have proposed to solve the interaction problem by using gloves or accelerometers attached to the hand. These kind of solutions are, in general, more accurate than visual information, but they have the problem that they are invasive, and not applicable to specific scenarios where a user needs to have its hand free to perform other activities. This is the case of medical applications or HMI solutions for manufacturing certain sensitive products.

Most of the works can be divided into three main stages: detection, tracking, and recognition. Detection is the stage that locates the hand in the whole image. Tracking is the stage that estimates the hand location over the time, i.e. describes the hand movement. Finally, recognition is the stage which identifies the kind of gesture that has been done during a certain period of time. On the other hand, hand gestures can be divided into two main groups according to the movement of the hand: dynamics (moving ones) and statics (the pose of the hand do not vary over the time). Dynamic ones are more complex and can be further divided into those that are made without an intention, the unconscious ones, and those that are made intentionally, the conscious ones. The second group is the most interesting because they are used to communicate or to interact with other people or devices [62]. Along the following sections, different approaches to solve the hand gesture recognition problem (by using static, dynamic, or a combination of both static and dynamic gestures) are introduced. In Section 3.4.2 different solutions that use visual information are described, then, Section 3.4.3 summarizes some solutions that use skeletal information, and finally, in Section 3.4.4, some solutions which make use of other non-visual sensors are presented.

3.4.2 Vision-based solutions

In this section, state-of-the-art works have been divided according to the application domain.

In the literature, many are the works that have proposed the use of visual information to solve the hand gesture recognition problem. Solutions in the field of HMI have considerably increased in the last years, specially those that face the problem of interacting with robots. This is the case of the work presented in [63] that allows a visitor of a museum to interact by direct vision with a robot which provides useful information. Other solutions have been proposed to control the payload of a robot that is placed in a dangerous environment, allowing to see the whole scene from a safe place. For example, in [64] and [65], they use the combination of color and depth information to segment the hand in color imagery, and then compute a feature descriptor by a combination of Fourier and cosine descriptors. In [66], a solution for managing a rescue robot using hand gesture recognition is proposed, they use color and depth information from a Kinect v1 sensor to control the movements of a robot by

Figure 3.10: A typical application scenario: The driver interacts with the infotainment system. The ToF-sensor captures the sensitive VOI marked in red. Extracted from [69].

means of static and dynamic gestures. To describe static gestures, they estimate a polygonal hand representation from the detected fingertips. For dynamic ones, they use Dynamic Time Warping (DTW) algorithm to define a gesture as a set of poses in temporal order. Other works consider the robot control as a fully-immersive interaction where a user controls the movement of the hands and arms of the robot [67]. In [68], a recognition system based on the hand shape and its motion, combined with a Hidden Markov Model (HMM), is proposed to control the robot.

In the field of vehicle interface applications, the aim is to minimize distractions while driving [70]. A solution to interact with different onboard systems of a car by using a ToF camera placed on the car dashboard (see Figure 3.10) is presented in [69]. A point cloud is obtained by the ToF that is then reduced by a PCA algorithm. Next, a descriptor is computed from the hand geometry and the normals of the point cloud, and finally, a cascade of Multi Layer Perceptrons (MLP) are used for classification purposes.

Sign language recognition is another prominent field for hand gesture recognition. In [71] and [72], depth imagery is used to recognize among different American Sign Language (ASL) hand gestures. In [73], they use color information to distinguish among different hand gestures of the ASL. In other cases, intensity values are used directly as input to a classifier based on a Deep Belief Network (DBN) [74]. The shape of the hand, along with the relationship among different parts of the hand, have been also used as features to be delivered to an SVM classifier to perform the gesture recognition [75]. Other solution describes the hand gesture information using the output of a bank of Gabor filters, which are then delivered to a random forest classifier [72].

Hand gesture recognition has also been applied in the area of control and management

Figure 3.11: (Top) A user doing the take-off gesture with a badminton racquet as the take-off pad. (Bottom left) Making a raise gesture. (Bottom right) Making a come gesture. Extracted from [76].

of UAVs. In this line, the work presented in [76] has focused on the control of a flying robot by means of a gesture-based interface, which tries to simulate the type of interaction that humans have with birds, specifically falconneering (see Figure 3.11). Other works are more concern in commanding teams of UAVs using hand gestures [77], multimodal interfaces [78], or artificial cognition [79].

Another field of application is the direct interaction between a human and a computer. For example, in [80], a method that makes use of different parts of the fist is used to make possible an interaction between the user and the computer. They firstly compute a SIFT feature vector to detect significant points of the fist, and then, the gesture recognition is performed by detecting the rotation and translation of those points over the time. There exist some solutions which are oriented for desktop applications replacing the interaction provided by the mouse or the keyboard for different tasks, such as virtual object manipulation [81], or edition of documents using pen-based gestures [82]. Other solutions propose to use hand gestures in substitution of the mouse for web browsing [83], or to respond directly to dialogue boxes [84]. A solution able to replace the mouse is presented in [85], where color and depth information extracted from the Creative Senz3D device are combined to detect and track the palm center and the position of the fingertips. In [86], a solution to substitute the mouse by a hand gesture recognition system is proposed, where a webcam is used to obtain color images. A skin-based segmentation is used to detect the hand, and finally a Jarvis March algorithm is used to find the convex hull points of the hand that represent different gestures. A system that interacts with different computer applications, such as Google Earth, video

Figure 3.12: Non-contact mouse system setup. Left, configuration in a minimally invasive surgery operating room. Right, preliminary operating room testing (Inselspital, Bern). Extracted from [88].

games, or to control the Windows cursor is introduced in [87]. Color images are obtained by a webcam and a Kinect v1 device. The hand position is obtained by combining background subtraction in both color and depth images, then the hand contour is detected. Next, the fingertip positions and orientation are extracted, and finally, a simple finite state machine is used to determine the performed hand gesture.

Some other solutions have applied hand gesture recognition for medical purposes. The solution presented in [88] allows to perform standard mouse functions with hand gestures for surgery applications (Figure 3.12). There are other solutions which use deep learning for the recognition task, such as the work presented in [89] that introduces an interaction system to be used in sterile environments, such as operating rooms. Images are acquired from the Kinect v1 device, and the provided skeletal information is used to perform the hand region segmentation. In [90], a system that allows a blind user to write a text in a computer is introduced, then an audio feedback is used to check the correctness of the introduced command. For this purpose, a new algorithm able to extract image features and distinguish among different gestures, called the Reduced Shape Signature (RSS) algorithm, has been specially designed.

Other field is the entertainment industry with applications such as computer games [91]. For example, the Eye Toy of PlayStation 2 is a camera that tracks the hand movements for interactive games [92].

With the advent of virtual reality helmets and augmented reality glasses, new works have made use of the cameras mounted on these devices for hand gesture recognition. These works use an egocentric perspective, trying to recognize the hand interaction from a first-person point of view. For example, the solution presented in [94] makes use of the camera mounted on an augmented reality device with the aim of detecting the hands of two persons who are

Figure 3.13: An example of a hand model on the left, and two different scenario applications of the work presented in [93].

interacting. They also recognize the performed activities. For the detection process, they use a Convolutional Neural Network (CNN) to detect a region where the hand is placed. Then, a background removal process based on a Markov Random Field (MRF) is applied to select the hand region. Finally, another CNN is used with the aim of detecting the performed activity. In [95], a study of how people use their hands in daily life to interact with different objects is proposed. For the hand detection, a pixel level process based on color and texture distribution is used, and then, a HOG feature is computed over the detected region with the aim of characterizing it. Another way of analyzing the daily human object interaction is presented in [96]. In this work, they propose to detect and track the human hands by means of a pixel-level solution that uses the color information provided by an ego-centric camera. For the hand modeling purpose, they use a combination of both local and global appearance information. For the local information, they combine Gabor filters, SIFT, and HOG descriptors to characterize the hand region. For the global information, they use a regression-based solution in which they take into account the illumination variations, and also the global appearance through an HSV color space histogram. Other applications are those in the field of virtual reality to enable more realistic interaction with objects in the virtual space using your own hands to simulate a 3D immersion on a 2D display [97]. In some cases, hand gestures are applied to navigate into a 3D virtual space, while the user is touching some virtual objects [98]. In other cases, hand gesture recognition is used to interact within smart room environments, combining the use of different devices to interact with light controllers [99] or large screen displays [100]. Another important application is the possibility of allowing different users in different places to interact simultaneously with the same virtual environment, which is known as Computer Supported Collaborative Work (CSCW) using tabletop or large screen displays [101]. There are other innovative solutions that allow a collaborative interaction between the instructor and the students of a course [102]. Another interesting work is the one presented in [93], where a hand tracker uses a smooth model of the hand, allowing a second-order optimizer to jointly search over the high-dimensional space that contains both the hand pose and the model positions that correspond to each

depth image pixel. Despite a higher per-iteration cost, this optimization has a wide basin of convergence and yields a real-time system that is both more accurate and more efficient than alternatives, enabling new interactions in virtual reality. A Kinect v2 device is used as acquiring device, as shown in Figure 3.13. A solution that uses infrared images to interact within an immersive environment is proposed in [103], where the hand contour is extracted by using a variant of the dynamic programming algorithm. Once the contour is extracted, depth values of the hand are obtained by combining a linear interpolation with a refinement on the heat diffusion model.

There are also many works addressing gesture recognition without a specific purpose. This is the case of the work presented in [104]. This work segments the hand regions to obtain a feature vector by means of a Bag-of-Words strategy and Speeded Up Robust Features (SURF). Then, a Radial Basis Function Neural Network (RBFNN) predicts the performed hand gesture. Other solutions make use of Fourier descriptors with Recurrent Neural Networks (RNN) [105]. In [106], they use a combination of Gabor features with SVM classifiers to perform hand gesture recognition under varying illumination scenes. Other techniques propose the use of a depth threshold to only make the detection of the nearest object, assuming that the nearest object should be the hand [107][108]. Other solutions propose the use of different algorithms to reduce the data dimensionality and develop more efficient hand gesture recognition solutions. It has been applied to color information [109], and depth information [110]. In many cases, they propose to generate a dictionary of gestures using a bag of features representation [111], and then perform the recognition task by posing a l_1 -minimization problem. In other cases, they use Compressive Sensing (CS) to directly recognize the gesture, using also a bag of features scheme [112]. Alternatively to CS for the dimensionality reduction task, other works have used the Singular Value Decomposition (SVD) technique [113], or even the combination of CS with PCA [114]. We can also find some techniques that simultaneously make use of depth and color information to minimize the errors of both kinds of information [115]. For example, in [116] the authors propose a solution which uses different CNNs to make a final pose prediction. This is a combination of two predictions, one based on the original image and another based on a synthesized image. In a first stage, the pose of the hand is predicted using the 3D location of the hand joints. Next, a trained model is used to fit the visual data with the precomputed model. Finally, an update is made over the model with the new input image. In the work presented in [117], they use a hierarchical sampling optimization method with the aim of inferring the hand pose. In the work [118], they present a solution to predict the hand position using depth images. Additionally, they introduce a new way to generate a hand model using depth imagery. The shape of the hand model is parametrized as a linear combination of a mesh in a neutral pose and a small number of offset vectors. The work presented in [119]

Solutions	Algorithm	Database	Imagery	Accuracy
[64]	Cosine-descriptors + ANN	Not public	Color + depth	93.4%
[71]	Rotation, translation, scale invariant feature + MLRF	ASL Finger Spelling Dataset	Depth	85%
[72]	Hand shape features + RF	ASL Finger Spelling Dataset	Color + depth	75%
[73]	Hu moments, elliptic fourier, RDF, LBP + HMM	Not public	Color	78%
[74]	Intensity-based features + DBN	ASL Finger Spelling Dataset	Color + depth	78%
[75]	Rotation invariant descriptor + SVM	NTU	Color + depth	97.1%
		ASL Finger Spelling Dataset	Color + depth	96.2%
[89]	CNN	Not public	Color	83.1%
[90]	RSS	Not public	Color	97.5%
[97]	Color-based features + HMM	Not public	Infrared	95%
[104]	SURF + RBF	Not public	Infrared	99.5%
[106]	Gabor features + SVM	Not public	Color	86.4%
[114]	SAE-PCA	ASL Finger Spelling Dataset	Color + depth	99%

Table 3.2: Details of different state-of-the-art solutions for static hand gesture recognition solutions using visual information.

introduces a solution that performs both hand detection and hand tracking processes using depth information. In [120], they introduce a hand pose estimation solution based on the use of a 3D cascade object pose regression scheme. Another solution is the one presented in [121], where they use a Superpixel-Markov Random Field (SMRF) algorithm to model different parts of the hand. In [122], a variant of the Convex Invariant Position Based on RANSAC (CIPBR) algorithm is used for feature extraction, and then a DTW algorithm is used for classification, which is refined by the use of an HMM algorithm.

Table 3.2 and Table 3.3 respectively summarize different state-of-the-art algorithms for static and dynamic hand gesture recognition, indicating the kind of visual information, the used algorithm, the database, and the mean accuracy of the proposed solution.

Solutions	Algorithm	Database	Imagery	Accuracy
[66]	Contour-based features + DTW	Not public	Color + depth	100%
[68]	Contour-based features + Kalman filter + HMM	Not public	Color	81.7%
[69]	PCA + point cloud descirptor + MLP	Not public	Depth	82.2%
[85]	Contour based feature extraction and classification	Not public	Color + depth	96.5%
[86]	Contour based features + Haar	Not public	Color + depth	92.2%
[103]	Contour-based features + SVM	Not public	Color + depth	97.5%
[105]	Fourier descriptors + HMM + RNN	Not public	Color	91.9%
[109]	Color-based features + Sparse Representation-based Classifier	Palm Grafti Digits	Color	100%
[110]	4DCov + Sparse Collaborative Classifier	Gesture3D	Depth	92.8%
		SKIG	Depth	93.8%
[112]	3D EMoSIFT + DTW	ChaLearn gesture database	Color + depth	86.7%
[122]	CIPBR + DTW + HMM	MSRGesture3D	Depth	97.4%
		RPPDI	Depth	97.4%

Table 3.3: Details of different state-of-the-art solutions for dynamic hand gesture recognition solutions using visual information.

Figure 3.14: CAD Interaction Laboratory demonstration. Extracted from [126].

3.4.3 Skeletal-based solutions

Similarly to the vision-based solutions, the state-of-the-art works proposing skeletal-based solutions have been divided according to its specific application.

Many works have used skeletal information to recognize hand gestures. The main reasons are that the number of devices providing this kind of information, and also the increasing robustness and accuracy of the estimated skeletons along the last years. Solutions using skeletal information have been applied in different fields solving a broad variety of problems. Some of them have been applied to driver assistance, where hand gestures have been used to control the radio via a near-infrared camera sensor that provides skeletal information [123].

Other solutions have proposed a system to perform a direct interaction with robots. This is the case of the work introduced in [124], where they obtain a feature vector based on different relationships among skeletal joints by a combination of 3D Histograms of Oriented Optical Flow (HOOOF) and Histogram of Oriented Trajectories (HOT). In [125], trajectories from both hand skeletons are used to control dual robots simultaneously.

Sign language recognition is one important field of application of such systems. The work presented in [43] uses several Leap Motion devices to fuse their skeletal information using an HMM, and thus achieving a higher recognition in the ASL. In [127], a feature vector based on the computation of different distances and angles between the fingertips and the palm position are delivered to an SVM classifier. An Artificial Neural Network (ANN) is used in [128] to be able to distinguish among different hand gestures, represented by a feature vector that computes the Euclidean distance between consecutive fingers and the Euclidean distance among fingertips positions.

The Leap Motion is used to interact within a virtual reality environment in combination with an Oculus Rift device in [126], as it is shown in Figure 3.14. A hand tracking algorithm is used to know the hand position and the fingers movements to make possible the HMI. In [129], an on-air text recognition application drawn by the fingers is proposed

Figure 3.15: A user is writing text in 3D space using his/her fingertip over the Leap Motion sensor. The corresponding 2D text is appeared in the computer screen. Extracted from [129].

(see Figure 3.15), which combines HMM and Bidirectional Long Short-Term Memory Neural Networks (BLSTM-NNs). A 3D air-writing interface is presented in [130], where the fingertips are tracked to describe the word written in the air, and then, an HMM algorithm is used to determine which word of the vocabulary has been written.

Another group of solutions have proposed to use skeletal information in the medical domain. For example, in [131], a feature vector composed by Euclidean distances among fingertips, pitch, roll, and yaw angle of the palm, and angles among fingers is extracted. For the classification stage, two different solutions are proposed: a multiclass SVM and a k-Nearest Neighbor (k-NN). Other solutions have also proposed to use Leap Motion information for rehabilitation purposes. In [132], a feature vector is computed using the raw information of the Leap Motion and some angles between different joints. Then, it is analyzed by a set of different classification processes, as decision trees, k-NN, and SVM. An application for the coronary heart disease preoperative is introduced in [133]. Gestures are characterized by the position and the velocity of different parts of the hand, and then a K-means algorithm is used to select the representative features of each gesture. Finally, the recognition process is performed by an HMM algorithm.

Other works do not have a specific application domain. An example is [134], where a hand feature descriptor is obtained by computing different dynamic hand gestures. The trajectories of the hand skeleton nodes are used to compute a robust descriptor, which is then delivered to a Hidden Conditional Neural Field (HCNF) classifier to recognize the gestures. The obtained trajectories are filtered by an Interval Kalman Filter (IKF) and an

Figure 3.16: Acquisition setup extracted from [136].

Improved Particle Filter (IPF), leading to a better estimation of the hand position. In [42], the signature of a person is proposed to check his identity by processing skeleton-derived trajectories. For this purpose, a combination of 3D HOOFF and HOT are used as input feature vectors to an SVM classifier. The solution proposed in [135] uses the raw features provided by the Leap Motion combined with an SVM classification algorithm. In [41], they use the trajectory of the palm to obtain the feature vector combined with an SVM solution for recognizing the palm trajectory describing the shape of a number.

Some works have also proposed to complement the skeletal information of the Leap Motion with imagery acquired by other sensors, with the purpose of avoiding to process the highly distorted infrared imagery of the Leap Motion. Depth imagery provided by the Kinect v1 and skeletal information acquired by the Leap Motion (see Figure 3.16) are used in [136] to compute two feature vectors, which are then delivered to a multi-class SVM classifier. In [137], color and depth information extracted from the Kinect v1 sensor are combined with the skeletal information provided by the Leap Motion. A CNN is used as feature extractor for both, depth and color imagery. Handcrafted features based on distances among fingertips are used to describe the skeletal information, and an SVM based solution is used for the classification step. Another work [138] uses a Coupled Hidden Markov Model (CHMM) to classify the skeleton information coming from both Leap Motion and Kinect v1. The Leap Motion has been also used in combination with non-visual sensors. For example, the Leap Motion skeleton is fused with the information provided by the accelerometers of a wearable watch in [139].

Table 3.4 summarizes different state-of-the-art algorithms for hand gesture recognition using skeletal information indicating the used algorithm, the database, the kind of gestures (static or dynamic), and the mean accuracy of the proposed solution.

Solutions	Algorithm	Database	Gestures	Accuracy
[124]	Feature using angles between hand skeleton parts + Random Forest	Not public	static	94.2%
[43]	Features from hand skeleton + HMM	Not public	static	93.1%
[127]	Distances among hand skeleton parts + SVM	Not public	static	79.8%
[128]	Distances among hand skeleton parts + MLP	Not public	static	100%
[131]	Distances among hand skeleton parts + k-NN	Not public	dynamic	97.7%
[132]	Features from hand skeleton + HMM	Not public	static	99.9%
[133]	Features from hand skeleton + HMM	Not public	dynamic	97.8%
[134]	Features from hand skeleton + HCNF	LeapMotion-Gesture3D	dynamic	89.5%
		Handicraft-Gesture	dynamic	95%
[42]	HOOF & HOT + SVM	IIITD LS	dynamic	66.84%
[41]	Features from hand skeleton + SVM	Not public	dynamic	91.3%
[137]	Features from hand skeleton + SVM	Not public	static	82%

Table 3.4: Details of different state-of-the-art solutions for hand gesture recognition using skeletal information.

Figure 3.17: The gesture recognition setup using a magnetic 3D position tracker extracted from [143].

3.4.4 Other devices solutions

Some works have used other information, instead of the visual one, for the task of hand gesture recognition. This is the case of the solution presented in [140], where they use an accelerometer to obtain 3D information of the hand movement with the aim of controlling an entertainment robot. Hand gesture recognition can also be used in rehabilitation processes by using resistive sensors placed on the fingers that can be used to simulate the pressure distributions on the forearm, and reproduce the movement on an arm articulated model [141]. In [142], a continuous hand gesture recognition system, suitable for a low-cost and energy-efficient HMI, is presented. The gesture characterization is made with the data from the accelerometer and the gyroscope sensors. And then, a DTW solution is used for gesture classification, and a gesture spotting process is used to distinguish between a database gesture and an involuntary gesture. A magnetic 3D position tracker attached to the user's hand (see Figure 3.17) is employed in [143]. Gestures are characterized according to the xy-plane trajectory, and a finite state machine is used to distinguish among different gestures. A combination of the information provided by a surface electromyography (sEMG) and a near infrared spectroscopy (NIRS) is presented in [144]. A combination of both signals produced by the hand movement is used. Other sensors are the data gloves [145] that can recognize the movements of a hand. That kind of sensors have as a main inconvenient the difficult calibration process and the high cost. Besides, special hardware is required. A solution based on the use of a glove with inertial sensors is proposed in [146] to provide feedback of the performed gesture.

Other works have proposed to combine information from a sensor and from a camera to perform the interaction. In [147], a glove with different optical emitters, shown in Fig-

Figure 3.18: Description of the functional components of GoldFinger smart glove extracted from [147].

ure 3.18, is used to send a signal that is received by a camera, which interprets the gesture and makes an interaction with a robot. The solution presented in [148] combines the visual information provided by the Leap Motion with a flex sensor for an HMI application. The fingertips positions are obtained from the flex sensor and both fingertips positions and fingertips velocities are acquired from the Leap Motion sensor. A combination of the information provided by both sensors is then used for a tracking purpose by means of a Kalman filter based algorithm.

3.5 Face and hand gesture recognition

There are some works that try to solve both problems at the same time, using different kinds of visual information. Along this chapter, different solutions are described divided according to their application domain.

