Treball Final de Màster

MÀSTER EN ENGINYERIA INFORMÀTICA

Segmentació de mans en imatges de profunditat

BERNAT GALMÉS RUBERT

Tutor
Gabriel Moyà Alcover

Universitat oberta de Catalunya
Palma, 14 de març de 2020

FITXA DEL TREBALL FINAL

Fitxa del treball final	
Títol del treball:	Segmentació de mans en imatges de profunditat
Nom de l'autor:	Bernat Galmés Rubert
Nom del consultor/a:	Gabriel Moyà Alcover
Nom del PRA:	Carles Ventura Royo
Data de lliurament (mm/aaaa):	06/2019
Titulació o programa:	Màster Universitari en Enginyeria Informàtica
Àrea del Treball Final:	Intel·ligència artificial
Idioma del treball:	Català
Paraules clau:	Imatges de profunditat, Segmentació de mans, Interacció persona-ordinador, *Randomized Decision Forests*, Temps real

Resum del Treball (màxim 250 paraules):

En aquest document es tracta un mètode de detecció de mans en imatges de profunditat. Consisteix en la classificació dels píxels d'una imatge segons la seva probabilitat de pertànyer a una mà. Per cada píxel es calculen una sèrie de característiques simples i s'obté la seva predicció amb un classificador tipus *Random Forest*. Amb aquest procés s'aconsegueix realitzar les prediccions de les imatges en temps real. L'objectiu del treball és presentar els detalls del seu funcionament i analitzar-lo. A més, a mesura que es vagin detectant problemes, s'aniran suggerint solucions que seran aplicades per anar millorant el model. Dues de les problemàtiques detectades són: un comportament incorrecte al detectar mans en un entorn no controlat i la confusió dels exemples de cares i de mans. Una de les solucions ha estat la creació d'un dataset en un entorn no controlat, i un altre ha estat afegir un major percentatge d'exemples negatius de cares al conjunt d'entrenament. Amb el canvi de dataset s'ha aconseguit una millora notable dels resultats en el nou entorn. En canvi amb l'addició dels exemples negatius no s'ha aconseguit gaire variació. Així i tot, el comportament final del classificador és excel·lent, normalment els casos en què falla són a causa de problemes inherents del mètode, són exemples negatius en els quals les característiques prenen els mateixos valors que els positius.

Fitxa del treball final

Abstract – English:

This document treats a hands detection method using depth information. It is achieved classifying the pixels of the image according to its probability to belong at a hand. A set of simple features are computed for each pixel, and its prediction is obtained using a Random Forest classifier. This way, real-time predictions are achieved. The aim of this work is to present the operation details of the method and analyse its behaviour. Besides, as well as problems are detected, solutions will be suggested to solve them, which will be applied in order to improve the model. Two problems detected are incorrect behaviour predicting hands on non controlled environments and the confusion of hands and face samples. One solution has been the creation of a new dataset composed by images took of a non controlled environment. Another solution has been the append of a major percentage of face samples in the training set. With the dataset change, a notable results improvement is achieved in the target environment. However, only a little variation is achieved adding the faces samples. Anyway, the behaviour of the classifier is excellent, all false predictions are caused by the classifier architecture, samples which features take the same value in some positives and false cases.

SUMARI

ÍNDEX DE FIGURES

ACRÒNIMS

RDF Randomized Decision Forest

RGB *Red, Green, Blue*

RGBD *Red, Green, Blue and Depth*

TOF *Time Of Flight*

TFM Treball Final de Màster

UIB Universitat de les Illes Balears

UOC Universitat Oberta de Catalunya

API *Application Programming Interface*

PAC Prova d'avaluació contínua

NYU New York University

HSV *Hue, Saturation and Value*

GUI *Graphical User Interface*

DBG Dataset Bernat i Gabriel

CAPÍTOL 1

INTRODUCCIÓ

La segmentació de mans en imatges i vídeo és un problema complex que té moltes aplicacions en branques de coneixement com la interacció persona-ordinador o la videovigilància. La seva principal aplicació és com a passa prèvia d'altres tasques, com estimar la posició 3D de les mans [4, 3] o reconèixer els gestos [5]. Entre les seves utilitats més important estan l'automatització de la traducció del llenguatge de signes al parlat [6], que és una estratègia important per comunicar-se amb robots d'una manera natural i fàcil [7]. A més, pot ser usada pels fisioterapeutes per a extreure mesures terapèutiques [8], o per jugar a videojocs. En aquest treball es tracta la detecció de mans en temps real, el que permet ampliar l'aplicació d'aquestes tècniques a la vigilància automàtica, que tindria moltes aplicacions per a la vigilància de seguretat [9].

L'aparició de les càmeres de profunditat "low-cost", com la *Kinect v2*, va obrir un gran ventall d'oportunitats per a solucionar problemes fonamentals de la visió per computador [10]. El seu ús en la segmentació de mans ha ajudat a solucionar el problema en temps real amb més precisió que usant només la informació dels colors. Aquesta tecnologia permet superar limitacions típiques del color al incorporar informació geomètrica de l'escena [11].

Aquest document es centra en la detecció de mans generalitzada a qualsevol escena on apareguin persones, no només quan es tenen les mans en el primer pla de la càmera. Així, amb el suport de la tecnologia de les càmeres de profunditat, es pretén que donada qualsevol escena es pugui identificar on estan totes les mans que hi apareixen.

La seva estructura és com segueix, en aquest capítol s'introdueix el tema tractat. En el 3 es defineixen els detalls de la metodologia utilitzada. En el capítol 4 es detallen els experiments que s'han dut a terme durant l'execució del projecte i en el capítol 5 s'exposen els seus resultats. Per a tancar el document en el capítol 6 es discuteixen els resultats obtinguts i en el capítol 7 s'exposen les conclusions del projecte.