In the field of sign language interpretation, there are some scenarios where the recognition of both hand and face is important. The solution presented in [149] uses color images to recognize the performed gesture. First, they segment both hands and face by a skin color segmentation scheme, then a feature extraction based on motion information is perform, and finally, an HMM is used for classification purposes. A solution based on the use of spatial saliency maps from color images is presented in [150] for sign language recognition. A solution combining information from the Kinect v1 and the Leap Motion is introduced in [151]. They propose an ASL recognition system combining face and hand information using an HMM solution in conjunction with an independent Bayesian classifier algorithm. An arabic sign language recognition system is presented in [152]. A Recursive Principle Components (RPCA) for feature extraction is used along with an MLP for distinguishing among different signs.

Some solutions are oriented to interact with a robot. This is the case of the work presented in [153], where a Kinect v1 sensor mounted on a robot is used to detect human

Figure 3.19: Face-hand feature detection, extracted from [156]. (a) Original image; (b) convert to YCrCb image; (c) face and hands detection; (d) binary image; (e) image after median filter; (f) face-hand regional centers and bounding rectangles.

faces using the Viola-Jones algorithm. Then, a Kalman filter is used to track the position of both human hands. In [154] a solution to recognize different human activities is proposed based on the combination of face and hands movements. Firstly, they determine the pose at each frame by a heuristic combination of different human parts. Then, they use an SVM to determine the activity, followed by a temporal study to determine the performed activity. A robot interaction system is presented in [155]. In that work, the face and hand detection is made using a skin color-based segmentation algorithm, and then a Least Square Matching (LSM) algorithm is used to determine the hand position.

Another field of application is the analysis of different patterns of human behaviors, such as the case of the work introduced in [156], where they use a stereo pair to detect both hands and face to determine the human behavior. They use a color skin-based segmentation as it is shown in Figure 3.19. Then, they determine the performed action according to the movements of different human parts. An activity detector combining hands and head information is presented in [157], where they extract features from hand, head, and human body skeleton, and then an SVM is used to distinguish among different actions.

An automatic control system is presented in [158] based on the combination of color and infrared images. The proposed method is based on the use of a CNN algorithm.

In other cases, a solution to detect both human body parts without a specific application

domain is proposed. This is the case of [159], where the hand is detected based on the face position. First of all, the face is detected using a Viola-Jones algorithm, and then, the skin color of the face is used to detect the hand through a color segmentation algorithm.

3.6 Conclusions

Many are the state-of-the-art solutions that tries to solve both problems: face and hand gesture recognition. The efforts to solve both problems indicate the relevance of these tasks. Moreover, most of the works use visual information, although the problem is not yet solved. Therefore, new solutions should be found, trying to obtain better performance.

In the case of the face recognition problem, color imagery is the most used, while depth one is mainly utilized as a complementary source of information. This fact has motivated that in this thesis some new feature descriptors specially designed for depth images have been developed. These new feature vectors can distinguish the face of different subjects in different orientations (not only frontal ones).

For the hand gesture recognition, there exist a vast variety of ways to solve the problem, and an even bigger group of application domains. Solutions based on the use of visual or skeletal information are more natural and friendly than the ones based on gloves or accelerometers, since the user is not forced to wear any special device. For this thesis, the decision about using visual information, instead of skeletal one, has been made due to the fact that visual information (color, depth or infrared imagery) acquired by different sensors is not so dependent on the device as the skeletal information provided by different devices (devices can provide a different number of joints or different configurations). Moreover, skeletal information can be a very simplistic model of the visual data provided by cameras. And therefore, using the original and rich information encoded by the images can be the key factor to boost the hand gesture recognition performance. Taking into account all this and also considering that the hand gesture recognition is still a problem to be solved, different hand gesture recognition solutions by using visual information from different devices have been developed along this thesis, trying to reach a satisfactory solution.

Chapter 4

Vision-based Human-Machine Interaction

4.1 Introduction

In the last few years, many solutions for both face and hand gesture recognition have been proposed in the computer vision field. Along this chapter different solutions to solve those problems are described, introducing some contributions oriented to use depth imagery to solve both face and hand gesture recognition problems. It is also explained the introduction of a Compressive Sensing framework to reduce the high dimensionality of the designed feature descriptors. Finally, solutions that can recognize both static and dynamic gestures in real-time are described.

4.2 Main system description

The designed system is a solution to allow a more natural and intuitive HMI based on the use of vision-based sensors. It is composed by two main modules: face and gesture recognition. Face recognition identifies a subject who wants to perform the interaction, while hand gesture recognition interprets the interaction. In the case of hand gesture recognition solutions, two scenarios are considered. One scenario in which real-time performance is not so demanding, but the robustness of the system is really important (see Section 4.4.1), and a second scenario in which real-time performance is more relevant (see Section 4.4.2).

A block diagram of the proposed system is shown in Figure 4.1. The face recognition module, which acts as a robust access control system, begins to acquire images from the camera and tries to recognize if the face belongs to a known subject. In case that the subject is unknown, the system blocks the hand gesture recognition process. Otherwise,

Figure 4.1: Blocks diagram of the proposed system.

the system enables the hand gesture recognition module during a period of time trying to recognize the performed hand gesture, either static or dynamic. If one gesture is recognized an event/action is generated.

4.3 Face recognition module

The face recognition module analyzes depth images provided by the Kinect v2 [9] device. For this purpose, three different novel descriptors have been designed, which fully exploit the characteristics of depth imagery: the Depth Local Quantized Pattern (DLQP), introduced in Section 4.3.1, the Depth Spatiogram of Local Quantized Patterns (DS-LQP), presented in Section 4.3.2, and the Bag of Dense Derivative Depth Patterns (BAG-D3P), introduced in Section 4.3.3. These descriptors take advantage of the depth information since the relative value between adjacent pixels on a face depth map is quite constant independently of the distance between the face and the sensor.

For every acquired depth map, a feature descriptor is computed by one of the three designed descriptors, which is then delivered to an SVM classifier (see Section 4.3.4). If the output of the SVM indicates that the subject is known the hand gesture recognition system is enabled.

4.3.1 Depth Local Quantized Pattern (DLQP)

Inspired by the success of the LBP [160] recognizing human faces in color imagery, the DLQP novel descriptor, specifically adapted to high depth resolution information, has been designed, resulting in a publication [1]. The computation of the DLQP can be divided into two stages: quantification of local depth differences and computation of histograms of quantized codes.

Figure 4.2: DLQP descriptor. $Q(x)$ represents the quantification of the x different value, e.g.: $x = -18$ belong to Bin1, so $Q(-18) = 1$ represented with 3 bits, i.e.: 001.

4.3.1.1 Quantification of local depth differences

The main modification with respect to the LBP occurs in the quantification process: differences between the depth value of the central pixel and the neighborhood ones are quantized with more than one bit (instead of a bit quantization), making richer the characterization of the depth information, and increasing the capacity of distinguishing depth patterns. Notice that this strategy is plausible because, unlike color imagery, depth values are not so affected by illumination changes. The quantification process has been adapted for the purpose of face characterization and recognition, adapting the range of depth differences to quantify. At the same time, the DLQP-based feature vector is kept short enough to be used in practice. For this purpose, $N_{int} = 8$ symmetric intervals of different widths are used: two intervals $(-10, 0)$ and $(0, 10)$ mm, representing small random depth variations (usually due to noise); four intervals $(-30, -20)$, $(-20, -10)$, $(10, 20)$, and $(20, 30)$ mm that encode the most representative depth reliefs of a human face; and two open end intervals, $(-\infty, -30)$ and $(+30, +\infty)$ mm, which represent the large depth variations usually corresponding to object borders (such as face to background). The decision values of the previous intervals are adapted to both capture the most discriminative depth information of a human face, and exploit the bit depth (bits per pixel) of depth sensors. This fact contributes to dramatically increase the capability of distinguishing different depth-based patterns, and ultimately the target human face. For the quantization of each interval, $N_b = \log(N_{int}) = 3$ bits are needed to encode every quantized depth difference. In addition, only 4 neighbors (west, north, east, and south) are considered to keep the DLQP descriptors with a reasonable size.

More formally, let it be (x_c, y_c) the coordinates of a pixel in the image, the computation of a DLQP code can be expressed as follows:

Figure 4.3: DLQP-based feature descriptor combined with the division approach.

$$DLQP_{P,R}(x_c, y_c) = \text{stack}_{P=0}^{P-1} q(d_p - d_c), \quad (4.1)$$

where R is the radius of the neighborhood that has P surrounding pixels, d_c and d_p are the depth values of the central pixel and one of the P pixels of the neighborhood, respectively. The function $q(d_p - d_c)$ computes the quantification of the difference between the depth values d_p and d_c , and stack is a function that simply concatenates all the binary codes resulting from the neighborhood difference quantification.

4.3.1.2 Computation of histograms of quantized codes

Once a DLQP code (equation 4.1) is computed for every pixel of the depth-based face image, the next step is to obtain a descriptor, in this case a histogram, with all of them. As it is shown in Figure 4.2, every DLQP code contributes to the bin of the histogram with its correspondence value. The size of the histogram is determined by the number of bits used in the quantification process (N_b), and the number of surrounding pixels (P). In this case, $N_b = 3$ and $P = 4$, resulting in a DLQP histogram of size:

$$DLQP_{hist} = 2^{N_b \times P} = 2^{4 \times 3} = 4,096. \quad (4.2)$$

To increase the relevance of the spatial information, which is lost with the histogram approach, every depth-based face image is divided into $N_s \times N_s$ non-overlapping blocks. In this case, $N_s = 4$ has been chosen so the image is divided into 16 non-overlapping blocks of equal size, and then a different DLQP histogram is computed for each one. Finally, the DLQP-based feature descriptor is obtained by concatenating all the histograms in a single

Figure 4.4: Stages of the DS-LQP descriptor.

vector. Figure 4.3 summarizes the process. The dimension of the DLQP descriptor can be expressed as follows:

$$DLQP_{desc} = DLQP_{hist} \times N_s \times N_s = 4,096 \times 4 \times 4 = 65,536. \quad (4.3)$$

4.3.2 Depth Spatiogram of Local Quantized Patterns (DS-LQP)

Inspired by the previously introduced DLQP feature descriptor, a modification is proposed, called Depth Spatiogram of Local Quantized Patterns (DS-LQP) (resulting in a publication [2]), which introduces more spatial information to the final descriptor. The computation of the DS-LQP can be divided into 3 stages (see Figure 4.4): quantification of local depth differences, computation of histograms of quantized codes, and computation of spatiograms of bins of quantized codes. Since the two first stages are the same as the ones described for the DLQP descriptor, only the third stage will be described.

4.3.2.1 Computation of spatiograms of bins of quantized codes

This final stage, which introduces the difference with respect to the DLQP feature descriptor, adds spatial information to the DLQP descriptor computed over each image region. To that end, each image block is subdivided into 4 sub-blocks as it is shown in Figure 4.5. Additionally to the DLQP feature vector computed for every image region, for each sub-block, one accumulative vector is computed that encodes the spatial distribution of the DLQP bins over the block with respect to the central pixel of the corresponding sub-block. The computation of each vector (H_{esp}) for every sub-block is described graphically in Figure 4.6; the value with which each pixel of the block weights to the vector of each sub-block is inversely proportional to the distance between the pixel and the central pixel of the block. That weight is just added to the vector element that corresponds with the DLQP value that this pixel has. Figure 4.6 shows an example of a pixel which DLQP value is 134, so when that pixel is being evaluated, a weight which is inverse to the distance to the central pixel of each sub-block is added to the element 134 of each vector for every sub-block.

Figure 4.5: DSLQP-based feature descriptor computation.

With this configuration, each image region is characterized by 5 histograms, the one that corresponds to the DLQP feature vector and the other 4 accumulative vectors that describe its spatial distribution along the block as it is shown in Figure 4.5. From that figure it can also be extracted that the final DS-LQP feature descriptor is a stack of all the feature descriptors for each image block which are composed by the DLQP feature vector and the 4 H_{esp} accumulative vector computed for each sub-block. As each H_{esp} vector, and each DLQP feature vector has a length of 2^{12} elements, and every depth-based face image is divided into $N_s \times N_s$ non-overlapping blocks of equal size, the dimension of the final DS-LQP descriptor can be expressed as follows, where $N_s = 4$ has been chosen as in the previous case:

$$DS-LQP_{desc} = (DLQP_{hist} + (4 \times H_{esp})) \times N_s \times N_s = (4,096 + (4 \times 4,096)) \times 4 \times 4 = 327,680$$

 Figure 4.6: H_{esp} histogram computation for an image region.

Figure 4.7: Block diagram of BAG-D3P feature descriptor.

4.3.3 Bag of Dense Derivative Depth Patterns (BAG-D3P)

The Bag-D3P, resulting in a publication [3], is a novel feature descriptor that fully exploits the characteristics of depth imagery taking advantage of the extended spatial and depth resolution of depth sensors (in comparison with color cameras), and is adapted to maximize the discrimination of human faces. The computation of the BAG-D3P can be divided into four stages (see Figure 4.7): dense spatial derivatives, non-uniform quantization and codification, multi-bag of words, and spatial block division.

4.3.3.1 Dense spatial derivatives

The first stage characterizes each pixel in a depth image region by densely computing all the pairwise pixel differences in a pixel neighborhood. From a signal processing point of view, these differences represent a bank of 2D derivative filters at multiple scales and orientations. Figure 4.8 illustrates this process for a neighborhood of 9 pixels. As a result, a vector of dense spatial derivatives values is obtained, whose length is:

Figure 4.8: First stage of Bag-D3P computation: dense spatial derivatives.

$$N_{dsd} = \frac{(P-1)^2 + (P-1)}{2}$$

where P is the number of pixels in the considered neighborhood. For example, for a neighborhood of $P = 9$, a value of $N_{dsd} = 36$ is obtained. Each value of this vector is the output of one filter of the derivative filter bank which can be expressed as follows:

$$H(n) = p(i) - p(j)$$

where $n = 0, 1, \dots, N_{dsd} - 1$; $i = 0, 1, \dots, P - 1$; $j = 0, 1, \dots, P - 1$ with $j \neq i$; $p()$ is the depth value of the depth-based image pixel; and n , i and j indices going from left to right and from top to bottom.

4.3.3.2 Non-uniform quantization and codification

The second stage uses the same scheme as the one used in the stage called quantification of local depth differences of the DLQP descriptor described in Section 4.3.1.1. The same number of intervals and the same values for their limits are considered.

4.3.3.3 Multi-bag of words processing

The third stage encodes the quantized values of the previous stage to implement a multi-bag of words scheme. Since there are $N_{int} = 8$ intervals, $N_b = \log(N_{int}) = 3$ bits are required to represent every quantized value following a fixed-length encoding strategy. Finally, all binary

Figure 4.9: Third stage of Bag-D3P computation: multi-bag of words concatenating $N_h = 12$ histograms for chunks of $N_{div} = 9$ bits.

numbers resulting from the quantization process from a pixel neighborhood are concatenated to form a binary word with the following number of bits:

$$N_{bin} = N_{dsd} \times N_b = 36 \times 3 = 108$$

The multi-bag of words is an extension of the bag of words strategy to compactly represent the set of $N_{bin} = 108$ bits words corresponding to a pixel region (notice that every pixel in the region is encoded by a 108-bit word). Without this stage, the high volume of data will be unmanageable. More specifically (see Figure 4.9), this stage starts by dividing every binary word into smaller chunks of length $N_{div} = 9$ bits, which are then converted into decimal values. Next, a histogram per chunk is computed considering all possible N_{div} bit binary words. As a result, multiple histograms are obtained, one per word chunk, concretely, considering that the 108 bits word has been divided into smaller chunks of length 9 bits, we have a total of $N_h = 108/9 = 12$ histograms. Every of those histograms are finally concatenated to form the BAG-D3P descriptor of a depth region. The final descriptor has a length of:

$$N_{dct} = N_h \times 2^{N_{div}} = 12 \times 2^9 = 6,144 \quad (4.4)$$

4.3.3.4 Spatial block division

The last stage incorporates global spatial information to the descriptor. The multi-bag of words strategy discards global spatial structure information to generate compact feature vectors at the expense to reduce the discrimination capability. However, the discrimination capability can be improved by adding some global spatial information. For this purpose, a face region is divided into $N_s \times N_s$ non-overlapping blocks, and then a BAG-D3P descriptor is computed per block. Finally, all BAG-D3P descriptors are concatenated into one vector, which is delivered to a bank of SVMs for the recognition task.

The resulting descriptor length is $N_T = N_s \times N_s \times N_{det}$, which for a face block division of 4×4 , and the previous parameter values, results in a vector of length:

$$N_T = 4 \times 4 \times 6144 = 98,304 \quad (4.5)$$

4.3.4 SVM classification process

For the three previously introduced feature descriptors, an SVM solution is proposed for the classification process. Regarding the multi-class classification process, a one-vs-all strategy has been adopted, using a binary SVM classifier per subject. Each SVM is trained with positive samples extracted from feature vectors from one specific subject, and with negative samples with feature vectors from the other subjects of the database. To improve the results, an approximation of the Hellinger kernel, more commonly known as Bhattacharyya distance [161], has been applied:

$$k(h, h') = \sum_i \sqrt{h(i)h'(i)} \quad (4.6)$$

where h and h' are the feature descriptors of the test and training samples.

As a result, the trained multi-class SVM can predict the identity of a person, allowing the use of the following hand gesture recognition module just to the known people.

4.4 Hand gesture recognition module

4.4.1 High discriminative scenario

The proposed system is oriented to achieve high accuracy and robustness values in the recognition process using the Kinect v2 [9] as depth sensor. Two different implementations have been developed, one for static gesture recognition (Section 4.4.1.1), and other for simultaneous static and dynamic gesture recognition (Section 4.4.1.2). Static hand gesture recognition solutions can be more useful in scenarios in which redundancy is needed as it is easier to use consecutive frames to check the detection of the same gesture, or even in scenarios where the required vocabulary is reduced and the gestures could be simpler, scenarios where the interaction needs to be performed repeatedly and with high frequency. In contrast, if a more complex and bigger vocabulary is needed, it is better to use a dynamic solution as a greater variety of gestures can be used. It is also a better solution in cases in which you want the system to work continuously as it is easier to distinguish between an involuntary gesture and a gesture from the database in the case of dynamic gesture recognition scenarios. For the static hand gesture recognition problem, the DLQP and the BAG-D3P descriptors have been

Figure 4.10: Block diagram of the DLQP-based hand pose recognition system.

proposed with one additional modification, the use of a Compressive Sensing (CS) strategy to reduce the length of the descriptor and also to increase the accuracy of the system. For the case of simultaneous static and dynamic hand gesture recognition, a new feature descriptor has been designed, the Array of Spatio-Temporal Histograms of Local Quantized Depth Patterns (ASTH-LQDP), which has some similarities with the DLQP and BAG-D3P descriptors used for static hand recognition.

4.4.1.1 Static gesture recognition

Hand pose recognition system combining DLQP descriptors with a CS solution

In this section, a system specifically designed to recognize among different static hand poses by means of depth information is presented (Figure 4.10). For each frame acquired from the depth sensor, a DLQP feature vector (Section 4.3.1) is computed to characterize it. This feature vector is used without a modification with respect to the one used in the case of face recognition as, in both cases, depth images are used. Both for human hands and human faces, the relevant information is more or less at the same relative distance so it is not necessary a specific design modifications. The key contribution is the introduction of a CS stage to largely reduce the dimensionality of the highly discriminative feature descriptor, which originally have an excessive and inoperative high dimension. The high dimension of the feature descriptor is a big problem for the classification stage as it is more difficult for it to find the borders among classes and, in some cases, it is unavoidable due to the high computational cost of the process. But with the application of the CS algorithm, the computational cost of the SVM classifier is considerably reduced and it is able to obtain better results as it is easier to find borders among different hand poses in a low dimensional space.

Hand pose recognition system combining BAG-D3P descriptors with a CS solution The solution depicted in Figure 4.11 represents an HMI system by using hand gesture

Figure 4.11: Block diagram of the BAG-D3P-based hand pose recognition system.

recognition. Concretely the interaction is going to be made by using different static hand gestures or hand poses. The solution makes use of the depth imagery acquired by the Kinect v2 device that provides more depth resolution. The main novelty is the introduction of a CS step to reduce the dimension of a depth-based feature descriptor, the BAG-D3P introduced in Section 4.3.3, which is a very discriminative descriptor, but also too large for its practical application. The system is composed by three steps: depth-based feature descriptor computation that robustly characterizes the hand gesture; CS-based dimensionality reduction algorithm that shortens the previous highly discriminative but also large feature vector with almost no information lost; and an SVM-based classification stage that recognizes the performed hand gestures. The introduction of the CS algorithm is really important not just to obtain better classification results but also to be able to perform a more natural and intuitive interaction. This is because the computational time is considerably reduced allowing a low time response between the gesture performance and the interactive system response.

Compressive Sensing (CS) With the purpose of reduce the high dimensionality of both, the DLQP and the BAG-D3P feature descriptors, a CS framework is proposed to characterize static hand gestures.

The CS technique allows the reconstruction of sparse signals with fewer measurements than the ones required by the Shannon-Nyquist criterion [162], allowing to highly compress the information. This process makes possible the reduction of the computational cost and the memory requirements of the recognition step, while keeping almost all the discriminative power of the DSQP and the BAG-D3P descriptors.

The CS algorithm uses a matrix ϕ of measurements with dimensions $(M \times N)$ to obtain a lower dimensionality vector y of length M from a sparse target vector x with dimension $N \gg M$ (see figure 4.12):

$$y = \phi \cdot x \quad (4.7)$$

Figure 4.12: Example of a CS process in which x is the input and sparse vector, ϕ is the transform matrix and y the output vector.

The design of the matrix ϕ is one of the keys of the CS framework. To ensure that distances among the original vectors and the reduced ones are preserved, the matrix must satisfy the Restricted Isometry Property (RIP) [163]:

$$(1 - \delta_k) \|x\|_2^2 \leq \|\phi x\|_2^2 \leq (1 + \delta_k) \|x\|_2^2 \quad (4.8)$$

where $\|x\|_2$ is the Euclidean norm of x , and δ_k is the error in the vector distances after the projection ϕ .

In the literature, it is shown that different kind of matrices satisfy the RIP property. We are going to use the Gaussian distribution matrix, which is one of the most commonly used. Thus, a matrix of size $(M \times N)$ is computed, whose elements are i.i.d. samples of a normal Gaussian distribution $N(0, 1)$. This matrix satisfies the RIP property if the following relationship is achieved:

$$M \geq c \cdot K \cdot \log(N/K), \quad (4.9)$$

where M is the length of the output vector y , N is the length of the input vector x , K is the sparsity of the vector x (i.e. the number of elements that are not zero, or close to zero), and c is a constant of value $c = 1/2 \cdot \log(\sqrt{24} + 1) \approx 0.38$ [164].

The dimensionality reduction obtained by applying the CS algorithm allows to obtain better results for those classification tasks that involves a non-convex optimization problem,

Figure 4.13: Block diagram of the ASTH-LQDP-based hand pose recognition system.

since a lower dimensionality space is considered.

The decision of using the CS algorithm, instead of other dimensionality reduction algorithms (PCA or SVD) was based on the computational cost. Both PCA and SVD algorithms need to make expensive operations using all features of all gestures. In the case of the CS algorithm, just a multiplication between each feature vector and the measurement matrix ϕ is required. This means that the CS cost is linear in contrast with the cost of the other techniques which is quadratic. Another important advantage is that CS can be applied to all kind of distributions while other techniques can be applied just when the data can be clustered in monomodal groups.

4.4.1.2 Dynamic gesture recognition

Array of Spatio-Temporal Histograms of Local Quantized Depth Patterns (ASTH-LQDP) The dynamic hand gesture recognition solution which main block diagram is presented in Figure 4.13 was designed as an HMI solution via hand gesture recognition by means of depth information. The main contribution of the system is the design of a specific feature descriptor, the Array of Spatio-Temporal Histograms of Local Quantized Depth Patterns (ASTH-LQDP), which has the ability to characterize different dynamic hand gestures. Then, an SVM is used to distinguish among different hand gestures characterized by the feature descriptor.

The proposed ASTH-LQDP [4] is a highly discriminative descriptor for depth video sequences that is able to efficiently encode hand gestures for recognition purposes. It is a radical evolution from the popular LBP [160] justified by two facts: the use of depth imagery, instead of color, and the introduction of the temporal dimension, needed to encode dynamic hand gestures. Regarding the first issue, the ASTH-LQDP descriptor is adapted to the special characteristics of the depth imagery, such as the spatial and depth resolution, and the absence of strong variations in pixel values (unlike the color imagery, where there can

Figure 4.16: Illustration of the block and image levels A histogram of LQDP codes is computed for each block of the image, and then a new feature vector is computed as the concatenation of all the histograms resulting from each block.

With this new configuration, considering $N_b = 3$ and $P = 8$, the neighborhood code has a length according to the following equation:

$$N_{code} = N_q \times P = 24 \text{ bits},$$

Block level The block level computes the LQDP codes in an image region, and then calculates a histogram of LQDP codes, called H-LQDP, which is used as feature vector to characterize the corresponding image region. The H-LQDP histogram is composed by the following number of bins:

$$2^{N_q \times P} = 2^{24} = 16,777,216 \text{ bins}.$$

The selection of the parameters N_b and P is critical to restrict the length of the resulting histogram.

Image level The image level divides the image into $N_s \times N_s$ non-overlapping block regions, in this case $N_s = 4$, and applies the block level stage to each block region to obtain an H-LQDP for each one. Then, a new feature vector is obtained by concatenating all the H-LQDP histograms, which is called Array of Spatial Histograms of Local Quantized Depth Patterns (ASH-LQDP). This division of the depth image into blocks allows to incorporate some spatial information to the H-LQDP feature descriptor, which lacked of any spatial reference due to the histogram computation of LQDP codes. Figure 4.16 illustrates the block and image levels.

Sequence level The video sequence level extends the ASH-LQDP descriptor to be able to recognize dynamic hand gestures that span temporally. In this level, a temporal

subsampling of N_i frames in the video sequence is performed, resulting in a selection of depth images. Then, an ASH-LQDP descriptor is computed for every depth frame. Finally, a new feature vector is obtained as the temporal concatenation of the N_i ASH-LQDP descriptors of the subset of sampled images. This is the final ASTH-LQDP vector. The temporal subsampling has a dual purpose. On the one hand, it allows to recognize hand gestures executed at different speeds. And on the other hand, the subsampling shortens the resulting feature vector in comparison with the exhaustive version that uses all the frames in the sequence. Since the length of the ASTH-LQDP descriptor is equal to:

$$ASTH-LQDP_{desc} = N_i \times N_s \times N_s \times 2^{N_b \times P} = N_i \times 4 \times 4 \times 16,777,216 = N_i \times 268,435,456$$

this length can give rise to prohibitive feature sizes for its use in practical machine learning systems, so the subsampling procedure becomes very important. Figure 4.17 illustrates the video sequence level.

Figure 4.17: Computation of the ASTH-LQDP feature vector at the video sequence level.

4.4.2 Real-time scenario

In this section, a description of different solutions adopted to perform a real-time HMI are described in detail. For all of those solutions, the Leap Motion device [27] is used to acquire hand images, and therefore, near-infrared images are used, instead of depth information used in previous sections. The specific design of the Leap Motion, with its three emitter LEDs, allows to obtain an image in which closer objects (the hand in our case) have greater intensity values on the image and the background information is reduced which allows the recognition algorithm to work better. Three different solutions are proposed in this thesis to solve the hand gesture recognition problem. The first solution recognizes static gestures using the DLQP feature descriptor (see Section 4.3.1). This descriptor is applied to the whole near-infrared image (containing a hand gesture and other background regions), then the descriptor

Figure 4.18: Block diagram of the proposed system.

length is reduced by using a CS algorithm (see Section 4.4.1.1), and finally, an SVM algorithm (see Section 4.3.4) is used for the recognition process. As all parts of the system have been previously described, in this section just the other two solutions are described. It is important to notice that in this section infrared images are used instead of depth images used by previously introduced solutions. The other two solutions, that can recognize both static and dynamic gestures, are described in Section 4.4.2.1 and Section 4.4.2.2. They introduce some novelties with respect to previously presented algorithms and systems. These solutions are more oriented to work in real-time by reducing the complexity, and therefore the computational cost, and also enabling a more intuitive interaction, minimizing the delay between the gesture performance and the gesture recognition.

4.4.2.1 Real-time hand gesture recognition system using HOG and LBP descriptors with an SVM classifier

The system presented along this section is able to recognize dynamic hand gestures in real-time. It can be divided into four stages (see Figure 4.18). The first one is a multi-proposal hand segmentation stage. The second stage computes a feature vector from every candidate hand region, obtained by combining HOG and LBP descriptors. The third stage predicts the type of hand pose using a bank of multi-class SVMs. The fourth and last stage recognizes the dynamic hand gesture using a sliding window in the temporal domain.

Multi-proposal hand segmentation The multi-proposal hand segmentation stage computes several putative hand region segmentations with the purpose that at least one of them represents a satisfactory hand segmentation. This allows the system to be robust to bad hand segmentations, which is a permanent problem for any visual segmentation algorithm. This is accomplished by combining the well-known Otsu's algorithm [165] guided by the hand reflectance characterization described in Appendix A. The Otsu's algorithm generates satisfactory segmentations when the image can be divided into foreground and background

Figure 4.19: Image segmentation with different thresholds: (a) original infrared, (b) outcome with the Otsu's threshold, (c) outcome with a threshold equal to that of Otsu's plus one standard deviation, and (d) outcome with a threshold equal to that of Otsu's minus one standard deviation.

in some feature space. It estimates an optimum threshold that divides the feature space into the two clusters (the desired foreground and background) by minimizing the total intra-class variance of the clusters. Classification errors are produced in practice ruining the posterior hand gesture recognition when just one segmentation is used. This is because the output of this process can vary from one sample to the next as the intensity values have certain noise value which changes the amount of forearm segmented. To solve this problem multiple hand segmentations are computed in parallel by applying a probabilistic wrapper to the Otsu's method with the purpose that at least one of the hand segmentations has the enough quality. Assuming that correct hand segmentation is somewhat close to the solution provided by Otsu's method, more segmentations candidates are generated by modeling the process using a Gaussian distribution. Thus, hand segmentation samples are obtained by drawing thresholds values from this Gaussian distribution, whose mean is the threshold obtained by the Otsu's method, and whose standard deviation is the value obtained from the hand reflectance characterization. Figure 4.19 shows an example of the segmentation stage using the threshold value provided by Otsu's algorithm and two other values, one greater and another lower than this threshold. Observe that the hand is segmented along with different forearm portions, while the background is discarded. The partially segmented forearm does not represent a problem for the recognition system, since the training images are subject to the same segmentation processing, and therefore the involved classifier learns that the hand gesture can include some forearm regions.

The hand reflectance characterization is also used to discard abnormal segmentations by comparing the mean intensity value of a segmented hand region with the corresponding value resulting from the previous study. This mechanism, for example, can discard those

abnormal segmentations resulting from situations where the hands are not present.

Feature extraction A feature vector is computed for every segmented hand image region that is compact, repeatable, and stable. This is achieved by combining the HOG [20] and the LBP [160] feature descriptors. The feature extraction stage can be divided into three steps: block division of the hand region, computation of feature descriptors per block, and computation of the final hand descriptor.

The first step, block division of the hand region, splits the image into $N_s \times N_s = 5 \times 5 = 25$ non-overlapping blocks as shown in Figure 4.20. This block division will enrich the final hand region descriptor with valuable global spatial information. The number of blocks has been selected as a tradeoff between the length of the vector and the discriminative power: more divisions implies a higher discriminative capability, but also a higher dimensional feature vector, and vice versa.

The second step, computation of feature descriptors per block, characterizes the structural content of every block by means of the HOG and LBP descriptors. HOG describes an image region by means of a set of local histograms that count the occurrences of gradient orientations. It is very robust to illumination and contrast changes. The original HOG descriptor parameters have been used: the input block is divided into $N_{bl} = 16$ sub-blocks, sub-blocks are then divided into $N_c = 4$ cells, and $N_{bi} = 9$ bins are used per cell histogram. This configuration generates a final HOG histogram of dimension:

$$N_{HOG} = N_{bi} \times N_c \times N_{bl} = 9 \times 4 \times 16 = 576$$

On the other hand, the LBP is based on the computation of histograms of intensity differences among pixels in a local neighborhood, and it is also robust to dramatic illumination changes. The classical configuration of the LBP descriptor uses a neighborhood of 9 pixels (8 neighbors per pixel), 8 pixels to encode the neighborhood differences, and histograms of 256 bins. This configuration generates feature vectors of $N_{LBP} = 2^8 = 256$ components per image block.

The last step, computation of the final feature vectors, stacks the HOG and LBP feature vectors of all the image blocks, obtaining a final descriptor with the following number of components (see Figure 4.20):

$$(N_s \times N_s) \times (N_{HOG} + N_{LBP}) = (5 \times 5) \times (576 + 256) = 20,800$$

Classification of hand poses For the classification process, a novel augmented multi-class SVM is used to improve the recognition accuracy with the highly distorted images acquired by the Leap Motion sensor. One of the main effects of this distortion is that the

Figure 4.20: Block division and HOG and LBP histograms.

aspect ratio of the bounding box that encloses a hand region can dramatically change, and subsequently the derived feature descriptor. This makes the learning process of a classifier very complex, since very dissimilar feature descriptors can represent the same entity class. To efficiently address this problem, an augmented multi-class SVM has been designed, which augments every hand pose class into multiple subclasses. Every subclass represents a subset of the whole range of bounding box aspect ratios associated to one specific hand pose. Thus, the complexity of dealing with a large range of different visual appearances per hand gesture is divided into several subclasses. Each pose of the dataset is clustered into three different groups using the K-means strategy according to its aspect ratio (see Figure 4.21). Then, an SVM is trained using as positive samples those belonging to one cluster of one pose, and as negative samples the rest of the samples of the same pose and all samples of the other poses of the dataset. The subclasses are automatically defined by clustering the annotated bounding boxes in a given training dataset according to the aspect ratio value. The number K of clusters should be carefully chosen so that every cluster contains a sufficient number of samples to properly train all the subclasses in the cluster.

As a result, a multi-SVM with $K \times N_c$ subclasses is obtained, where N_c is the original number of hand poses to be recognized. Every output of the multi-SVM is a $K \times N_c$ binary vector that indicates the belonging to one subclass, which is in turn associated to a specific hand pose class.

Lastly, to improve the recognition score, a non-linear Hellinger kernel [161], more commonly known as Bhattacharyya distance, is used for the input features descriptors to the multi-SVM.

Dynamic hand gesture recognition The output of the previous classification stage is the detection of a hand pose per frame, which only could recognize static hand gestures. To be able to also recognize dynamic hand gestures, a temporal voting strategy has been proposed. This strategy creates a robust feature descriptor able to discriminate dynamic

Figure 4.21: Example of the hand pose classification strategy using four hand poses and three different clusters (K-means strategy).

hand gestures, even when the computed sequence of hand gestures contains errors. The voting strategy considers a sliding window in the temporal domain to compute a moving histogram of hand poses. Thus, at each time step, the histogram contains the distribution of the detected hand poses in the last frames (the length of the temporal sliding window). This histogram compactly represents a dynamic hand gesture. To increase the discriminative power of the temporal histogram of hand poses, multiple histograms are considered inside the temporal sliding windows, which introduces more temporal information useful to encode the order of blocks of consecutive hand poses. Figure 4.22 shows an example of the previously described feature vector for hand gestures considering two histograms inside the temporal sliding window. As can be observed, most of the detections of the first half belong to the hand pose number 2, while most of the detections of the second half correspond to the hand pose number 3. Notice that the multi-histogram based feature vector is robust to the erroneously detected hand poses (different from 2 and 3 in the Figure 4.22). These errors can be originated from either transitions between hand poses, or just errors in the hand pose recognition stage.

Finally, dynamic hand gestures are recognized by using a multi-class SVM classifier whose input is the previous multi-histogram based feature vector. Therefore, every time that a new frame is available, the multi-histogram based feature vector is computed using the detected

Figure 4.22: Example of the dynamic hand gesture recognition.

hand poses in the last N_w frames, and then it is delivered to the multi-class SVM to identify the hand gesture performed by the user. To avoid errors derived from situations without gestures or when an unknown one is performed, an additional class for that purpose is considered in the classifier.

4.4.2.2 NN and HMM real-time solution based on the combination of HOG and LBP descriptors

This system is an evolution of the one introduced in Section 4.4.2.1. This solution is able to recognize among different static and dynamic hand gestures. One of the differences with respect to the previous one is that the feature descriptor is computed over the entire image, and therefore there is no need to apply a hand segmentation stage. This fact allows us to reduce the time needed to perform the recognition task.

With this new configuration, the solution is composed by three different stages (see Figure 4.23). The first one computes a feature vector from every image, which is obtained by combining HOG and LBP descriptors, this stage is the same as the one used in the previous solution (see Section 4.4.2.1), but considering the whole image, instead of a region obtained from a segmentation process. The second stage predicts the type of hand poses using a Neural Network (NN). The third, and last stage, recognizes the dynamic hand gesture using a bank of HMMs, in the temporal domain. Only the new stages, second and third, are presented below.

Classification of hand poses For the hand pose classification process, a MultiLayer Perceptron (MLP) neural network [22] is used as it allows not only to improve the performance with respect to the SVM-based solution (see Section 4.4.2.1), but it also allows an on-line training, which is more difficult to implement with the SVM-based solution. This fact enables the system to be trained with more samples, increasing the classification accuracy. The MLP is a supervised algorithm which has been successfully used for pattern recognition and classification. This kind of networks are fully-connected ones, meaning that all neurons of one layer are connected with every neuron of the next layer with the aim of exchange messages between each other. This kind of networks are composed by one input layer, some hidden layers, and an output layer. Each connection has associated a numerical weight,

Figure 4.23: Block diagram of the proposed solution.

which is learned during the training process, and makes the network to learn according to the input samples. For the training process, the Backpropagation algorithm [166] is used, which trains the network for a given set of labeled training data. As a first step of the training process, the network examines the output response to the input sample. Then, the response is compared to the expected output value to compute the error value. The error is then passed back through the network, and the network weights are adjusted according to the derivative of the error value.

The network used in this solution has an input layer with the number of neurons equal to the length of the feature vector, a first hidden layer with the size equal to the half of neurons of the input layer, a second hidden layer with the number of neurons equal to a quarter of the size of the input layer, and an output layer which size is equal to the number of hand poses that is wanted to be recognize (see Figure 4.24). As each layer of the proposed network has a large number of neurons, to avoid problems related to overfitting, or vanishing gradient descend in the training process, a dropout scenario has been considered in which some neurons of the network are deactivated during the training process. With respect to the activation functions, for the two hidden layers, a rectifier linear unit has been used as it allows to obtain better and faster results than the ones achieved by using the sigmoid function. And for the output layer, a softmax activation function has been used as it is wanted to output the gesture for which the maximum accuracy value has been obtained by the network. At this point the gesture with higher probability is selected as the detected one.

Dynamic hand gesture recognition For the recognition of dynamic hand gestures, a solution based on the use of HMMs is proposed at this point. HMM [167], which is an

Figure 4.24: Structure of the proposed Neural Network architecture. Where $N = 20800$ is the feature vector length, and L is the number of gestures.

extension of the Markov Model, has been widely and successfully used in both speech and image recognition systems with the aim of model temporally correlated data streams, as they effectively capture correlations along time sequences. This kind of models are discrete state-space stochastic ones that can be used to solve hand gesture recognition problems as is the case of this system. The entire HMM can be expressed as follows:

$$I = (N, M, A, B, P) \quad (4.10)$$

where N is the number of hidden states (which in this case represents the number of consecutive frames that conform the hand gesture), M indicates the number of observable states, A is the transition matrix, B is the observation matrix, and P represents the initial state probability.

The training process of the HMM is performed by means of the Baum-Welch algorithm [24], which uses a forward-backward algorithm to update the variables of the model making use of the Expectation-Maximization algorithm [168]. This algorithm estimates the parameters of the HMM via the maximum likelihood value given a set of input observed features. For the decoding or prediction process, the Viterbi algorithm [169] is used to predict the most likely hidden state sequence given an input sequence of observations.

In this solution, it has been proposed to use a bank of HMMs, one per dynamic hand gesture. The input for each HMM is a consecutive number of contiguous temporal hand pose detections representing the dynamic gesture that it is wanted to be recognize. The final detection is associated to the HMM whose output has a higher likelihood value. A model able to detect the absence of gesture has also been trained.

4.5 Conclusions

Along this chapter different solutions have been presented to solve both, face and hand gesture recognition problems using visual information. Different feature descriptors have been designed to solve both problems taking into account the source of visual information used, depth in the case of face recognition, and depth or infrared in the case of the hand gesture solutions. For the face recognition problem, three feature vectors have been designed, the DLQP, the DS-LQP and the BAG-D3P which exploit the characteristics of the high-resolution depth information provided by the Kinect v2 device. For the three presented solutions an SVM algorithm was used to classify among different subjects.

Two of the descriptors used for face recognition, the DLQP, and the BAG-D3P, have also been used to characterize depth-based hand gestures in combination with a Compressive Sensing (CS) algorithm used to reduce the size of the obtained descriptors to make them more tractable for the classifier. For the case of dynamic hand gesture recognition, the ASTH-LQDP descriptor has been designed to include temporal information. In the three cases, an SVM algorithm was used for classification tasks. For the solutions that make use of infrared information acquired by the Leap Motion device, a first approximation using the DLQP descriptor has been presented to distinguish among different static hand gestures by means of an SVM algorithm. A first approximation to recognize dynamic gestures combines HOG and LBP descriptors to characterize each image, then a novel augmented multi-class SVM identifies the hand pose for every image, and finally, a temporal voting strategy in combination with a multi-class SVM algorithm is used for the temporal recognition of the hand gestures. A final solution for dynamic hand gesture recognition also uses HOG and LBP descriptors to characterize every image, but then, a MultiLayer Perceptron (MLP) based Neural Network is used for the hand pose recognition, finally, a bank of HMMs are used to recognize among different dynamic hand gestures.

Chapter 5

Databases

5.1 Introduction

With the aim of validating the solutions presented in Chapter 4, different databases have been created to test their performance with other state-of-the-art works. Since there are hardly databases acquired by the new sensors presented in Chapter 3, the created databases not only are used to test the presented systems, but also to provide a reference for evaluating HMI systems in general. In addition to describe the used databases for face and hand gesture recognition, a data augmentation technique is presented to generate more samples by modifying the acquired ones. A summary of the proposed databases with their purpose, the device used to acquire the information and the type of visual information of their images is presented in Figure 5.1.

5.2 Face databases

With respect to the face recognition, two different databases, composed of the same subjects, have been created. One of them has high-resolution depth images obtained by the Kinect

Database	Task	Device	Visual information
HRRFaceD	face recognition	Kinect v2	depth
LRRFaceD	face recognition	Senz3D	depth
depthGestRecog	dynamic hand recognition	Kinect v2	depth
leapGestRecog	static hand recognition	Leap Motion	infrared
MultiModHandGestRec	dynamic hand recognition	Leap Motion	infrared
leapHandGestUAV	dynamic hand recognition	Leap Motion	infrared

Table 5.1: Summary of the created databases.

v2 device (Section 5.2.1), and the other one has low-resolution depth images acquired by the Creative Senz3D device (Section 5.2.2).

5.2.1 High Resolution Range based Face Database (HRRFaceD)

A new depth-based face database acquired by the last generation of depth devices, which provides extended spatial and depth resolutions has been created. This database has been built by using the new Microsoft Kinect v2 device [9], which has a significant higher depth resolution than the first Kinect device [25], and other similar depth sensors. The database is composed of 22 sequences, each one composed of 1,000 images from different perspectives, of 18 different subjects (15 men and 3 women, 4 people with glasses). They were recorded while each subject was sitting about 50 cm away from the sensor and the head placed at the same height as the sensor. All subjects rotated their heads with the aim of obtaining as much information as possible from the head: nose, eyes, mouth, ears, etc. The database has been made publicly available [170].

Figure 5.1 shows three sets of depth images from three different subjects of the HRRFaceD database, where the main characteristics of the face can be distinguished: eyes, mouth, ears and nose. Depth maps have been saved with 16 bits of depth resolution, and with a spatial resolution of 512×424 pixels.

5.2.2 Low Resolution Range based Face Database (LRRFaceD)

A depth-based face database obtained by the Creative Senz3D device [8] (which provides depth images with lower depth resolution than the ones provided by the Kinect v2 device) has been created. The database is composed of 22 sequences, each one composed of 1000 images from different perspectives of 18 different subjects (15 men and 3 women, 4 people with glasses). They were recorded while each subject was sitting about 50 cm away from the sensor and the head placed at the same height as the sensor. All subjects rotated their heads with the aim of obtaining as much information as possible from the head: nose, eyes, mouth, ears, etc. The database has been made publicly available [171].

Figure 5.2 shows three sets of depth images from three different subjects of the LRRFaceD database, where it can be seen some parts of the face, but they are not so distinguishable as in the case of the HRRFaceD ones. In this case, depth maps have been saved with 8 bits of depth resolution and a spatial resolution of 320×240 pixels.

Figure 5.1: Three sets of depth maps for different subjects from the HRRFaceD database, acquired by the Kinect v2 device: (a) a man without glasses, (b) a man with glasses and (c) a woman without glasses.

Figure 5.2: Three sets of depth maps for different subjects from the LRRFaceD database, acquired by the Senz3D device: (a) a man without glasses, (b) a man with glasses and (c) a woman without glasses.

5.3 Hand databases

In the case of hand gesture recognition, different databases have been generated by using both, depth information acquired by the Kinect v2 device, and near-infrared information acquired by the Leap Motion device. Also, different databases composed of both static and dynamic gestures were created to be able to test all the different proposed systems. In the databases, the hand gestures were performed using the right hand.

5.3.1 Depth based Gesture Recognition Database (depthGestRecog)

A new depth-based hand gesture database has been generated using the Microsoft Kinect v2 device [9], which is publicly available in [172]. The database is composed of 11 different hand gestures, 5 of them are dynamic (Figure 5.3) and other 6 are static (Figure 5.4). The dynamic gestures are:

- The open palm moving from up to down in front of the device (palm ud).
- The open palm moving from right to left (palm rl).
- The close palm with the thumb extended moving from the position in which the thumb is pointing up to the position in which the thumb points down (thumb ud).
- The index and middle fingers extended and separated from the palm moving to touch the thumb extended (pick)
- The open palm with all fingers separate moving to the closed palm (grab).

The static ones are:

- An open palm with all fingers extended and separated (open palm).
- A closed palm with the thumb pointing down (thumb d).
- A closed palm with the thumb pointing up (thumb u).
- A closed palm with the index pointing to the camera (one finger).
- The index and middle fingers touching the thumb pointing to the camera (three fingers).
- The closed palm (fist).

Each gesture was performed by 6 different subjects (3 men and 3 women), whose hands are of different sizes. A total number of 100 frames were recorded per gesture and per subject, resulting in 600 depth frames available for each gesture. The regions containing the hand-gestures were manually selected using a bounding box of 100×100 pixels.

Figure 5.3: Dynamic hand gestures of the depthGestRecog database. From left to right: open palm moving up and down, open palm moving from left to right, close hand with the thumb extended and moving up and down, moving three fingers like piking something, and open and close the hand like grabbing something. The gray level of those depth-maps represents distances between 600 mm (darker) and 700 mm (lighter) in the hand and forearm pixels.

Figure 5.4: Static hand gestures of the depthGestRecog database. From left to right in the first row: static palm open, close hand with the thumb pointing down, close hand with the thumb pointing up. From left to right in the second row: one finger pointing to the camera, three joint fingers pointing to the camera and close hand. The gray level of those depth-maps represents distances between 600 mm (darker) and 700 mm (lighter) in the hand and forearm pixels.

Figure 5.5: Database samples of the ten different gestures performed by different subjects.

5.3.2 Hand Gesture Recognition Database Using Leap Motion (leapGestRecog)

A near-infrared based hand gesture database, composed of different static hand gestures acquired using the Leap Motion device [27], has been created. The sensor has been placed over a table, where a user moved their right hand over the sensor at a distance between 10 and 15 cm in front of it. A group of 10 different gestures performed by 10 different subjects (5 women and 5 men) have been acquired. A total number of 200 frames have been recorded for each gesture and each subject. The database is publicly available in [173]. In Figure 5.5, a sample of each hand gesture performed by different subjects is shown. In the first row from left to right, there are the following hand poses:

- An open palm parallel to the sensor (palm).
- A closed palm with the thumb and index fingers extended (l).
- A closed palm (fist).
- A fist perpendicular to the sensor (fist m).
- A closed palm with the thumb extended (thumb).

In the second row from left to right, there are the following hand poses:

- A closed palm with the index extended (index).
- An open palm with the index and thumb making a circle (ok).
- An open palm perpendicular to the sensor (palm m).
- A semi closed palm in a shape like a 'C' (c).
- An open palm with all its fingers separated and extended (palm d).

5.3.3 Multi-modal dataset for Hand Gesture Recognition (MultiModHandGestRec)

A hand gesture database composed of different static and dynamic hand gestures created by a set of infrared images acquired by the Leap Motion device [27]. The database has been made publicly available in [174]. For every temporal step, the Leap Motion device acquires two infrared images (one per camera), and the following skeletal information: hand and finger locations, information related with the finger extension, position of the wrist and palm center, and information about different finger parts. The database is composed of 15 different hand gestures (11 statics, and 4 dynamics), each one performed by 15 different subjects (5 women and 10 men). The static gestures are composed of only one kind of hand pose, while the dynamic ones are composed of two different kinds of hand poses. For the image acquisition process, each subject was sat in front of a computer with the Leap Motion placed horizontally on the table/console between the subject and the computer. Each subject was free to move the right hand over the sensor inside the Leap Motion field of view. A total number of 30 repetitions for every dynamic gesture were acquired by each subject. In the case of static gestures, a total number of 200 frames were obtained per gesture.

Figure 5.6 shows samples of every hand pose of the database for both static and dynamic gestures performed by different subjects. In the first row from left to right:

- A closed palm with the thumb and index fingers extended (l).
- A closed palm with the thumb pointing out of the sensor (fist m).
- A closed palm with the index extended (index).
- An open palm with the index and thumb making a circle (ok).

In the second row from left to right:

- A semi closed palm in a shape like a 'C' (c).
- A closed palm with the pinky and index fingers extended (heavy).
- A closed palm with the pinky and thumb fingers extended (hang).
- A closed palm with the index and middle fingers extended (two).

In the third row from left to right:

- A closed palm with the index, middle, and ring fingers extended (three).
- An open palm with the thumb closed to the palm (four).

Figure 5.6: Hand poses of the database.

- An open palm with all fingers separated (five).
- An open palm with all the fingers fixed (palm).

In the last row from left to right:

- A closed palm (fist).
- An open palm perpendicular to the sensor (palm m).
- An open palm with the back of the palm facing the sensor (palm u).
- A closed palm with its back facing the sensor (up).

On the other hand, Figure 5.7 shows the set of hand poses that compose every dynamic gesture. From left to right:

- A palm moving from the open palm position to the closed palm position (go down).
- An open palm moving from the palm perpendicular to the sensor to the palm facing the sensor (go left).

Figure 5.7: Dynamic hand gestures of the database.

- A palm moving from the palm perpendicular to the sensor to the back of the palm facing the sensor (go right).
- A palm with its back facing the sensor moving to the closed palm with its back facing the sensor (go up).

5.3.4 Leap Motion Hand Gesture dataset for UAV management (leapHandGestUAV)

This database is composed of near infrared images obtained from the Leap Motion device [27] and it is oriented to the interaction with an UAV from a GCS. The database has been made publicly available in [175]. The database is composed of 11 different hand gestures (10 dynamics and 1 static), each one performed by 13 different subjects (6 women and 7 men). For the image acquisition process, each subject was sat in front of a computer with the Leap Motion placed horizontally on the table/console between the subject and the computer. Each subject was free to move the right hand over the sensor inside the Leap Motion field of view. A total number of 30 repetitions for every dynamic gesture were acquired by each subject. In the case of the static gesture, a total number of 200 frames were obtained.

The gestures can be divided into two groups according to their objective. The ones which are going to be used to command the movements of the UAV are shown in Figure 5.8, and the ones used to interact with the payload of the UAV are shown in Figure 5.9.

In Figure 5.8, a sample of the hand poses for the gestures used to command the UAV is shown. In the following list, all the gestures and the relationship between each one and the related command are described:

- Take off. The used gesture is an 'L', with thumb and index extended and the other fingers closed (See Figure 5.8a).

Figure 5.8: Hand poses for the gestures to command the UAV.

- Decrease pitch angle to make the UAV go to the left. The gesture starts with the palm perpendicular to the sensor. Then, the palm is rotated to the left until the palm is shown to the sensor (See Figure 5.8b).
- Increase pitch angle to make the UAV go to the right. The gesture starts with the palm perpendicular to the sensor. Then, the palm is rotated to the right until the back of the hand is shown to the sensor (See Figure 5.8c).
- Increase the altitude value to make the UAV go up. The gesture starts with the palm completely open showing the back of the hand to the sensor. Then fingers are moving up until the palm is almost close (See Figure 5.8d).
- Decrease the altitude value to make the UAV go down. The gesture starts with the palm completely open showing the palm to the sensor. Then, fingers are moving down until the palm is almost close (See Figure 5.8e).

In Figure 5.9, a sample of the hand poses for the gestures used to interact with the payload of the UAV is shown. In the following list, all the gestures and the relationship between each gesture and the related command are described:

- Zoom in. The gesture starts with all the fingertips joint together pointing to the sensor with the finger extended. Then, the fingertips separate opening the hand up to all the fingers are completely separated and extended (See Figure 5.9a).
- Zoom out. The gesture starts with the palm open with all the fingers extended and completely separated. Then, the fingers begin to move until all the fingertips are joint together pointing to the sensor with the finger extended (See Figure 5.9b). This gesture is composed of the same poses that the zoom in one but in inverse order.

Figure 5.9: Hand poses for the gestures to interact with the payload of the UAV.

- Move the camera pan to the left. The gesture starts with all the fingertips joint together pointing to the sensor with the fingers extended. Then, the hand in that position moves to the left (See Figure 5.9c).
- Move the camera pan to the right. The gesture starts with all the fingertips joint together pointing to the sensor with the fingers extended. Then, the hand in that position moves to the right (See Figure 5.9d). This is a symmetric gesture of the one used to move the pan to the left.
- Move the camera tilt up. When a tilt up gesture is recognized, the camera moves upwards in its vertical axis. The gesture starts with all the fingertips joint together pointing to the sensor with the fingers extended. Then, the hand in that position moves front (See Figure 5.9e).
- Move the camera tilt down. When a tilt down gesture is recognized, the camera moves downwards in its vertical axis. The gesture starts with all the fingertips joint together pointing to the sensor with the fingers extended. Then, the hand in that position moves back (See Figure 5.9f). This is a symmetric gesture of the one used to move the tilt up.

5.4 Data augmentation

One of the biggest problems in machine learning solutions is the absence or the scarce of samples to properly train a model. This fact affects face and hand gesture recognition, being time consuming and costly the creation of appropriate databases. To increase the number of samples of the previously presented databases, the use of different data augmentation

techniques have been proposed, which obtain new samples by making small appearance changes in the image samples.

It is important to notice that the number of samples needs to be high in both cases: static data (face and static hand recognition), and temporal dynamic data (dynamic hand recognition). For this reason, different data augmentation techniques have been applied to obtain more samples. In the case of static recognition, small modifications over the original images have been randomly performed, which include illumination changes, rotations, and translations. These modifications have been randomly applied (separately or in conjunction) over each image in the database. For the case of the dynamic recognition, new vector samples containing hand pose detections of a certain dynamic hand gesture have been generated by changing their original size, and also by adding random misdetections. The misdetections simulate erroneous detections in the original vector.

The data augmentation techniques do not only allow the classifier to obtain a more accurate model, but they also let the system to generalize better, obtaining a higher performance when the system is used in a real scenario application. This happens due to the fact that when images for the database are acquired, the environment is controlled and the subjects are really focus in the gesture performance, but conditions change when the system is used in a real scenario. In a real scenario, the subject is maybe focused in performing other tasks apart from the gesture, or the environment conditions could vary. Data augmentation techniques are proposed to simulate those changes trying to adequate the samples of the database as much as possible to the real scenario.

5.5 Conclusions

Along this chapter, the databases created as reference ones for evaluating HMI systems have been described. Two are the presented databases for face recognition: the HRRFaceD one which is filled by depth information acquired by the Kinect v2 device, and the LRRFaceD one which has depth information acquired by the Creative Senz3D device. Four are the databases created for hand gesture recognition: the depthGestRecog which is composed of different static and dynamic hand gestures with images acquired by the Kinect v2 device, the leapGestRecog which has different static gestures acquired by the Leap Motion device, the MultiModHandGestRec which is drafted by different static and dynamic hand gestures acquired by the Leap Motion device, and the leapHandGestUAV which is composed of different static and dynamic gestures acquired by the Leap Motion device. Finally, a data augmentation solution has been described to increase the number of samples of the previously presented databases.

Chapter 6

Results

6.1 Introduction

Once the proposed solutions and the created databases have been shown, this chapter presents evaluation results of the solutions developed in Chapter 4 using the databases introduced in Chapter 5. A comparison with other state-of-the-art solutions is also included. First of all, a description of the metrics used for comparison are introduced in Section 6.2, then, performance results of face recognition are included in Section 6.3. Similarly, Section 6.4 describes performance results of hand gesture recognition, and finally, Section 6.5 presents integration results of hand gesture recognition in the SAVIER demonstrator.

6.2 Metrics

This section presents the metrics used to test the performance of all the proposed algorithms. The main metric used to evaluate the recognition performance is the Confusion Matrix (CM). The CM is a table which allows to visualize the performance and widely used for supervised learning algorithms, such as face and hand gesture recognition. Each column of the CM represents the instances in an actual class, and each row represents the instances in a predicted class. From the CM values, three different metrics are extracted to further evaluation. The first one is the precision which is a relationship for each class between the correct classified samples (true positives) and the total number of samples classified as this class (true and false positives). The true positive values correspond to the elements of the main diagonal of the CM, and the false positive values for each class are the sum of the corresponding row except of the element of the main diagonal. The precision indicates the percentage of well classified samples of each class with respect to the total samples classified as belonging to that class. Numerically, it can be expressed as follows:

$$\text{Precision} = 100 \times \frac{\text{true positives}}{(\text{true positives} + \text{false positives})} \quad (6.1)$$

The second metric is the recall, which is a relationship between the correct classified samples of each class (true positives) and the total number of samples of each class (true positives and false negatives). The false negative values of each class are obtained as the sum of the corresponding column in the CM. The recall indicates the percentage of well classified samples of each class, and it is expressed as follows:

$$\text{Recall} = 100 \times \frac{\text{true positives}}{(\text{true positives} + \text{false negatives})} \quad (6.2)$$

Finally, the third metric is the F-score that is a relationship between precision and recall values. This metric is a good indicator of the global performance. The range is between 0 and 100, where 100 is the maximum score. This metric can be expressed as follows:

$$\text{F-score} = 100 \times \left(2 \times \frac{(\text{precision} \times \text{recall})}{(\text{precision} + \text{recall})} \right) \quad (6.3)$$

Additionally, the Receiver Operating Characteristics (ROC) curve has also been used to measure the performance of each binary classifier. This curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at different threshold values. The best classifier performance takes place when a value of both of 100% is achieved for both TPR and FPR, and a random classifier will achieve a value of 50%. The TPR is the precision metric explained above. The FPR metric indicates the percentage of samples classified as false positives against all samples classified as negatives for each class, and it can be expressed as follows:

$$\text{FPR} = 100 \times \frac{\text{false positives}}{(\text{true negatives} + \text{false positives})}$$

Commonly, the Area Under the Curve (AUC) is computed from the ROC curve. The AUC value is just a number between 0 and 1 that indicates how bad or good the ROC curve is. If a classifier obtains a value of 1, it indicates the best performance, whereas a value of 0.5 indicates a random classifier, and values lower than 0.5 indicates a bad performance classifier. In general, classifiers with an AUC value greater than 0.8 are considered as good ones.

		Databases				
		HRRFaceD	LRRFaceD	IIIT-D RGB-D [176]	EURECOM [177]	Biwi [178]
Solutions	DLQP	Depth	Depth	—	—	—
	DS-LQP	Depth	Depth	—	—	—
	BAG-D3P	Depth	Depth	Depth	Depth	Depth
	SIFT [52]	Depth	Depth	—	—	—
	LBP [57]	Depth	Depth	Depth	Depth	Depth
	PCA [48]	—	—	—	Depth+Color	—
	LGBP [49]	—	—	—	Depth+Color	—
	LQP [50]	—	—	—	—	Depth+Color
	HOG [20]	—	—	—	—	Depth+Color
	BSIF [51]	—	—	—	—	Depth+Color

Table 6.1: Summary of the databases used to test different face recognition solutions. The three first solutions are the proposed ones.

6.3 Face recognition

In this section, experiments testing the performance of different solutions using various databases are described. To have an overview of which solutions have been tested using which databases, Table 6.1 summarizes those tests indicating also the kind of visual information used by each solution from each database.

The proposed face recognition algorithms have been tested using the two created depth-based face databases: the HRRFaceD (Section 5.2.1), and the LRRFaceD (Section 5.2.2). Additionally, publicly available databases have been used. For all the proposed solutions (DLQP, DS-LQP, and BAG-D3P), an SVM algorithm has been used for classification purposes. The following parameters have been considered for each proposed descriptor:

- For the DLQP descriptor (Section 4.3.1), the following configuration parameters have been used: radius of the neighbor $R = 1$, number of neighbors $P = 4$ (west, north, east and south), $N_b = 3$ bits for the quantifier, and an image division of $N_s = 4$ horizontal and vertical non-overlapping blocks (a total number of $N_s \times N_s = 16$ blocks). With these parameters, the final DLQP descriptor is a histogram of $16 \times 2^{4 \times 3} = 65,536$ bins.
- For the DS-LQP descriptor (Section 4.3.2), the same parameters than the DLQP descriptor have been used: $R = 1$, $P = 4$, $N_b = 3$, and $N_s = 4$. Additionally, each image block has been divided into $N_g = 2$ horizontal and vertical non-overlapping sub-blocks. With these parameters, the final DS-LQP descriptor is a histogram of $5 \times 16 \times 2^{4 \times 3} = 327,680$ bins.
- Finally, for the BAG-D3P descriptor (Section 4.3.3), the following parameters have been used: radius of the neighbor $R = 1$, number of neighbors $P = 9$, $N_b = 3$

Subject	DLQP			DS-LQP			BAG-D3P		
	precision	recall	F-score	precision	recall	F-score	precision	recall	F-score
Mean	48.69	45.54	44.15	75.41	71.06	70.73	77.22	71.53	71.67
1	82.35	75.68	78.87	100.00	97.44	98.70	97.50	95.00	96.23
2	46.88	38.46	42.25	78.13	65.79	71.43	32.50	60.00	42.16
3	50.00	18.92	27.45	96.43	79.41	87.10	70.00	90.00	78.75
4	50.00	3.03	5.71	48.57	51.52	50.00	62.50	62.50	62.50
5	28.89	50.00	36.62	28.85	88.24	43.48	92.50	80.00	85.80
6	36.11	34.21	35.14	82.35	45.16	58.33	85.00	22.50	35.58
7	69.44	75.76	72.46	76.92	85.71	81.08	100.00	87.50	93.33
8	30.23	39.39	34.21	69.57	66.67	68.09	95.00	65.00	77.19
9	34.15	43.75	38.36	50.94	84.38	63.53	90.00	92.50	91.23
10	26.87	52.94	35.64	88.46	71.88	79.31	95.00	55.00	69.67
11	29.31	47.22	36.17	44.44	68.57	53.93	47.50	65.00	54.89
12	44.44	34.29	38.71	71.05	75.00	72.97	95.00	67.50	78.92
13	58.33	35.00	43.75	94.12	40.00	56.14	32.50	70.00	44.39
14	36.00	25.71	30.00	65.52	55.88	60.32	35.00	52.50	42.00
15	57.14	70.59	63.16	88.89	88.89	88.89	70.00	85.00	76.77
16	67.35	84.62	75.00	96.00	72.73	82.76	92.50	85.00	88.59
17	80.00	36.36	50.00	95.83	69.70	80.70	100.00	77.50	87.32
18	48.84	53.85	51.22	81.25	72.22	76.47	97.50	75.00	84.78

Table 6.2: Precision, recall, and F-score results of the DLQP, DS-LQP, and BAG-D3P algorithms using the LRRFaceD database.

bits for the quantifier, a word of 108 bits divided in words of $N_{div} = 9$ bits, and an image division of $N_s = 4$ horizontal and vertical non-overlapping blocks. With these parameters, the BAG-D3P descriptor is a histogram of $4 \times 4 \times 12 \times 2^9 = 98,304$ bins.

The descriptor performance is evaluated in two different scenarios: firstly, they are going to be tested using the LRRFaceD database that is composed of depth images with low resolution, which limits the amount of information that can be extracted from them, and secondly, they are going to be tested using the HRRFaceD database that have depth images with higher resolution, allowing the descriptors to obtain more discriminative information of the human faces. The results of both evaluations are explained bellow.

Table 6.2 shows the obtained metrics precision, recall and F-score for the three proposed descriptors (DLQP, DS-LQP, and BAG-D3P), designed to exploit the 3D information of the human faces, using the LRRFaceD database. The database has been randomly split into a training set with 80% of the samples of each subject, and a testing set with 20% of the samples of each subject. The BAG-D3P feature descriptor is the one that in general obtains better results, achieving an F-score mean value of 71.67%, as it is the most complex one

Figure 6.1: ROC curves for each classifier of the BAG-D3P algorithm using the LRRFaceD database.

and it is the one that takes into account more discriminative information from the depth data considering a neighborhood of 9 pixels, while in the other cases, a neighborhood of 4 pixels is just considered. The performance of the DS-LQP feature descriptor, measured as the F-score, is somewhat worse than the one achieved by the BAG-D3P, reducing its performance in a 1% value. Finally, the results obtained by the DLQP descriptors are really worse in comparison with the ones obtained by the other two descriptors, as its mean F-score value is decreased to 44.15%. This is because this descriptor is not able to obtain enough discriminative information from the depth maps to be able to distinguish among different subjects, i.e. the obtained feature vectors are quite similar for many of the subjects.

Figure 6.1 shows the ROC curves for each binary classifier corresponding to each subject of the LRRFaceD database, using the BAG-D3P feature descriptor. It can be observed that all the classifiers have a ROC curve close to the optimal one. The worst AUC value is 98.79% that corresponds to subject 14, and the mean AUC value is 99.88%. These values indicate that for all cases both TPR and FPR are really close to 100%, so all classifiers works really good.

Subject	BAG-D3P	DS-LQP	DLQP	SIFT [52]	LBP[57]
Mean	71.67	70.73	44.15	35.71	38.14
1	96.23	98.70	78.87	79.41	93.51
2	42.16	71.43	42.25	33.33	0.00
3	78.75	87.10	27.45	51.61	0.00
4	62.50	50.00	5.71	42.55	78.79
5	85.80	43.48	36.62	4.88	41.08
6	35.58	58.33	35.14	46.81	31.25
7	93.33	81.08	72.46	0.00	40.00
8	77.19	68.09	34.21	0.00	41.79
9	91.23	63.53	38.36	5.26	62.86
10	69.67	79.31	35.64	5.26	43.14
11	54.89	53.93	36.17	43.14	33.33
12	78.92	72.97	38.71	15.69	21.02
13	44.39	56.14	43.75	48.15	36.73
14	42.00	60.32	30.00	29.05	46.15
15	76.77	88.89	63.16	66.67	50.00
16	88.59	82.76	75.00	37.86	5.88
17	87.32	80.70	50.00	91.43	23.53
18	84.78	76.47	51.22	41.67	37.50

Table 6.3: Comparison of F-score values of the proposed descriptors (DLQP, DS-LQP, and BAG-D3P) with other state-of-the-art solutions: SIFT [52], and LBP [57] using the LR-FaceD database.

	DLQP			DS-LQP			BAG-D3P		
Subject	precision	recall	F-score	precision	recall	F-score	precision	recall	F-score
Mean	79.22	73.73	70.46	88.27	80.90	80.90	96.06	94.31	94.57
1	100.00	97.50	98.73	41.24	100.00	58.39	100.00	100.00	100.00
2	90.24	92.50	91.36	100.00	85.00	91.89	100.00	92.50	96.10
3	34.21	65.00	44.83	97.14	85.00	90.67	97.56	100.00	98.77
4	35.14	65.00	45.61	94.67	88.75	91.61	58.82	100.00	74.07
5	78.57	82.50	80.49	92.77	96.25	94.48	100.00	100.00	100.00
6	70.21	82.50	75.86	86.15	70.00	77.24	92.68	95.00	93.83
7	86.05	92.50	89.16	70.18	100.00	82.47	100.00	75.00	85.71
8	100.00	7.50	13.95	100.00	25.00	40.00	100.00	90.00	94.74
9	94.87	92.50	93.67	95.24	100.00	97.56	100.00	100.00	100.00
10	100.00	85.00	91.89	100.00	80.00	88.89	93.02	100.00	96.39
11	100.00	60.00	75.00	100.00	32.50	49.06	100.00	80.00	88.89
12	79.31	57.50	66.67	47.48	82.50	60.27	86.96	100.00	93.02
13	100.00	82.50	90.41	98.55	85.00	91.28	100.00	97.50	98.73
14	0.00	84.62	0.00	73.81	77.50	75.61	100.00	90.00	94.74
15	81.82	45.00	58.06	91.67	82.50	86.84	100.00	77.50	87.32
16	82.50	82.50	82.50	100.00	100.00	100.00	100.00	100.00	100.00
17	100.00	85.00	91.89	100.00	98.75	99.37	100.00	100.00	100.00
18	93.10	67.50	78.26	100.00	67.50	80.60	100.00	100.00	100.00

Table 6.4: Precision, recall, and F-score results of the DLQP, DS-LQP, and BAG-D3P solutions applied over the HRRFaceD database.

Table 6.3 shows a comparison, using F-score values, with other state-of-the-art descriptors using the same LRRFaceD database. These state-of-the-art descriptors are: an LBP-based method [57] that uses the standard LBP technique; and a SIFT-based method [52] that utilizes a dense SIFT feature extraction. These are widely used feature descriptors in the field of pattern recognition, and different works have been using them to classify different human body parts as face, hands, or the entire body. From that table, it can be observed how the best results are obtained by the proposed algorithms (BAG-D3P, DS-LQP, and DLQP). Next, the LBP follows, whose results are slightly better than the ones achieved by the SIFT feature descriptor. This also indicates us that it is a good idea to base our feature descriptors in the LBP. It also shows how important it is to change the feature descriptors taking into account the characteristics of the depth information, specially in the way that they have been considered by the BAG-D3P, as for all the proposed solutions the obtained results are better than the ones achieved by the original LBP.

Next, a comparison among the three proposed descriptors with high-resolution depth data using the HRRFaceD database is performed. Precision, recall, and F-score values are shown in Table 6.4. The database has been randomly split into a training set (80%) and a

testing set (20%). From that table it can be drawn that the BAG-D3P is the descriptor that achieves better results as it is the one able to obtain a bigger amount of information from the high discriminative depth maps of the database. For that descriptor, just for four subjects the F-score value is lower than 90% which indicates that the algorithm works well for all the subjects. Observing the results of the DS-LQP descriptor, it can be seen that the mean F-score value is lower than the one achieved by the BAG-D3P, decreasing in 14%. But if we observe the F-score value for each subject, it can be seen that for some subjects it has a value lower than 50 which indicates us that the algorithm is not able to discriminate well where the number of subjects is big. Finally, the obtained results for the DLQP descriptor, the simpler one, are the worst ones. Its mean F-score value decreases even more to a mean value of 70.46% and, in this case, the variance in the F-score values is higher as there are some subjects with an F-score value above 90% while for others, the F-score value is 0%. This is because the algorithm is not really able to distinguish between some of the subjects, like it is the case of subjects 8 and 14, which have similar face shapes and, in short, this simpler algorithm obtains more information from the shape of the face rather than from the face itself. From all of this, we can conclude that adding more information of the neighborhood to the descriptor is really important to characterize human faces in high resolution depth images.

Figure 6.2 shows the ROC curves for each subject of the HRRFaceD database, using the BAG-D3P feature descriptor. All the classifiers have a curve close to the optimal. The worst AUC value is 98.83% that corresponds to subject 10, and the mean AUC value is 99.97%. Comparing results on this figure with the ones introduced in Figure 6.1 using the LRRFaceD, it is claimed that the use of high resolution depth imagery aids to obtain better performance in the classification process.

Table 6.5 shows the results of the comparison, using F-score values, with other state-of-the-art descriptors: the LBP-based solution [57] and SIFT-based solution [52]. From that table it can be drawn that the BAG-D3P is the solution which obtains best results, a 94.57% in the F-score mean value. In the case of the LBP descriptor, a mean F-score value of 59.17% is achieved. If that value is compared with the results obtained with the three proposed methods (BAG-D3P, DS-LQP, and DLQP) which are an evolution of the LBP, it can be seen how a considerably increase in the mean F-score value is achieved. This indicates that just applying small modifications over feature descriptors specially designed to work over color imagery does not perform well when they are applied over depth imagery, but if they are modified according to the depth imagery characteristics, as the ones proposed in this thesis, the accuracy of the results increases considerably.

Comparing the recognition results using low-resolution and high-resolution depth im-

Figure 6.2: ROC curves for each classifier of the BAG-D3P algorithm using the HRRFaceD database.

Subject	BAG-D3P	DS-LQP	DLQP	SIFT [52]	LBP[57]
Mean	94.57	80.90	70.46	71.94	59.17
1	100.00	58.39	98.73	57.50	100.00
2	96.10	91.89	91.36	90.00	35.00
3	98.77	90.67	44.83	82.50	65.00
4	74.07	91.61	45.61	78.75	96.25
5	100.00	94.48	80.49	78.75	96.25
6	93.83	77.24	75.86	70.00	62.50
7	85.71	82.47	89.16	90.00	82.50
8	94.74	40.00	13.95	77.50	0.00
9	100.00	97.56	93.67	90.00	97.50
10	96.39	88.89	91.89	82.50	7.50
11	88.89	49.06	75.00	60.00	47.50
12	93.02	60.27	66.67	27.50	50.00
13	98.73	91.28	90.41	82.50	77.50
14	94.74	75.61	0.00	87.50	80.00
15	87.32	86.84	58.06	42.50	60.00
16	100.00	100.00	82.50	85.00	50.00
17	100.00	99.37	91.89	82.50	57.50
18	100.00	80.60	78.26	30.00	0.00

Table 6.5: Comparison in F-score value of the proposed descriptors (DLQP, DS-LQP, and BAG-D3P) with other state-of-the-art solutions: SIFT [52], and LBP [57] using the HRRFaceD database.

agery, it can be claimed that all the feature descriptors achieve better results when more information is available. This fact is specially significant for the BAG-D3P feature descriptor, which increases its performance from a mean F-score value of 71.67% with the LRRFaceD database up to a value close to 95% with the HRRFaceD database. This fact indicates us that both high-resolution imagery and specific descriptors to exploit it are key factors to achieve a high recognition performance.

The BAG-D3P descriptor (the one with best performance) has been additionally evaluated using other publicly available depth-based face databases (whose depth images have a noticeable lower depth resolution). These databases are briefly described as follows:

- The Biwi Kinect Head Pose database [178], which is composed of more than 15,000 images (in color and depth) of 20 male and female subjects, comprising different head poses per subject.
- The IIIT-D RGB-D database [176] is composed of 106 male and female subjects with a number of images between 11 and 254 per subject. The images are captured in color and depth, and contain different expressions per subject.
- The EURECOM Kinect database [177] is composed of 52 male and female subjects. Two acquisition sessions at different times are performed per subject, and 9 different states are available per session: neutral, smile, open mouth, left profile, right profile, occlusion eyes, occlusion mouth, occlusion paper (a paper partially occludes the face), and light on. The session 1 is used to train the system, and results over both session 1 and session 2 are reported. Also, color and depth imagery are acquired.

Apart from the previously used LBP-based and the SIFT-based solutions, other methods are also used for comparison purposes: a PCA-based method [48] that builds a low-dimensional face subspace; an LGBP-based method [49] that combines Gabor and LBP features; an LPQ-based method [50] that claims to be more robust than LBP to blurriness; an HOG-based method [20] that exploits the gradient information; and a BSIF-based method [51] that is a combination of LBP and LPQ methods. Not all the algorithms have been evaluated for all the databases since either there is no publicly available results for all databases or their code is not available (see Table 6.1 for more details on the solutions and databases used for each comparison). Databases have been split into training and testing sets following a similar setting than in the original papers. In the following tables, results are compared according to the accuracy value, as this is the available metric in the original papers. This metric is defined as follows:

$$\text{Accuracy} = 100 \times \frac{\text{Total number of correct faces}}{\text{Total number of faces}}$$

	BAG-D3P	LBP [57]
Biwi [178]	94.13	85.20

Table 6.6: Mean accuracy results using the Biwi database.

	BAG-D3P	LBP [57]	LPQ [50]	HOG [20]	BSIF [51]
IIITD [176]	84.72	84.40	84.70	82.80	77.80

Table 6.7: Mean accuracy results using the IIIT-D RGB-D database.

which represents the percentage of faces per subject that are correctly assigned to that subject.

Table 6.6, Table 6.7, and Table 6.8 present the results in mean accuracy value for all the subjects of the Biwi, the IIIT-D RGB-D, and the EURECOM databases, respectively. Again, the proposed BAG-D3P descriptor achieves the best recognition performance, despite that only depth information has been used in contrast with other works which use a combination of depth and color information. Notice that the performance between the proposed face recognition algorithm and the others is noticeable, but not so high as in the case of the HRRFaced database. This is specially significant for the IIIT-D RGB-D database, where not only a lower depth resolution sensor is used, but also there is a pre-processing quantization stage applied to the database that reduces further the number of bits to codify the depth values. For this reason, the proposed solution does not substantially improve the results obtained by other solutions. Anyway, the recognition performance of the proposed solution is still the best. Noticeable are also the results obtained in the EURECOM database, where the results obtained in the session 2 are significantly worse than for the session 1. The reason is that the database setting requires to use the session 1 as training and the session 2 as testing. In any case, the results obtained by the proposed algorithm are quite satisfactory, improving by far the other algorithms.

As conclusion, it is important to take into account the characteristics of the depth information in the specific design of the feature descriptor. Also, the discriminative power of the proposed descriptor is not only useful in the case of high quality depth information, but also with low resolution depth imagery.

		BAG-D3P	PCA [48]	LBP [57]	LGBP [49]
EURECOM [177]	Session 1	96.16	63.94	89.90	81.73
	Session 2	86.54	33.85	82.69	66.92

Table 6.8: Mean accuracy results using the EURECOM database.

		Databases	
		leapGestRecog	ASL Finger Spelling Database [72]
Solutions	BAG-D3P+SVM	—	Depth
	BAG-D3P+CS+SVM	Infrared	Depth
	Huang [52]	Infrared	Depth

Table 6.9: Summary of the databases uses to test different static hand recognition solutions.

		Databases		
		depthGestRecog	MultiModHandGest Rec	leapHandGestUAV
Solutions	DLQP+SVM	Depth	—	—
	DLQP+CS+SVM	Depth	—	—
	ASTH-LQDP+SVM	Depth	—	—
	ASH-LQDP+SVM	Depth	—	—
	LBP+SVM	Depth	—	—
	Real-time-SVM	—	Infrared	Infrared
	Real-time-NN	—	Infrared	Infrared
	HOG+NN	—	Infrared	Infrared
	LBP+NN	—	Infrared	Infrared
	Runqing [104]	—	Infrared	—
	Schmidt [124]	—	Skeletal	—
	Marin [136]	—	Skeletal	—
	Chuan [127]	—	Skeletal	—

Table 6.10: Summary of the databases uses to test different dynamic hand recognition solutions.

6.4 Hand gesture recognition

In the literature, there are some solutions that just recognize static hand gestures and others which are designed to distinguish between both static and dynamic gestures. For this reason, this section is split into two subsections: one for the static hand gesture results (Section 6.4.1), and another for both static and dynamic hand gestures (Section 6.4.2). In both sections, tests using different types of visual information that make up the different databases have been carried out. Also, different state-of-the-art solutions have been used to compare the proposed algorithms. For that reason, Table 6.9 and Table 6.10 summarize the tests indicating the type of visual information used by each solution from each database for static and dynamic hand gestures respectively.

6.4.1 Static hand gestures

Along this section, evaluation results for the proposed static hand gestures algorithms introduced using the databases presented in Chapter 5 and others publicly available ones are

going to be described. Also, a comparison with other state-of-the-art solutions is presented. For the proposed solutions (BAG-D3P+SVM and BAG-D3P+CS+SVM), an SVM algorithm has been used for the classification task. The section is divided into two sub-sections: one presenting tests using infrared imagery (Section 6.4.1.1), and other one with tests using depth imagery (Section 6.4.1.2).

6.4.1.1 Static hand gesture recognition using infrared imagery

A comparison between the proposed algorithm BAG-D3P+CS+SVM (which uses a BAG-D3P feature descriptor in combination with a CS dimensionality reduction technique) and the hand gesture recognition proposed by Huang [52] is presented. This work has three main stages: the first one is a segmentation stage to determine where the hand is placed in the image; a second stage computes the image descriptors using a bank of Gabor filters; and the last stage uses an SVM classification algorithm for the recognition. The leapGestRecog database (Section 5.3.2) is used, which has been randomly split in a testing set (80%) and a training set (20%). The following parameters have been considered for the configuration of the BAG-D3P+CS+SVM algorithm:

- For the BAG-D3P algorithm (Section 4.3.3): radius of the neighbor $R = 1$, number of neighbors $P = 9$, $N_b = 3$ bits for the quantifier, 108 bits-word division into words of $N_{div} = 6$ bits, and image division into $N_s = 6$ horizontal and vertical non-overlapping blocks. With these parameters, the BAG-D3P descriptor is a histogram of $6 \times 6 \times 18 \times 2^6 = 4,1472$ bins.
- For the CS algorithm (Section 4.4.1.1): the parameter M has been selected according to Equation 4.7. Taking into account that the length of the BAG-D3P feature descriptor is $N = 41,472$ and the sparsity of input vectors is $K = 15,352$ on average, then results a value of $M = 5,797$.

Table 6.11 shows the CM results for the BAG-D3P+CS+SVM algorithm. As it can be seen, the precision for all the gestures, corresponding to the elements of the main diagonal, are higher than 99.7%, proving the high performance of the algorithm.

Table 6.12 shows the precision, recall, and F-score values for the BAG-D3P+CS+SVM algorithm and the one proposed by Huang [52]. The proposed solution is the best one, achieving scores higher than 99% in precision, recall, and F-score values. In addition, the proposed solution does not need a segmentation stage as the one used by Huang, decreasing significantly the complexity.

	Palm	L	Fist	Fist_m	Thumb	Index	OK	Palm_m	C	Palm_d
Palm	100	0	0	0	0	0	0	0	0	0
L	0	99.9	0	0	0	0	0	0	0	0
Fist	0	0	99.8	0	0	0	0.3	0	0	0
Fist_m	0	0	0	100	0	0	0	0	0	0
Thumb	0	0.1	0.2	0	100	0	0	0	0	0
Index	0	0	0	0	0	100	0	0	0	0
OK	0	0	0	0	0	0	99.7	0	0	0
Palm_m	0	0	0	0	0	0	0	100	0	0
C	0	0	0	0	0	0	0	0	100	0
Palm_d	0	0	0	0	0	0	0	0	0	100

Table 6.11: Confusion Matrix with results of testing the BAG-D3P-based feature descriptor in combination with a CS solution over the leapGestRecog database.

	BAG-D3P+CS+SVM			Huang [52]		
	precision	recall	F-score	precision	recall	F-score
Mean	99.94	99.94	99.94	93.30	93.50	93.31
Palm	100.00	100.00	100.00	95.00	86.36	90.48
L	99.90	100.00	99.95	100.00	100.00	100.00
Fist	99.80	99.70	99.75	87.00	86.14	86.57
Fist_m	100.00	100.00	100.00	100.00	95.24	97.56
Thumb	100.00	99.70	99.85	95.00	96.94	95.96
Index	100.00	100.00	100.00	100.00	100.00	100.00
OK	99.70	100.00	99.85	87.00	100.00	93.05
Palm_m	100.00	100.00	100.00	95.00	90.48	92.68
C	100.00	100.00	100.00	87.00	94.57	90.63
Palm_d	100.00	100.00	100.00	87.00	85.29	86.14

Table 6.12: Precision, recall, and F-score values for the BAG-D3P+CS+SVM algorithm and the work presented by Huang in [52] using the leapGestRecog database.

6.4.1.2 Static hand gesture recognition using depth imagery

The performance of the BAG-D3P+CS+SVM algorithm has also been tested using the ASL Finger Spelling Database proposed in [72]. This database is composed of 24 gestures that represent 24 letters from the ASL. Each sign is recorded from 5 different subjects (non-native to sign language). For each gesture, over 500 samples are recorded, so the entire database is composed of around 48,000 samples. Each sample is composed of a depth and a color image. Nonetheless, only depth information is used for the BAG-D3P+CS+SVM algorithm. In Figure 6.3 one sample from each letter is shown. Observe that there are small differences in the position of the hand of each subject and the difficulty in distinguishing among some of the letters, as their performance is really similar.

Regarding the BAG-D3P+CS+SVM algorithm, the following parameters have been considered:

- For the BAG-D3P algorithm (Section 4.3.3): radius of the neighbor $R = 1$, number of neighbors $P = 9$, $N_b = 3$ bits for the quantifier, 108 bits-word division into words of $N_{div} = 9$ bits, and image division into $N_s = 4$ horizontal and vertical blocks. With these parameters, the BAG-D3P descriptor is a histogram of $4 \times 4 \times 12 \times 2^9 = 98,304$ bins.
- For the CS algorithm (Section 4.4.1.1): the parameter M has been selected according to Equation 4.7. Taking into account that the length of the BAG-D3P feature descriptor is $N = 98,304$ and the sparsity of input vectors is $K = 3,848$ on average, then results a value of $M = 4,739$.

Table 6.13 shows the confusion matrix. Looking at the main diagonal of that table, it can be observed that the performance of many of the gestures is more than an 80% indicating also a stable performance. Concretely, a mean value of 81% is obtained which, taking into account the large number of gestures that make up the database, is a high value. In the case of the gestures with less performance, it can be explained as some of the hand poses for different letters are quite close. This is the case of letter “c” which is confused with letter “b” because their poses are quite similar as in both cases the palm is open and in letter “c” the fingers are a little bit bent. Letter “k” is confused with letters “i” and “l” as in the three cases the index is extended and it just change the position of the thumb among the three poses. Also, the “r” is quite confused with letters “s”, “t”, “u”, “v”, “w”, “x”, and “y”, and if we see the samples for all those cases, there are two or three fingers extended but the difference among poses is not really appreciable. Considering the difficulty of distinguishing among all different poses, obtained results are quite good.

Next, a comparison with the state-of-the-art solution introduced by Pugeault in [72]

Figure 6.3: Samples of each gesture performed by the five subjects of the ASL Finger Spelling Database [72].

	a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y
a	83	7	1	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	2	0	0	0	0	0
b	0	90	14	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
c	1	0	81	3	2	0	0	0	0	0	0	0	0	0	1	2	0	1	0	0	0	0	0	0
d	1	0	0	94	2	2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	0	0	0	0	92	3	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
f	1	0	0	0	0	91	2	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
g	1	0	0	0	0	0	88	4	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
h	0	0	1	0	0	0	3	92	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
i	0	0	1	1	0	0	0	0	93	13	1	1	0	0	0	0	0	0	0	0	0	0	0	1
k	0	0	0	0	0	0	0	0	0	85	10	7	6	6	6	6	0	0	0	0	0	0	0	0
l	0	0	0	0	0	0	0	0	0	85	10	7	6	6	6	6	0	0	0	0	0	0	0	0
m	0	0	0	0	0	0	0	0	0	0	85	1	5	1	0	0	0	2	4	1	1	1	1	0
n	1	0	0	0	0	0	0	0	0	0	0	2	83	2	0	0	1	0	16	0	0	0	0	0
o	1	0	1	0	0	0	0	0	0	0	0	0	0	88	3	1	0	2	2	0	0	0	0	0
p	0	0	0	0	3	0	1	0	0	0	0	0	0	2	84	5	1	0	0	0	0	0	0	3
q	5	0	0	0	0	0	0	1	0	0	0	0	0	0	3	84	23	1	1	0	0	0	0	0
r	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	75	24	14	14	15	14	15	14
s	2	0	0	0	0	0	2	0	0	0	1	1	0	0	0	0	0	67	4	0	0	0	0	0
t	2	0	0	0	0	0	0	0	0	1	0	1	3	1	0	0	0	2	55	1	1	0	0	0
u	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	81	4	3	3	0
v	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	2	77	12	3	1
w	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	70	0	0	1
x	0	0	0	0	0	0	0	0	0	0	1	0	1	0	3	1	0	0	1	0	0	78	1	1
y	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	77	77

Table 6.13: Confusion matrix obtained of testing the BAG-D3P-based feature descriptor in combination with a CS solution over the ASL Finger Spelling Database.

is presented. This solution extracts hand shape features using Gabor filters, which are used with a multi-class random forest classifier. Additionally, a variation of the BAG-D3P+CS+SVM algorithm is used, without the CS algorithm, and that will be called BAG-D3P+SVM algorithm.

Table 6.14 shows the precision, recall, and F-score results of the comparison among BAG-D3P+CS+SVM, BAG-D3P+SVM, and Pugeault solutions. It can be seen how the BAG-D3P+CS+SVM solution is the one that obtains better results, achieving a mean F-score value of 82% and obtaining for many of the gestures a F-score value greater than 80%. Just for some gestures which pose is not easy to distinguish, as it was discussed previously, the F-score value is lower. Comparing these results with the ones obtained by the BAG-D3P+SVM solution, we observe how its F-score value decreases in 10% but it is still close to 75%. This fact indicates that the CS algorithm not only reduces the computational cost of the system but also helps the SVM to achieve better performance. Finally, it can be seen how the Pugeault performance decreases considerably as it just obtains a mean F-score value of 51%. It is important to notice that these results are obtained by combining depth and color information while in both BAG-D3P+CS+SVM and BAG-D3P+SVM solutions, only depth information is used. This relevant fact indicates the importance of the design of a specific descriptor to obtain as much relevant information as possible from depth images.

Figure 6.4 shows the ROC curves for each binary classifier corresponding to each subject of the LRRFaceD database, these classifiers are the ones used to obtain the results using the BAG-D3P feature descriptor. In that figure it can be seen how for all the classifiers the curve is close to the optimal. The worst AUC value is 93.02% that corresponds to letter d, and a mean AUC value of 98.29%, those values indicate that for all cases both TPR and FPR are really close to 100% so all classifiers works really good.

6.4.2 Dynamic hand gestures

This section presents performance results with the algorithms introduced in Chapter 4 using the databases presented in Chapter 5. A comparison with other state-of-the-art solutions is also presented. As in the case of static hand gestures recognition, this section is divided into two subsections: one with tests using depth imagery (Section 6.4.2.1), and other presenting tests using information obtained from the Leap Motion device (infrared or skeletal) (Section 6.4.2.2).

	BAG-D3P+SVM			BAG-D3P+CS+SVM			Pugeault [72]		
	precision	recall	F-score	precision	recall	F-score	precision	recall	F-score
Mean	76	74	73	85	81	82	52	53	51
a	61	91	73	82	83	83	39	75	51
b	70	88	78	85	90	87	54	83	66
c	90	75	82	89	81	85	39	57	46
d	84	95	89	92	94	93	31	37	33
e	89	68	77	96	92	94	58	63	60
f	93	87	90	92	91	91	36	35	36
g	88	94	91	91	88	89	62	60	61
h	83	85	84	93	92	92	86	80	83
i	70	93	80	84	93	88	50	73	60
k	60	73	66	67	85	75	73	43	54
l	85	87	86	97	85	90	67	87	76
m	47	77	58	84	85	85	44	17	24
n	71	52	60	79	83	81	38	23	29
o	74	72	73	90	88	89	29	13	18
p	73	81	77	85	84	84	54	57	55
q	76	47	58	71	84	77	87	77	81
r	33	58	42	40	75	52	56	63	59
s	87	56	68	87	67	76	17	17	17
t	61	37	46	82	55	66	10	7	8
u	75	77	76	84	81	83	80	67	73
v	82	64	72	79	52	63	34	87	49
w	86	73	79	97	70	81	95	53	68
x	90	60	72	91	78	84	36	20	26
y	94	74	83	96	77	86	86	76	81

Table 6.14: Comparison of precision, recall, and F-score values of testing different algorithms: DSQP-SVM, DSQP-CS-SVM and Pugeault [72] over the ASL Finger Spelling Database.

Figure 6.4: ROC curves for each classifier of the BAG-D3P+CS+SVM algorithm using the ASL Finger Spelling Database.

6.4.2.1 Dynamic hand gesture recognition using depth information

A comparison among the DLQP+CS+SVM (Section 4.4.1.1), the ASTH-LQDP+SVM (Section 4.4.1.2), the ASH-LQDP+SVM (which is the ASTH-LQDP+SVM solution but without considering temporal information), and the LBP+SVM solutions is carried out using the depthGestRecog database (Section 5.3.1). The following parameters are used for each solution:

- For the DLQP descriptor (used in DLQP+CS+SVM and in DLQP+SVM): radius of the neighbor $R = 1$, number of neighbors $P = 4$ (west, north, east and south), $N_b = 3$ bits for the quantifier, and an image division of $N_s = 4$ horizontal and vertical non-overlapping blocks (a total number of $N_s \times N_s = 16$ blocks). With these parameters, the final DLQP descriptor is a histogram of $16 \times 2^{4 \times 3} = 65,536$ bins.
- For the ASTH-LQDP descriptor: number of pixels of the neighborhood $P = 4$ (in a west, north, east and south spatial arrangement), number of samples for the temporal subsampling $N_i = 3$, number of bits for the quantization $N_q = 3$, and each depth image was divided into $N_s = 4$ horizontal and $N_s = 4$ vertical blocks. With this configuration, the ASTH-LQDP descriptor histogram has a length of: $N_i \times N_s \times N_s \times 2^{N_b \times P} = 3 \times 4 \times 4 \times 16,777,216 = 80,536,368$ bins.
- For the ASTH-LQDP descriptor: number of pixels of the neighborhood $P = 4$ (in a west, north, east and south spatial arrangement), number of samples for the temporal subsampling $N_i = 1$, number of bits for the quantization $N_q = 3$, and each depth image was divided into $N_s = 4$ horizontal and $N_s = 4$ vertical blocks. With this configuration, the ASTH-LQDP descriptor histogram has a length of: $N_i \times N_s \times N_s \times 2^{N_b \times P} = 1 \times 4 \times 4 \times 16,777,216 = 26,845,456$ bins.
- For the LBP descriptor: radius of the neighbor $R = 1$, number of neighbors $P = 8$, and each depth image was divided into $N_s = 4$ horizontal and $N_s = 4$ vertical blocks. The resulting descriptor histogram has $N_s \times N_s \times 2^P = 4 \times 4 \times 2^8 = 4,096$ bins.

In the following tables, the gestures, described in Section 5.3.1, are going to be numerated from 1 to 11 as follows: (1) open_palm, (2) palm_ud, (3) palm_lr, (4) thumb_d, (5) thumb_u, (6) thumb_ud, (7) one_finger, (8), three_fingers, (9) pick, (10) fist, and (11) grab. Images of each gesture of the database have been randomly split into a training set (80%) and a testing set (20%).

F-score values of the comparison among the previously cited solutions are shown in Table 6.15. Results with the DLQP+CS+SVM solution are in average the best ones, achieving a mean value of 96.59% in the F-score metric for this descriptor. It is also important

	DLQP+CS+SVM	ASTH-LQDP+SVM	ASH-LQDP+SVM	LBP+SVM
Mean	96.59	94.62	91.71	90.97
(1)	96.58	97.50	92.05	91.29
(2)	90.96	94.17	88.89	84.77
(3)	97.52	92.50	88.99	90.99
(4)	96.77	85.83	94.47	99.59
(5)	97.89	86.67	89.89	94.86
(6)	91.63	97.50	91.35	87.76
(7)	100.00	100.00	96.77	96.77
(8)	95.80	100.00	97.91	95.90
(9)	98.35	100.00	97.44	95.20
(10)	100.00	96.67	90.12	88.03
(11)	97.94	90.00	81.55	75.60

Table 6.15: F-score values of the compared algorithms tested over the depthGestRecog database.

to notice that the solution works good for all the gestures, as all of them have a F-score value greater than 90%. Results obtained for the ASTH-LQDP+SVM solution reduce the mean performance (mean F-score value) only in a 2%, but in this case, the performance for some gestures goes below a 90% value. If the temporal information is eliminated, as in the ASH-LQDP+SVM solution, the performance decreases in an additional 3%, indicating that adding temporal information helps to obtain better results, as it was expectable. Checking the results when the original LBP configuration is used, it can be seen how the performance decreases in another 1%, which results in a decrease of a 6% with respect to the results obtained by the DLQP+CS+SVM solution. These results indicate us that modifications on the original LBP are made in the right direction to obtain better results when the recognition process is made over dynamic hand gestures using depth information.

From Table 6.15, it can be seen how the DLQP+CS+SVM is the one that achieves the best performance. A specific test with this algorithm over the depthGestRecog database (Section 5.3.1) to determine the importance of the CS algorithm and the configuration of the neighborhood has been carried out. A range of tests changing the parameters R and P of the DLQP descriptor is presented, with the purpose of analyzing the relevance of the number of pixels of the descriptor neighborhood. The following parameters of the DLQP descriptor have been considered: $N_b = 3$ bits for the quantifier, and image division of $N_s = 4$ horizontal and vertical blocks.

Table 6.16 shows the values of the parameters M and K for different values of N (see equation 4.9). The M value increases much slower than N (the size of the original vector). In that table values for the following descriptor configurations are shown:

	R	P	N	K	M
CS_4	1	4	65,536	1,483	1,574
CS_8	1	8	131,072	2,662	2,905
CS_12	2	12	196,608	3,891	4,274
CS_24	2	24	393,216	6,790	7,717

Table 6.16: Values of M , N , and K for different values of R and P .

Gesture	DLQP+SVM	CS_4	CS_8	CS_12	CS_24
Mean	68.46	96.59	97.48	97.49	97.68
fist	77.14	96.58	99.59	98.76	98.77
grab	27.34	90.96	96.33	96.69	97.89
one_finger	77.87	97.52	96.64	95.62	96.36
palm	84.25	96.77	99.17	97.56	96.00
palm_lr	74.24	97.89	97.12	96.67	97.05
palm_ud	61.75	91.63	94.07	93.81	96.67
pick	79.31	100.00	100.00	99.17	100.00
three_fingers	68.87	95.80	98.32	97.44	97.48
thumb_d	76.19	98.35	99.59	99.59	99.59
thumb_u	77.67	100.00	97.96	99.17	97.96
thumb_ud	48.42	97.94	93.56	97.87	96.67

Table 6.17: F-score results for the proposed solutions over the depthGestRecog database.

- CS_4 is the DLQP+CS+SVM algorithm with $R = 1$ and $P = 4$. This is the original configuration of the DLQP+CS+SVM algorithm.
- CS_8 is the DLQP+CS+SVM algorithm with $R = 1$ and $P = 8$.
- CS_12 is the DLQP+CS+SVM algorithm with $R = 2$ and $P = 12$.
- CS_24 is the DLQP+CS+SVM algorithm with $R = 2$ and $P = 24$.

Table 6.17 shows the F-score results comparing the performance of the DLQP+CS+SVM algorithm with different neighborhood configurations. It can be seen how the better performance in average is achieved with a neighborhood configuration of $R = 2$ and $P = 24$ obtaining a mean F-score value of 97.68%. Decreasing the size of the neighborhood to $R = 2$ and $P = 12$ and to $R = 1$ and $P = 8$ it can be seen how the algorithm performance in mean value just decreases in a 0.2%, which is despicable. Taking into account that the decrease in the size of the reduced vectors is considerably, from 7717 elements in the configuration of $R = 2$ and $P = 24$ to 2,905 elements in the configuration of $R = 1$ and $P = 8$, implying a reduction in the computational cost, it is worth using the configuration with $R = 1$ and $P = 8$. Finally, it is noticeable the increase of the performance when the CS algorithm is

used with a neighborhood configuration of $R = 1$ and $P = 4$ with respect to the solution obtained when the CS is not used, as the mean F-score value increases in a value of 30% (F-score mean value of 96.59% using the CS versus a value of 68.46% without using it). So again, the use of a CS algorithm helps the solution to obtain better results.

6.4.2.2 Dynamic hand gesture recognition using Leap Motion information

The Real-Time-SVM (Section 4.4.2.1) and Real-Time-NN (Section 4.4.2.2) proposed algorithms have been evaluated using the MultiModHandGestRec database (Section 5.3.3) and compared with other state-of-the-art solutions [124] [127] [104] [136] in the literature. All the solutions use information acquired by the Leap Motion device. The solutions [124] [127] [136] just use the skeleton information, while only the solution presented in [104] uses the near-infrared imagery (as the ones proposed in this thesis). Under this perspective, the purpose of the comparison is to evaluate the performance of every algorithm regarding the hand gesture recognition task, where each of them uses the data for which it was designed (either skeleton or imagery data). The proposed solutions and the one introduced in [104] just make use of the near-infrared imagery as input for the recognition process, while the other works just make use of the skeletal information. Both kind of information are synchronous, so a skeleton and an infrared image are obtained for every acquisition step. The database has been randomly split into a training set and a testing set composed of 80% and 20% of the samples, respectively.

The following parameters have been used for the configuration of each Real-Time solution:

- For the HOG descriptor (used in the Real-Time-SVM, Real-Time-NN, and HOG-NN solutions), $N_{bl} = 16$ blocks, $N_c = 4$ cells, and $N_{bi} = 9$ bins are used.
- For the LBP descriptor (used in the Real-Time-SVM, Real-Time-NN, and LBP-NN solutions), a neighborhood of 9 pixels, and 8 bits to characterize the neighborhood differences are used.
- The image is divided into $N_r = 5 \times 5 = 25$ non-overlapping regions.
- For the Real-Time-SVM a linear kernel for the SVM is used. The number of clusters used for the K-means process is 17. This value has been obtained by experimentation. Figure 6.5a shows the evolution of precision, recall, and F-score values with respect to the number of K-means clusters for the hand pose recognition. Observe that the K-means clustering strategy improves the values of precision, recall, and F-score from around 93% (without clustering strategy or zero clusters) up to near 97%. Similarly, Figure 6.5b represents the evolution of precision, recall and F-score values for the hand

	Real-Time-NN	Real-Time-SVM	HOG-NN	LBP-NN	[104]	[124]	[136]	[127]
Mean	97.49	96.94	95.06	84.97	66.94	62.10	75.85	45.80
l	96.67	96.15	96.15	89.80	74.42	90.77	95.90	20.32
fist m	100.00	97.14	98.11	86.32	65.16	16.95	22.29	0.00
index	97.04	98.11	95.15	87.50	58.65	91.43	89.78	91.04
ok	100.00	97.14	92.00	85.11	68.03	49.32	86.63	65.45
c	93.33	95.15	95.15	91.09	67.64	0.00	57.35	35.37
heavy	96.67	96.15	93.07	81.32	61.38	91.87	91.90	70.53
hang	93.33	99.07	97.14	81.32	63.65	92.62	87.38	87.58
two	100.00	98.11	94.12	77.27	66.17	96.10	94.19	47.79
three	97.04	98.11	94.23	72.94	58.90	94.01	90.13	40.60
four	94.55	95.15	94.12	63.29	71.97	0.00	86.94	9.11
five	97.04	97.14	94.12	87.50	80.40	60.00	31.81	36.02
go down	100.00	97.14	98.11	93.07	—	—	—	—
go left	96.67	97.20	97.20	96.30	—	—	—	—
go right	100.00	95.15	93.07	89.80	—	—	—	—
go up	100.00	97.14	94.12	92.00	—	—	—	—

Table 6.18: Hand gesture recognition performance comparison of F-score value for the proposed algorithms and state-of-the-art works over the MultiModHandGestRec database.

gesture recognition using different number of K-means clusters. Again, an improvement is obtained using the cluster strategy.

- For the Real-Time-NN, a neural network with three fully connected layers has been used. For the two first layers a rectifier linear unit activation function has been used, and for the last one a softmax activation function. A dropout of 50% has been used along the net to avoid overfitting problems in the training process.

Table 6.18 shows F-score values for different hand gestures (static and dynamics) using the proposed algorithms and different algorithms of the state-of-the-art. The comparison with the solutions presented in [104], [124], [136], and [127] is made considering only static hand gestures, since none of them is able to recognized dynamic hand gestures (unlike the proposed system that can recognize both of them). This is the reason why a hyphen appears in those cells of the table (indicating no recognition capability for dynamic hand gestures). As it can be observed, the proposed solutions, Real-Time-SVM, Real-Time-NN, HOG-NN (which is the Real-Time-NN solution but using only the HOG descriptor instead of combining HOG and LBP descriptors), and LBP-NN (which is the Real-Time-NN solution but using only the LBP descriptor instead of combining HOG and LBP descriptors), are the ones that achieves better results. Results obtained by the Real-Time-NN system are slightly better, boosted by the fact that some of the gestures are dynamic and the Real-Time-NN solution

(a)

(b)

Figure 6.5: Evolution of precision, recall and F-score values with respect to the number of K-means clusters for: (a) the hand pose recognition, and (b) the hand gesture recognition.

Process	Time (ms)	
	Real-Time-SVM	Real-Time-NN
Thresholding	8.15	—
Features computation	6.59	6.59
Classification	16.51	25.37
Dynamic gesture recognition	0.05	0.12
Total system	31.3	32.08

Table 6.19: Time performance of the Real-Time-SVM and the Real-Time-NN solutions.

obtains better results than the Real-Time-SVM one for these gestures. These results are also better than those obtained by solutions that only use the LBP or the HOG descriptors in combination with the NN algorithm, proving their performance. Notice that the performance of some gestures using the solutions presented in [124] and [127] are zero or close to zero. This is because for those gestures the Leap Motion is not able to identify the location of the information needed to obtain the feature descriptors for those methods (i.e. the skeleton could not be computed).

The computational time of the Real-Time-SVM and Real-Time-NN systems has been measured to prove its real-time operation using a PC with an i7-3540M processor at 3 GHz and 12 GB of RAM. Overall computational time and per stage has been measured (see Table 6.19). In the case of the Real-Time-SVM, the thresholding process takes 8.15 milliseconds (ms), the computation of HOG and LBP feature vectors 6.59 ms, the hand pose classification 16.51 ms, and the recognition of the dynamic hand gesture 0.05 ms. The total process takes 31.3 ms to process each frame, which allows the system to work at a frame rate close to 30 fps, and therefore to operate in real-time. In the case of the Real-Time-NN, the computation of HOG and LBP feature vectors 6.59 ms, the hand pose classification 25.37 ms, and the recognition of the dynamic hand gesture 0.12 ms. The total process takes 32.08 ms to process each frame, which allows the system to work at a frame rate close to 30 fps, and therefore to operate in real-time. For both cases the obtained value is a mean among a total of 100 repetitions, each repetition is the recognition process of one dynamic gesture.

6.5 Integration into SAVIER project

This thesis has been developed within the project SAVIER, whose main goal is the integration of all the research activity within the Airbus Defense & Space's demonstrator which emulates a Ground Control Station (GCS) from which a simulated Medium-Altitude Long Endurance (MALE) UAV is controlled. The real-time hand gesture recognition system called Real-Time-NN (Section 4.4.2.2) has been selected for its integration in the demonstrator

since is able to robustly recognize multiple static and dynamic hand gestures in real time, according to the results described in Section 6.4.2. A visual feedback has been included to indicate which gesture has been recognized. Once the hand gesture is recognized, the system communicates the underlying commands/actions to the GCS, which are not only for the control, management, and pilot the UAV, but also for controlling the payload of the UAV.

From the point of view of the computational cost, locating/detecting the hands in a flow of video is one of the main bottlenecks. Although in some state-of-the-art solutions, using Leap Motion, this problem was solved by using skeletal information, they introduce some problems and restrictions. The main one is that, in some cases, the hand skeleton provided by the Leap Motion is erroneous, specially when the hand is not placed facing the sensor. To overcome this limitation, our implementation analyzes the raw infrared images. No sliding window techniques are needed and all the active area of the sensor is useful. This algorithm is firstly trained with all the hand gestures to be recognized, and then tested with the new incoming ones. Regarding acquisition, the Leap Motion controller has been used due to its ability to naturally reduce the image clutter.

The communication between the developed software and the demonstrator is via messages that report about the recognized gesture. Each gesture is associated to a different GCS demonstrator command. The message-passing based communication is accomplished via the STANAG4586 format. The commands can be grouped into two main families: the ones to control the payload, and those to command the vehicle (see Section 5.3.4). When the recognition system detects a known gesture, a message is sent to the demonstrator which interprets it and sends the corresponding command to the UAV. For the communication process, two different scenarios are considered (see Figure 6.6): the first when the recognized gesture is one to command the vehicle, in that case, a User Datagram Protocol (UDP) based message with the necessary information is sent to the GCS demonstrator indicating the action that the UAV needs to perform; the second when the recognized gesture is related to the payload management, in that case, a Data Distribution Service (DDS) based message is sent to the GCS demonstrator. When the message is received by the GCS demonstrator an action is performed and a change in the UAV flight or on the camera position is made.

In the demonstrator, the sensor is placed over the table where the keyboard is usually located (see Figure 6.7). The operator will perform the gestures over the sensor inside a small virtual cube of around 1 m^3 . The operator has to perform an initial training of the recognition system by performing several times the considered gestures. This training will be ideally performed only once. This process is important to allow a better recognition rate of the system. Nonetheless, it will not be necessary in a future deployment for the same operator.

Figure 6.6: Block diagram of the connection between the real-time hand gesture recognition system and the GCS demonstrator.

	Real-Time-NN	Real-Time-SVM	HOG-NN	LBP-NN
mean	93.79	66.87	93.58	70.50
cam_b	100.00	79.43	99.46	90.06
cam_f	100.00	34.15	99.41	88.62
cam_l	90.40	40.58	91.62	85.23
cam_r	91.98	76.67	91.98	90.22
down	100.00	36.52	100.00	89.27
ele	99.48	79.78	100.00	0.00
left	97.46	90.68	96.97	86.32
right	92.50	87.88	93.83	55.46
up	96.22	92.90	97.80	79.28
zoom_in	81.36	89.05	78.16	77.19
zoom_out	82.30	27.98	80.20	33.85

Table 6.20: Hand gesture recognition performance comparison on F-score value of the real-time solutions.

Figure 6.7: Leap Motion sensor position over the GCS.

Figure 6.8 shows the ROC curves for each subject of the LRRFaceD database, using the BAG-D3P feature descriptor. It can be observed that for all the classifiers the curve is close to the optimal one. The worst AUC value is 95.16% that corresponds to zoom out gesture, and the mean AUC value is 98.16%, indicating that all classifiers work really good.

Table 6.20 shows the results, in F-score values, of the performance of the four real-time solutions, Real-Time-SVM, Real-Time-NN, HOG-NN (which is the Real-Time-NN solution but using only the HOG descriptor instead of combining HOG and LBP descriptors), and LBP-NN (which is the Real-Time-NN solution but using only the LBP descriptor instead of combining HOG and LBP descriptors), over the leapHandGestUAV database (Section 5.3.4). These are the solutions that were included in the SAVIER demonstrator. Comparing the obtained results over the MultiModHandGestRec previously presented in Table 6.18 with the ones presented in Table 6.20, it can be seen the F-score values of the second one are slightly reduced. This is because the database used for GCS demonstrator is more complex and the number of dynamic gestures has considerably increased, but still now, the F-score mean value is higher than 90% for the Real-Time-NN solution. The performance of the system combining HOG and LBP feature descriptors works better than just using one of them separately, indicating that it is worth combining both, although HOG feature descriptor has more discriminative information. Comparing the NN solution with the SVM one, it can be seen that when the database is more complex, the NN-based solution obtains considerably better results than the ones achieved by the SVM-based solution.

Below, some visual results are shown to indicate how the visual information is displayed to the demonstrator operator. It is important to notice that the system is not always in recognition mode. The user indicates by a foot switch when the system should start the

Figure 6.8: ROC curves for each classifier of the Real-Time-NN algorithm using the leapHandGestUAV database.

hand gesture recognition. When the foot switch is pressed, a small green square is overlaid on the image. When a gesture is recognized and a message is being sent to the GCS, no other gesture can be recognized, and a small red square is overlaid on the image. This feedback can be observed in Figure 6.9 to Figure 6.19, where a pictogram overlaid on the real time gesture image appears when the bank of HMMs recognizes one gesture. The end of a gesture is indicated to the system when there is no hand inside the image region to be processed.

Figure 6.9: Take off.

Figure 6.10: Go up.

Figure 6.11: Go down.

Figure 6.12: Go right.

Figure 6.13: Go left.

Figure 6.14: Zoom in.

Figure 6.15: Zoom out.

Figure 6.16: Camera pan right.

Figure 6.17: Camera pan left.

Figure 6.18: Camera tilt up.

Figure 6.19: Camera tilt down.

Figure 6.20 shows an image of the demonstrator environment with a user performing the Take off gesture, and how the system recognizes it. It can be observed that the sensor was placed over the table, and the user performs the gesture at a distance between 10 and 20 cm away from the sensor.

Figure 6.20: Demonstrator environment.

6.6 Conclusions

In this chapter, results of testing the proposed solutions over the created databases have been reported. Comparisons with other state-of-the-art solutions and using other publicly

available databases are also detailed. Solutions of the performed tests of the face recognition solutions indicate that the proposed algorithms work well when the depth information has low resolution but they work even better when the depth information has high resolution which is the objective for which the solutions are designed for. This fact indicates that the proposed solutions are well designed and accomplish the robustness objective.

For the case of the hand gesture recognition solutions, two different scenarios have been considered: one in which depth information is used, and another one for which infrared information is used. In the first one results indicate that the proposed solutions obtain really good results, and in the case of the increase in the number of gestures in the database, the system is able to distinguish among them with a high performance indicating again the robustness of the solutions. In the case of the use of infrared imagery, also good performance results are obtained while real-time performance is also achieved.

Finally, it was shown that the integration of the designed solution in the SAVIER demonstrator was a success as a specific hand gesture database was well designed for the specific task, the system works on real-time, and the connection with the demonstrator was made successfully.

Chapter 7

Conclusions and future work

7.1 Conclusions

In the last years, the number of works proposing new HMI solutions has increased considerably, as a consequence of the demand for new ways of human interaction to communicate with robots, consumer electronic devices, and virtual reality or augmented reality environments. In these fast evolving scenarios, different human parts can be used to perform the interaction such as voice, hands, face, arms, and so on. In all cases, the common premise is the use of contact-less devices to perform the interaction. For this reason, many solutions have proposed to use visual information with the aim of performing the interaction in a non-intrusive way.

Regarding human face recognition applied to HMI, the main field of application is security, where a person is identified to allow or deny the entrance to a certain area, unlock the user's laptop or mobile, etc. In this thesis, two different solutions for face recognition, specifically designed to exploit depth imagery, have been proposed, which outperforms other techniques thanks to the design of specific feature descriptors. These descriptors are adapted to the specific characteristics of depth imagery and also to the structure of the human face. Another important contribution is the recognition capability from different visual perspectives, unlike most of the existing systems that are restricted to frontal faces. Last but not least, the creation of two depth-based face databases represents an important step, not only for evaluating the proposed algorithms, but also to allow other researchers to test their own systems, especially in the current situation of lack of depth face databases. On overall, the main conclusion is that the specific design of depth and face visual descriptors can boost the discriminative capability of face recognition systems. The design of the solutions for face recognition was made considering that the robustness of the system were more important than the real-time performance. The evolution of the presented solutions was made by

increasing the complexity of the feature descriptor trying to capture more relevant information from high resolution depth imagery provided by the Kinect v2 device. As a result, the accuracy of the systems increased.

In the case of the hand gesture recognition, this thesis has proposed different algorithms that target specific objectives: recognition of static gestures and recognition of both static and dynamic gestures. Another dimension to be considered is the trade-off between robustness and real-time performance of the system. This trade-off has been achieved by combining the design of adapted feature descriptors for the hand gesture recognition using depth information with a Compressive Sensing framework to reduce the length of the highly discriminative designed feature descriptors. The inclusion of such dimensionality reduction technique not only favors real-time performance, but also improves the accuracy, alleviating the so-called problem of the curse of dimensionality. Also, an evolution of the hand gesture recognition system has been proposed to operate in real-time with hardware of limited computational capability. It can recognize both static and dynamic gestures, allowing to cover a high variety of scenarios. This system was included into the SAVIER demonstrator for the interaction with a GCS to pilot an UAV and manage its payload in real scenarios. Finally, data augmentation techniques played an important role, increasing considerably the number of samples to train the system, and thus improving the system performance. Initial hand gesture recognition solutions considered more important the robustness of the system than the account real-time performance. This fact led to use the same solutions designed for face recognition as for hand gesture recognition with only small modifications. But to fulfill real-time restrictions required in the second period of the project more oriented to the demonstrator, new solutions were designed to reduce the computational cost. Those solutions changed depth information for near-infrared one and were focused on reducing the complexity of the designed feature descriptors. Another important fact was the aim of detecting dynamic hand gestures which implies the design of a more complex system. As a result, the final system is able to work in real-time, recognizing static and dynamic hand gestures in few milliseconds, while obtaining a high performance.

7.2 Future work

Analyzing the overall performance of the proposed algorithms, and taking into account the needs of HMI applications, several lines of future have been considered:

- In the solutions presented along this thesis, only one kind of visual information has been considered for each system, either depth or infrared. The use of several types of visual information (color, depth, and infrared) simultaneously can lead to greater improvements in the recognition tasks.

- The combination of different devices can also improve the reliability of the systems, in both face and hand gesture recognition, obtaining information from different points of view of the hand. This could be achieved, for example, by placing one sensor on the table and other in front of the user.
- For the case of the hand gesture recognition, the interaction is performed only with one hand. An improvement can be achieved by utilizing two hands, allowing a low inter-sing interference value as the number of gestures in the vocabulary increases.
- The use of the hand and the face can be combined in HMI systems with other body parts, even with the voice. This can lead to a better communication or introducing redundancy to increase the system reliability.

Appendix A

Leap Motion hand reflectance characterization

The intensity values of the hand regions in the near-infrared images acquired by the Leap Motion depend on the amount of infrared light reflected by the hands, which in turn depends on the distance between the hands and the sensor (where the infrared LEDs are mounted). In order to improve the hand location and its segmentation from the background (which are tasks involved in the hand recognition system), a study about the relationship between the intensity level of the hand regions and the hand-sensor distance has been conducted. For this purpose, the palm of the hand has been placed parallel to the upper face of the sensor at different distances, acquiring a set of images per every measured distance. Next, all the acquired images have been manually annotated indicating the hand location. This information has been used to compute the mean and variance of the intensity value of the hand regions as a function of the distance to the sensor. Figure A.1 shows a graphic that plots the mean and variance of the intensity value according to the distance. As expected, the hand intensity value decreases with the distance following a quadratic polynomial model: $I = 180 - 0.17 \times d - 0.00035 \times d^2$, where I is the intensity value in the range $[0, 255]$, and d represents the distance in millimeters between the hand and the sensor.

As can be observed from the mean values in Figure A.1, the active area of the Leap Motion sensor is between 100 mm and 400 mm. Below 100 mm, the sensor is blind by the hand reflection. And beyond 400 mm, the sensor can not distinguish the hand from the background.

Regarding the standard deviation values, the behavior is more complex. At the beginning, the values decrease with the distance, but then the values increase from 200 mm up to 280 mm to decrease again from that distance. This behaviour can be explained by the effect of the automatic gain control module of the cameras.

(a) Mean value.

(b) Standard deviation value.

Figure A.1: Mean and variance intensity value of the hand regions with respect to the distance to the sensor.

Bibliography

- [1] T. Mantecón, C. R. del Blanco, F. Jaureguizar, and N. García. Depth-based face recognition using local quantized patterns adapted for range data. In *IEEE International Conference on Image Processing (ICIP)*, pages 293–297, Oct. 2014.
- [2] T. Mantecón, C. R. del Blanco, F. Jaureguizar, and N. García. Access control based on visual face recognition using Depth Spatiograms of Local Quantized Patterns. In *IEEE International Conference on Consumer Electronics (ICCE)*, pages 530–531, Jan. 2015.
- [3] T. Mantecón, C. R. del Blanco, F. Jaureguizar, and N. García. Visual face recognition using bag of dense derivative depth patterns. *IEEE Signal Processing Letters*, 23(6):771–775, Jun. 2016.
- [4] T. Mantecón, C. R. del Blanco, F. Jaureguizar, and N. García. New generation of human machine interfaces for controlling UAV through depth-based gesture recognition. In *Proceedings SPIE*, volume 9084, pages 9084.1 – 9084.11, May 2014.
- [5] A. Mantecón, T. Mantecón, C. R. del Blanco, F. Jaureguizar, and N. García. Gesture-based human-machine interface system using compressive sensing. In *IEEE International Symposium on Consumer Electronics (ISCE)*, pages 1 – 2, Jun. 2015.
- [6] T. Mantecón, A. Mantecón, C.R. del Blanco, F. Jaureguizar, and N. García. Enhanced gesture-based human-computer interaction through a compressive sensing reduction scheme of very large and efficient depth feature descriptors. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, Aug. 2015.
- [7] T. Mantecón, C. R. del Blanco, F. Jaureguizar, and N. García. Hand gesture recognition using infrared imagery provided by leap motion controller. In *Advanced Concepts for Intelligent Vision Systems (ACIVS)*, pages 47 – 57, Oct. 2016.

- [8] Creative Corporation. Creative Senz3D <http://creative.com/p/web-cameras/creative-senz3d>.
- [9] Microsoft Corporation. Kinect for Xbox ONE <http://www.microsoft.com/en-us/kinectforwindows/>.
- [10] F. Comaschi, S. Stuijk, T. Basten, and H. Corporaal. Rasw: A run-time adaptive sliding window to improve viola-jones object detection. In *International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–6, Oct. 2013.
- [11] J. L. Crowley, F. Bérard, and J. Coutaz. Finger tracking as an input device for augmented reality. In *International Workshop on Automatic Face and Gesture Recognition (WAFG)*, pages 195–200, Jun. 1995.
- [12] T.J. Darrell, I.A. Essa, and A.P. Pentland. Task-specific gesture analysis in real-time using interpolated views. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 18(12):1236–1242, Dec. 1996.
- [13] K. Silanon and N. Suvonvorn. Finger-spelling recognition system using fuzzy finger shape and hand appearance features. In *International Conference on Digital Information and Communication Technology and its Applications (SDIWC)*, pages 419–424, May 2014.
- [14] M.I.A. Lourakis A. Argyros. Three-dimensional tracking of multiple skin-colored regions by a moving stereoscopic system. *Appl Opt*, 43(2):366–378, Jan. 2004.
- [15] A. Argyros and M.I.A. Lourakis. Real-Time Tracking of Multiple Skin-Colored Objects with a Possibly Moving Camera. In *European Conference on Computer Vision (ECCV)*, volume 3023, pages 368–379. May 2004.
- [16] A. Utsumi and J. Ohya. Multiple-hand-gesture tracking using multiple cameras. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 474–478, Jun. 1999.
- [17] M. Isard and A. Blake. CONDENSATION Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 29(1):5–28, Aug. 1998.
- [18] Chenyang Chen, Mingmin Zhang, Kaijia Qiu, and Zhigeng Pan. Real-time robust hand tracking based on camshift and motion velocity. In *5th International Conference on Digital Home.*, pages 20–24, Nov 2014.

BIBLIOGRAPHY

- [19] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, Jul. 2002.
- [20] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893, Jun. 2005.
- [21] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep. 1995.
- [22] Y. I. Lee, S.-H. Oh, H. K. Song, and M. W. Kim. Design rules of multilayer perceptrons. In *Science of Artificial Neural Networks*, volume 1710, pages 329 – 340, Jul. 1992.
- [23] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE assp magazine*, 3(1):4–16, Jan. 1986.
- [24] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *The Annals of Mathematical Statistics*, 41(1):164–171, Feb. 1970.
- [25] Microsoft Corporation. Kinect for Xbox 360 <http://dx.doi.org/10.1007/s10107-010-0420-4>.
- [26] ASUSTeK Computer Inc. Xtion PRO LIVE https://www.asus.com/3D-Sensor/Xtion_PRO_LIVE/.
- [27] Leap Motion. <https://www.leapmotion.com/company>.
- [28] P. Jyothilakshmi, K. R. Rekha, and K. R. Nataraj. A framework for human-machine interaction using Depthmap and compactness. In *International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT)*, pages 462–467, Dec. 2015.
- [29] Y. Li. Multi-scenario gesture recognition using Kinect. In *International Conference on Computer Games (CGAMES)*, pages 126 – 130, Jul. 2012.
- [30] Y. Yusoff, A. H. Basori, and F. Mohamed. Interactive hand and arm gesture control for 2d medical image and 3d volumetric medical visualization. In *International Conference on Cognitive Science*, volume 97, pages 723 – 729, Nov. 2013.
- [31] T. Kopinski, S. Geisler, L. C. Caron, A. Gepperth, and U. Handmann. A real-time applicable 3D gesture recognition system for automobile HMI. In *International Conference on Intelligent Transportation Systems (ITSC)*, pages 2616 – 2622, Oct. 2014.

- [32] T. O. Costalonga, L. M. Avila, L. Muniz, and A. S. Brandao. Gesture-Based Controllers to Guide a Quadrotor Using Kinect Sensor. In *Joint Conference on Robotics: SBR-LARS Robotics Symposium and Robocontrol (SBR LARS Robocontrol)*, pages 109–112, Oct. 2014.
- [33] B. J. Chen, C. M. Huang, T. E. Tseng, and L. C. Fu. Robust head and hands tracking with occlusion handling for human machine interaction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2141–2146, Oct. 2012.
- [34] A. Begalinova and A. Shintemirov. Design of embedded gesture recognition system for robotic applications. In *IEEE International Conference on Application of Information and Communication Technologies (AICT)*, pages 1 – 4, Oct. 2014.
- [35] P. Chochai, T. Mekrungrroj, and T. Matsumaru. Real-time gesture recognition with finger naming by RGB camera and IR depth sensor. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 931 – 936, Dec. 2014.
- [36] A. P. Placitelli and L. Gallo. Low-Cost Augmented Reality Systems via 3D Point Cloud Sensors. In *International Conference on Signal-Image Technology and Internet-Based Systems (SITIS)*, pages 188 – 192, Nov. 2011.
- [37] T. Pham, P. N. Pathirana, H. Trinh, and P. Fay. A Non-Contact Measurement System for the Range of Motion of the Hand. *Sensors*, 15(8):18315 – 18333, 2015.
- [38] D. Han, Z. Wang, W. Xu, and N. Wang. Rehabilitation Training System for the Motor Deficit of Upper Extremity. In *IEEE International Symposium on Autonomous Decentralized Systems (ISADS)*, pages 162 – 166, Mar. 2015.
- [39] Y.-D. Han, S.-H. Saw, and B.-G. Lee. A Multilayer Air-Touch Interactive System with Fingertip Using Depth Camera. In *International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, volume 2, pages 319 – 322, Aug. 2015.
- [40] V. D. Nguyen, M. T. Chew, and S. Demidenko. Vietnamese sign language reader using Intel Creative Sens3D. In *International Conference on Automation, Robotics and Applications (ICARA)*, pages 77 – 82, Feb. 2015.
- [41] X. Yuanrong, W. Qianqian, B. Xiao, C. Yen-Lun, and W. Xinyu. A novel feature extracting method for dynamic gesture recognition based on support vector machine. In *IEEE International Conference on Information and Automation (ICIA)*, pages 437 – 441, Jul. 2014.

- [42] I. Nigam, M. Vatsa, and R. Singh. Leap signature recognition using HOOF and HOT features. In *IEEE International Conference on Image Processing (ICIP)*, pages 5012 – 5016, Oct. 2014.
- [43] F. Kai-Yin, N. Ganganath, C. Chi-Tsun, and C.K. Tse. A Real-Time ASL Recognition System Using Leap Motion Sensors. In *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pages 411–414, Sep. 2015.
- [44] Y. Wang and M. Meng. 3D Facial Expression Recognition on Curvature Local Binary Patterns. In *International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, volume 2, pages 123 – 126, Aug. 2013.
- [45] S. Anith, D. Vaithiyathan, and R. Seshasayanan. Face recognition system based on feature extraction. In *International Conference on Information Communication and Embedded Systems (ICICES)*, pages 660–664, Feb. 2013.
- [46] B. Jun, I. Choi, and D. Kim. Local Transform Features and Hybridization for Accurate Face and Human Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1423–1436, Jun. 2013.
- [47] G. Kayim, C. Sari, and C.B. Akgul. Facial feature selection for gender recognition based on random decision forests. In *Signal Processing and Communications Applications Conference (SIU)*, pages 1–4, Apr. 2013.
- [48] M. Turk and A. Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1):71 – 86, Jan. 1991.
- [49] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang. Local Gabor binary pattern histogram sequence (LGBPHS): a novel non-statistical model for face representation and recognition. In *IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 786 – 791, Oct. 2005.
- [50] T. Ahonen, E. Rahtu, V. Ojansivu, and J. Heikkila. Recognition of blurred faces using Local Phase Quantization. In *International Conference on Pattern Recognition (ICPR)*, pages 1–4, Dec. 2008.
- [51] J. Kannala and E. Rahtu. BSIF: Binarized statistical image features. In *International Conference on Pattern Recognition (ICPR)*, pages 1363 – 1366, Nov. 2012.
- [52] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen. Local Binary Patterns and Its Application to Facial Image Analysis: A Survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 41(6):765–781, Nov. 2011.

- [53] O. Nikisins and M. Greitans. Local binary patterns and neural network based technique for robust face detection and localization. In *Proceedings of the International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6, Sep. 2012.
- [54] H. Tang, B. Yin, Y. Sun, and Y. Hu. 3D face recognition using local binary patterns. *Signal Processing*, 93(8):2190 – 2198, Aug. 2013.
- [55] S. Elaiwat, M. Bennamoun, F. Boussaid, and A. El-Sallam. 3-D Face Recognition Using Curvelet Local Features. *IEEE Signal Processing Letters*, 21(2):172–175, Feb. 2014.
- [56] L. Ding, X. Ding, and C. Fang. Continuous Pose Normalization for Pose-Robust Face Recognition. *IEEE Signal Processing Letters*, 19(11):721–724, Nov. 2012.
- [57] T. Gao, X.L. Feng, H. Lu, and J.H. Zhai. A novel face feature descriptor using adaptively weighted extended LBP pyramid. *Optik - International Journal for Light and Electron Optics*, 124(23):6286 – 6291, Dec. 2013.
- [58] X. Li, W. Hu, Z. Zhang, and H. Wang. Heat Kernel Based Local Binary Pattern for Face Representation. *IEEE Signal Processing Letters*, 17(3):308–311, Mar. 2010.
- [59] K. Jeong, J. Choi, and G.-J. Jang. Semi-Local Structure Patterns for Robust Face Detection. *IEEE Signal Processing Letters*, 22(9):1400–1403, Sep. 2015.
- [60] M. Pamplona, S. Sarkar, D. Goldgof, L. Silva, and O. Bellon. Continuous 3D Face Authentication Using RGB-D Cameras. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 64–69, Jun. 2013.
- [61] S. Rautaray and A. Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, 43(1):1 – 54, Jan. 2015.
- [62] M.-B. Kaâniche. *Gesture recognition from video sequences*. PhD thesis, Université Nice Sophia Antipolis, Nov. 2009.
- [63] T. Thanh H. Tran. How can human communicate with robot by hand gesture? In *International Conference on Computing, Management and Telecommunications (Com-ManTel)*, pages 235 – 240, Jan. 2013.
- [64] M. Tornow, A. Al-Hamadi, and V. Borrmann. A multi-agent mobile robot system with environment perception and HMI capabilities. In *IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 252 – 257, Oct. 2013.

- [65] M. Tornow, A. Al-Hamadi, and V. Borrmann. Gestic-Based Human Machine Interface for Robot Control. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2706 – 2711, Oct. 2013.
- [66] W. Shang, X. Cao, H. Ma, H. Zang, and P. Wei. Kinect-Based Vision System of Mine Rescue Robot for Low Illuminous Environment. *Journal of Sensors*, 1(1):1–10, Feb. 2016.
- [67] S. M. Goza, R. O. Ambrose, M. A. Diftler, and I. M. Spain. Telepresence Control of the NASA/DARPA Robonaut on a Mobility Platform. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 623 – 629, Apr. 2004.
- [68] R. Aditya, V. Namrata, C. Santanu, and B. Subhashis. Recognition of dynamic hand gestures. *Pattern Recognition*, 36(9):2069 – 2081, Sep. 2003.
- [69] T. Kopinski, S. Geisler, and U. Handmann. Gesture-based human-machine interaction for assistance systems. In *International Conference on Information and Automation (ICIA)*, pages 510 – 517, Aug. 2015.
- [70] C.A. Pickering. Gesture recognition driver controls. *Computing Control Engineering Journal*, 16(1):26 – 27, Feb. 2005.
- [71] A. Kuznetsova, L. Leal-Taixe, and B. Rosenhahn. Real-Time Sign Language Recognition Using a Consumer Depth Camera. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 83–90, Dec. 2013.
- [72] N. Pugeault and R. Bowden. Spelling it out: Real-time ASL fingerspelling recognition. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 1114 – 1119, Nov. 2011.
- [73] A. A. Kindiroglu, H. Yalcin, O. Aran, M. Hrúz, P. Campr, L. Akarun, and A. Karpov. Automatic recognition fingerspelling gestures in multiple languages for a communication interface for the disabled. *Pattern Recognition and Image Analysis*, 22(4):527–536, Dec. 2012.
- [74] L. Rioux-Maldague and P. Giguere. Sign Language Fingerspelling Classification from Depth and Color Images Using a Deep Belief Network. In *Canadian Conference on Computer and Robot Vision (CRV)*, pages 92 – 97, May 2014.
- [75] Y. Wang and R. Yang. Real-time hand posture recognition based on hand dominant line using Kinect. In *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pages 1 – 4, Jul. 2013.

- [76] W. S. Ng and E. Sharlin. Collocated interaction with flying robots. In *IEEE International symposium on robot and human interactive communication (RO-MAN)*, pages 143 – 149, Aug. 2011.
- [77] V.M. Monajjemi, J. Wawerla, R. Vaughan, and G. Mori. HRI in the sky: Creating and commanding teams of UAVs with a vision-mediated gestural interface. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 617–623, Jan. 2013.
- [78] G. Jones, N. Berthouze, R. Bielski, and S. Julier. Towards a situated, multimodal interface for multiple UAV control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1739 – 1744, Jul. 2010.
- [79] C. Meitinger and A. Schulte. Human-UAV Co-operation Based on Artificial Cognition. In *Engineering Psychology and Cognitive Ergonomics (EPCE)*, volume 5639, pages 91 – 100. 2009.
- [80] T. Ma, W. Wee, C. Han, and X. Zhou. A Method for Single Hand Fist Gesture Input to Enhance Human Computer Interaction. In *Human-Computer Interaction. Interaction Modalities and Techniques (HCI)*, volume 8007, pages 291 – 300, Jul. 2013.
- [81] R. A. Bolt and E. Herranz. Two-handed Gesture in Multi-modal Natural Dialog. In *Annual ACM Symposium on User Interface Software and Technology (UIST)*, pages 7 – 14, Nov. 1992.
- [82] G. M. Smith and M. C. schraefel. The Radial Scroll Tool: Scrolling Support for Stylus- or Touch-based Document Navigation. In *ACM Symposium on User Interface Software and Technology (UIST)*, pages 53 – 56, Oct. 2004.
- [83] M. Moyle and A. Cockburn. Gesture navigation: an alternative 'back' for the future. In *Extended Abstracts on Human Factors in Computing Systems (CHI)*, pages 822 – 823, Apr. 2002.
- [84] J. W. Davis and S. Vaks. A Perceptual User Interface for Recognizing Head Gesture Acknowledgements. In *Workshop on Perceptive User Interfaces (PUI)*, pages 1 – 7, Nov. 2001.
- [85] R. Agrawal and N. Gupta. Real Time Hand Gesture Recognition for Human Computer Interaction. In *International Conference on Advanced Computing (IACC)*, pages 470–475, Feb. 2016.

- [86] A. Haria, A. Subramanian, N. Asokkumar, S. Poddar, and J. S. Nayak. Hand Gesture Recognition for Human Computer Interaction. volume 115, pages 367 – 374, Aug. 2017. International Conference on Advances in Computing & Communications (ICIIP).
- [87] H.-S. Yeo, B.-G. Lee, and H. Lim. Hand tracking and gesture recognition system for human-computer interaction using low-cost hardware. *Multimedia Tools and Applications*, 74(8):2687 – 2715, Apr. 2015.
- [88] C. Gratzel, T. Fong, S. Grange, and C. Baur. A Non-contact Mouse for Surgeon-computer Interaction. *Technology Health Care*, 12(3):245 – 257, Aug. 2004.
- [89] E. Lesnes-Cuisiniez, J. Z. Flores, and J.-P. Radoux. Gesture ToolBox: Touchless Human-Machine Interface Using Deep Learning. In *Advances in Artificial Intelligence (KI)*, pages 323 – 329, Sep. 2017.
- [90] G. Modanwal and K. Sarawadekar. Towards hand gesture based writing support system for blinds. *Pattern Recognition*, 57:50 – 60, Sep. 2016.
- [91] A. Paiva, G. Andersson, K. Höök, D. Mourão, M. Costa, and C. Martinho. Sen-Toy in FantasyA: Designing an Affective Sympathetic Interface to a Computer Game. *Personal and Ubiquitous Computing*, 6(5-6):378 – 389, Dec. 2002.
- [92] Eyetoy (2003) <http://asia.gamespot.com/eyetoy-play/>.
- [93] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff, A. Topalian, E. Wood, S. Khamis, P. Kohli, S. Izadi, R. Banks, A. Fitzgibbon, and J. Shotton. Efficient and Precise Interactive Hand Tracking Through Joint, Continuous Optimization of Pose and Correspondences. *ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 143:1–143:12, Jul. 2016.
- [94] S. Bambach, S. Lee, D. J. Crandall, and C. Yu. Lending A Hand: Detecting Hands and Recognizing Activities in Complex Egocentric Interactions. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1949 – 1957, Dec. 2015.
- [95] D.-A. Huang, M. Ma, W.-C. Ma, and K. M. Kitani. How do we use our hands? Discovering a diverse set of common grasps. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 666–675, Jun. 2015.
- [96] C. Li and K. M. Kitani. Pixel-Level Hand Detection in Ego-centric Videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3570 – 3577, Jun. 2013.

- [97] T. Starner, J. Auxier, D. Ashbrook, and M. Gandy. The gesture pendant: a self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring. In *International Symposium on Wearable Computers (ISWC)*, pages 87 – 94, Oct. 2000.
- [98] N. Osawa, K. Asai, and Y. Y. Sugimoto. Immersive Graph Navigation Using Direct Manipulation and Gestures. In *ACM Symposium on Virtual Reality Software and Technology (VRST)*, pages 147–152, Oct. 2000.
- [99] Kai N. and Rainer S. Pointing gesture recognition based on 3D-tracking of face, hands and head orientation. In *Workshop on Perceptive User Interfaces*, pages 140 – 146, Nov. 2003.
- [100] C. Hardenberg and F. Birard. Bare-Hand Human-Computer Interaction. In *ACM Workshop on Perceptive User Interfaces (PUI)*, pages 1 – 8, Nov. 2001.
- [101] X. Cao and R. Balakrishnan. VisionWand: Interaction Techniques for Large Displays Using a Passive Wand Tracked in 3D. In *Annual ACM Symposium on User Interface Software and Technology (UIST)*, pages 173 – 182, Nov. 2003.
- [102] J. Ou, S. R. Fussell, X. Chen, L. D. Setlock, and J. Yang. Gestural Communication over Video Stream: Supporting Multimodal Interaction for Remote Collaborative Physical Tasks. In *International Conference on Multimodal Interfaces (ICMI)*, pages 242 – 249, Nov. 2003.
- [103] Y. Pei and G. Ma. Fast 3D hand estimation for mobile interactions. In *International Conference on Pattern Recognition (ICPR)*, pages 979 – 984, Dec. 2016.
- [104] R. Zhang, Y. Ming, and J. Sun. Hand gesture recognition with SURF-BOF based on Gray threshold segmentation. In *International Conference on Signal Processing (ICSP)*, pages 118 – 122, Nov. 2016.
- [105] C. W. Ng and S. Ranganath. Real-time gesture recognition system and application. *Image and Vision Computing*, 20(1314):993 – 1007, Dec. 2002.
- [106] H. Deng-Yuan, H. Wu-Chih, and C. Sung-Hsiang. Gabor filter-based hand-pose angle estimation for hand gesture recognition under varying illumination. *Expert Systems with Applications*, 38(5):6031 – 6042, Nov. 2011.
- [107] F. Klompmaker, K. Nebe, and A. Fast. dSensingNI: a framework for advanced tangible interaction using a depth camera. In *International Conference on Tangible, Embedded and Embodied Interaction (TEI)*, pages 217 – 224, Feb. 2012.

- [108] P. Breuer, C. Eckes, and S. Müller. Hand gesture recognition with a novel IR time-of-flight range camera - a pilot study. In *International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications (MIRAGE)*, pages 247 – 260. Mar. 2007.
- [109] S. Poularakis, G. Tsagkatakis, P. Tsakalides, and I. Katsavounidis. Sparse representations for hand gesture recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3746 – 3750, May 2013.
- [110] P. Cirujeda and X. Binefa. 4DCov: A Nested Covariance Descriptor of Spatio-Temporal Features for Gesture Recognition in Depth Sequences. In *International Conference on 3D Vision (3DV)*, volume 1, pages 657–664, Dec. 2014.
- [111] A. Boyali and M. Kavakli. A robust gesture recognition algorithm based on Sparse Representation, random projections and Compressed Sensing. In *IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 243 – 249, Jul. 2012.
- [112] J. Wan, Q. Ruan, W. Li, and S. Deng. One-shot Learning Gesture Recognition from RGB-D Data Using Bag of Features. *Journal of Machine Learning Research*, 14:2549 – 2582, Sep. 2013.
- [113] R. Ptucha and A. Savakis. LGE-KSVD: Flexible Dictionary Learning for Optimized Sparse Representation Classification. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 854 – 861, Jun. 2013.
- [114] S.-Z. Li, B. Yu, W. Wu, S.-Z. Su, and R.-R. Ji. Feature learning based on SAE-PCA network for human gesture recognition in RGBD images. *Neurocomputing*, 151(0):565 – 573, Mar. 2015.
- [115] M. Van den Bergh and L. Van Gool. Combining RGB and ToF cameras for real-time 3D hand gesture interaction. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 66 – 72, Jan. 2011.
- [116] M. Oberweger, P. Wohlhart, and V. Lepetit. Training a Feedback Loop for Hand Pose Estimation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3316 – 3324, Dec. 2015.
- [117] D. Tang, J. Taylor, P. Kohli, C. Keskin, T.-K. Kim, and J. Shotton. Opening the Black Box: Hierarchical Sampling Optimization for Estimating Human Hand Pose. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3325 – 3333, Dec. 2015.

- [118] S. Khamis, J. Taylor, J. Shotton, C. Keskin, S. Izadi, and A. Fitzgibbon. Learning an efficient model of hand shape variation from depth images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2540 – 2548, Jun. 2015.
- [119] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt. Fast and robust hand tracking using detection-guided optimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213 – 3221, Jun. 2015.
- [120] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun. Cascaded hand pose regression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 824 – 832, Jun. 2015.
- [121] H. Liang, J. Yuan, and D. Thalmann. Parsing the Hand in Depth Images. *IEEE Transactions on Multimedia*, 16(5):1241 – 1253, Aug. 2014.
- [122] D. G. Santos, B. J. T. Fernandes, and B. L. D. Bezerra. HAGR-D: A Novel Approach for Gesture Recognition with Depth Maps. *Sensors*, 15(11):28646 – 28664, Nov. 2015.
- [123] J. Kim, J. Ryu, and T. Han. Multimodal Interface Based on Novel HMI UI/UX for In-Vehicle Infotainment System. *ETRI Journal*, 37(4):793 – 803, Aug. 2015.
- [124] T. Schmidt, F. P. Araujo, G. L. Pappa, and E. R. Nascimento. Real-Time Hand Gesture Recognition Based on Sparse Positional Data. In *Brazilian Workshop on Computer Vision (WVC)*, pages 243 – 248, Oct. 2014.
- [125] G. Du, P. Zhang, and X. Liu. Markerless Human-Manipulator Interface Using Leap Motion With Interval Kalman Filter and Improved Particle Filter. *IEEE Transactions on Industrial Informatics*, 12(2):694–704, Apr. 2016.
- [126] N. Beattie, B. Horan, and S. McKenzie. Taking the LEAP with the Oculus HMD and CAD - Plucking at thin Air? *Procedia Technology*, 20:149 – 154, Aug. 2015.
- [127] C.-H. Chuan, E. Regina, and C. Guardino. American Sign Language Recognition Using Leap Motion Sensor. In *International Conference on Machine Learning and Applications (ICMLA)*, pages 541–544, Dec. 2014.
- [128] D. Naglot and M. Kulkarni. ANN based Indian Sign Language numerals recognition using the Leap Motion controller. In *International Conference on Inventive Computation Technologies (ICICT)*, volume 2, pages 1–6, Aug. 2016.
- [129] P. Kumar, R. Saini, P. P. Roy, and D. P. Dogra. Study of Text Segmentation and Recognition Using Leap Motion Sensor. *IEEE Sensors Journal*, 17(5):1293–1301, Mar. 2017.

- [130] P. Kumar, R. Saini, P. P. Roy, and D. P. Dogra. 3D text segmentation and recognition using Leap Motion. *Multimedia Tools and Applications*, 76(15):16491 – 16510, Aug. 2017.
- [131] W. J. Li, C. Y. Hsieh, L. F. Lin, and W. C. Chu. Hand gesture recognition for post-stroke rehabilitation using Leap Motion. In *International Conference on Application System Innovation (ICASI)*, pages 386 – 388, May 2017.
- [132] S. Gieser, A. Boisselle, and F. Makedon. Real-Time Static Gesture Recognition for Upper Extremity Rehabilitation Using the Leap Motion. In *International Conference on Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management (DHM)*, volume 9185, pages 144 – 154, Aug. 2015.
- [133] Y. Zou, Y. Chen, M. Gao, Q. Liu, S. Jiang, J. Lu, C. Huang, Z. Li, and D. Zhang. Coronary Heart Disease Preoperative Gesture Interactive Diagnostic System Based on Augmented Reality. *Journal of Medical Systems*, 41(8):126, Jul. 2017.
- [134] W. Lu, Z. Tong, and J. Chu. Dynamic Hand Gesture Recognition With Leap Motion Controller. *IEEE Signal Processing Letters*, 23(9):1188 – 1192, Sep. 2016.
- [135] A. M. Bernardos, J. M. Sánchez, J. I. Portillo, J. A. Besada, and J. R. Casar. A Contactless Identification System Based on Hand Shape Features. volume 52, pages 161 – 168, 2015.
- [136] G. Marin, F. Dominio, and P. Zanuttigh. Hand gesture recognition with Leap Motion and Kinect devices. In *IEEE International Conference on Image Processing (ICIP)*, pages 1565–1569, Oct. 2014.
- [137] P. M. Ferreira, J. S. Cardoso, and A. Rebelo. Multimodal Learning for Sign Language Recognition. In *Iberian Conference Pattern Recognition and Image Analysis (IbPRIA)*, pages 313 – 321, Jun. 2017.
- [138] P. Kumar, H. Gauba, P. P. Roy, and D. P. Dogra. Coupled HMM-based multi-sensor data fusion for sign language recognition. *Pattern Recognition Letters*, 86:1 – 8, Jan. 2017.
- [139] P. Zhang, B. Li, G. Du, and X. Liu. A Wearable-based and Markerless Human-manipulator Interface with Feedback Mechanism and Kalman Filters. *International Journal on Advanced Robot Systems*, 12(12):164–170, Jan. 2015.
- [140] J. Cheng, W. Bian, and D. Tao. Locally regularized sliced inverse regression based 3D hand gesture recognition on a dance robot. *Information Sciences*, 221:274 – 283, Feb. 2013.

- [141] L. Nan, W. Shourong, W. Minfeng, L. Bo, H. Hong, and J. Li. Hand motion recognition based on pressure distribution maps and LS-SVM. In *International Conference on Mechatronics and Control (ICMC)*, pages 1027 – 1031, Jul. 2014.
- [142] H. P. Gupta, H. S. Chudgar, S. Mukherjee, T. Dutta, and K. Sharma. A Continuous Hand Gestures Recognition Technique for Human-Machine Interaction Using Accelerometer and Gyroscope Sensors. *IEEE Sensors Journal*, 16(16):6425–6432, Aug 2016.
- [143] N. Ç. Kiliboz and U. Güdükbay. A hand gesture recognition technique for human-computer interaction. *Journal of Visual Communication and Image Representation*, 28:97 – 104, Apr. 2015.
- [144] W. Guo, P. Yao, X. Sheng, H. Liu, and X. Zhu. A wireless wearable sEMG and NIRS acquisition system for an enhanced human-computer interface. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2192 – 2197, Oct. 2014.
- [145] Y. Han. A low-cost visual motion data glove as an input device to interpret human hand gestures. *IEEE Transactions on Consumer Electronics*, 56(2):501–509, Jul. 2010.
- [146] J. Lobc and P. Trindade. Interactive demonstration of InerTouchHand-iTH a glove device with distributed inertial sensors and vibro-tactile feedback. In *Experiment@ International Conference (ext.at)*, pages 178 – 179, Sep. 2013.
- [147] G. De Pasquale, S. G. Kim, and D. De Pasquale. GoldFinger: Wireless Human Machine Interface with Dedicated Software and Biomechanical Energy Harvesting System. *IEEE/ASME Transactions on Mechatronics*, 21(1):565–575, Feb. 2016.
- [148] G. Ponraj and H. Ren. Sensor Fusion of Leap Motion Controller and Flex Sensors Using Kalman Filter for Human Finger Tracking. *IEEE Sensors Journal*, 18(5):2042 – 2049, Mar. 2018.
- [149] N. Kumar. Motion trajectory based human face and hands tracking for sign language recognition. In *IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics*, pages 211–216, Oct. 2017.
- [150] G. A. Rao and P. V. V. Kishore. Selfie sign language recognition with multiple features on adaboost multilabel multiclass classifier. *Journal of Engineering Science and Technology*, 13(8):2352–2368, 2018.
- [151] P. Kumar, P. P. Roy, and D. P. Dogra. Independent Bayesian classifier combination based sign language recognition using facial expression. *Information Sciences*, 428:30 – 48, 2018.

- [152] A. S. Elons, M. Ahmed, and H. Shedid. Facial expressions recognition for arabic sign language translation. In *International Conference on Computer Engineering Systems*, pages 330–335, Dec. 2014.
- [153] R. Berri, D. Wolf, and F. Osório. Telepresence robot with image-based face tracking and 3d perception with human gesture interface using kinect sensor. In *Joint Conference on Robotics: SBR-LARS Robotics Symposium and Robocontrol*, pages 205–210, Oct. 2014.
- [154] B. Seddik, S. Gazzah, and N. E. Ben Amara. Hands, face and joints for multi-modal human-action temporal segmentation and recognition. In *European Signal Processing Conference*, pages 1143–1147, Aug. 2015.
- [155] M. Pateraki, H. Baltzakis, and P. Trahanias. Visual estimation of pointed targets for robot guidance via fusion of face pose and hand orientation. In *IEEE International Conference on Computer Vision Workshops*, pages 1060–1067, Nov 2011.
- [156] Qing Ye, Junfeng Dong, and Yongmei Zhang. 3d human behavior recognition based on binocular vision and face & hand feature. *Optik - International Journal for Light and Electron Optics*, 126(23):4712 – 4717, 2015.
- [157] A. S. Patwardhan. Multimodal mixed emotion detection. In *International Conference on Communication and Electronics Systems*, pages 139–143, Oct. 2017.
- [158] M. N. S. Jahromi, M. B. Bonderup, M. Asadi-Aghbolaghi, E. Avots, K. Nasrollahi, S. Escalera, S. Kasaei, T. B. Moeslund, and G. Anbarjafari. Automatic access control based on face and hand biometrics in a non-cooperative context. In *IEEE Winter Applications of Computer Vision Workshops*, pages 28–36, Mar. 2018.
- [159] S. Han, S. Park, M. Kim, and J. Paik. Hand detection using facial information. In *IEEE International Conference on Consumer Electronics - Berlin*, pages 167–168, Sep. 2016.
- [160] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51 – 59, Jan. 1996.
- [161] E. Choi and C. Lee. Feature extraction based on the Bhattacharyya distance. *Pattern Recognition*, 36(8):1703 – 1709, Aug. 2003.
- [162] Y. C. Eldar and G. Kutyniok. *Compressed sensing: theory and applications*. Cambridge University Press, 2012.

- [163] D. Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671 – 687, Jun. 2003.
- [164] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Applied and Numerical Harmonic Analysis, 2013.
- [165] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on System Man, and Cybernetics*, 9(1):62 – 66, Jan. 1979.
- [166] R. R. Leighton and B. C. Conrath. The autoregressive backpropagation algorithm. In *International Joint Conference on Neural Networks (IJCNN)*, volume 2, pages 369 – 377, Jul. 1991.
- [167] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257 – 286, Feb. 1989.
- [168] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society, series B*, 39(1):1–38, Dec. 1977.
- [169] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260 – 269, Apr. 1967.
- [170] High Resolution Range based Face Database <https://www.kaggle.com/gti-upm/hrrfaced>.
- [171] Low Resolution Range based Face Database <https://www.kaggle.com/gti-upm/lrrfaced>.
- [172] Depth based Gesture Recognition Database <https://www.kaggle.com/gti-upm/depthgestrecog>.
- [173] Hand Gesture Recognition Database Using Leap Motion <https://www.kaggle.com/gti-upm/leapgestrecog>.
- [174] Multi-modal dataset for Hand Gesture Recognition <https://www.kaggle.com/gti-upm/multimodhandgestrec>.
- [175] Leap Motion Hand Gesture dataset for UAV management <https://www.kaggle.com/gti-upm/leaphandgestuav>.

BIBLIOGRAPHY

- [176] G. Goswami, M. Vatsa, and R. Singh. RGB-D Face Recognition With Texture and Attribute Features. *IEEE Transactions on Information Forensics and Security*, 9(10):1629–1640, Oct. 2014.
- [177] R. Min, N. Kose, and J.-L. Dugelay. KinectFaceDB: A Kinect Database for Face Recognition. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(11):1534 – 1548, Nov. 2014.
- [178] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool. Random Forests for Real Time 3D Face Analysis. *International Journal on Computer Vision*, 101(3):437–458, Feb. 2013.