1.1 Estat de l'art

Existeix una gran quantitat de literatura sobre la segmentació de mans: en la majoria només usen la informació del color, però n'hi ha que només usen la profunditat i d'altres que combinen ambdues informacions. D'entre les que usen únicament el color, Zimmermann *et al.* ho aconsegueix fent ús d'una xarxa neuronal convolucional [4], i Qiu *et al.* aplicant l'algorisme d'agrupació *k-means* a l'espai de color YCbCr [12]. També existeixen altres autors que detecten la mà però no la segmenten [13]. Cap d'aquestes tècniques disposa de la suficient exactitud per a ser aplicables al problema tractat, o són massa lentes per a ser executades en temps real. Amb l'aparició dels sensors de profunditat ha aparegut un nou conjunt de mètodes que combinen la informació del color i de la profunditat dels píxels. Aquestes noves tècniques han sobrepassat a totes les mencionades anteriorment, que només usaven la informació del color.

Alguns autors utilitzen el color per a segmentar la pell i finalment usen la informació de profunditat per a refinar la segmentació de la mà. Wen *et al.* segmenta el color de la pell de tota la imatge i després aplica l'algorisme d'agrupació *k-means* amb la informació de profunditat per a identificar la mà [5]. Van *et al.* utilitza la profunditat per a mitigar la superposició de les mans amb altres elements amb color de pell [14]. Altres tècniques, com la de Lin *et al.*, combinen el color de la pell, la profunditat i coes de moviment; aquesta tècnica només és útil per a segmentar mans en moviment en un vídeo [15]. Tots aquests mètodes comparteixen un problema, depenen de la colorimetria de la imatge, poden fallar si es duen guants o amb persones vestint colors semblants a la pell, problema que es coneix com a camuflatge per color [16].

Les metodologies més robustes presents en la literatura només utilitzen la informació de la profunditat per a realitzar la segmentació, el que evita els problemes de la colorimetria de la imatge. Liang *et al.* proposa un mètode que funciona baix la premissa que la mà és l'element més proper a la càmera i altres restriccions morfològiques [17]. Tara *et al.* també assumeix que les mans estan ubicades a davant del cos humà, i no estan ocultes [18]. Aquests mètodes tenen el problema que estan limitats per les seves suposicions, *ie.:* les mans han d'estar a davant el cos, que no es compliran en qualsevol escena en un entorn real.

La tècnica estudiada al present document ha estat usada per molts d'autors, només usa informació de profunditat, i és independent de la seqüència. La seva definició la va fer Shotton *et al.* a [19], on utilitzava un conjunt de característiques simples i un RDF per a identificar 31 diferents parts del cos humà. Existeixen treballs posteriors que utilitzen aquesta tècnica centrant-se en la segmentació de mans. Tompson *et al.* estudia la identificació de la postura de la mà estenent la metodologia de Shotton *et al.* per a segmentar les mans [3], usa dues etiquetes (mà o cos) en lloc de les 31 originals. Finalment, aplica una xarxa neuronal convolucional a sobre de la segmentació per a trobar les articulacions de la mà. Els autors també ofereixen un dataset públic de les imatges *Red, Green, Blue and Depth* (RGBD) amb les mans etiquetades.

Sharp *et al.* també utilitza aquesta tècnica per a segmentar les mans [20]. La seva contribució principal és que comença identificant la regió de la imatge que hauria de contenir la mà, després aplica únicament a sobre d'aquesta regió la metodologia de Shotton *et al.* d'una manera semblant a Tompson *et al.*. Una altra característica d'aquest treball és que també etiqueta l'avantbraç, el que facilita la identificació de la separació de la mà i el braç. Els resultats obtinguts amb aquest mètode tenen una gran exactitud,

i són molt robusts. Ara bé, té una limitació molt important: depèn d'altres tècniques per a trobar la regió inicial on s'ha de cercar la mà. Tant el treball de Sharp *et al.* com el de Tompson *et al.* tenen la problemàtica que confonen els caps de les persones amb les mans. Sharp *et al.* mitiga la problemàtica vestint unes ulleres de sol.

Una altra extensió de l'algorisme és la segmentació de mans interactuant amb objectes. Un problema típic, en les imatges de profunditat, és la dificultat de diferenciar dos objectes que estan en contacte, ja que estan a la mateixa profunditat, aquest problema es coneix com a camuflatge per profunditat [16]. Kang *et al.* soluciona aquest problema utilitzant dos RDF seqüencials [21], el primer identifica la regió que conté l'objecte i la mà, i el segon segmenta la mà de l'objecte. El problema d'aquesta solució és que necessita executar dos RDF, duplicant el temps d'execució i els recursos utilitzats, i a més no soluciona la resta de problemes que té el mètode.

1.2 Dades

Hi ha una gran quantitat de datasets disponibles centrats en reconeixement de gestos de les mans. La major part d'ells no proporcionen els píxels de mans etiquetats, que és un requisit del mètode. Un altre requeriment de la tècnica és que el fons de l'escena ha d'estar segmentat, o almenys totes les imatges del dataset han de formar seqüències perquè s'hi pugui aplicar algun algorisme de *Background Substraction* [16] satisfactòriament.

La Taula 1.1 conté un resum dels principals datasets públics considerats. Es pot observar que els únics que compleixen els requisits del projecte són el *NYU Hand Dataset* [3] i el DBG, que tenen les mans segmentades i el fons marcat com a tal. Encara que el *HOI dataset* [21] sembli interessant per avaluar el comportament de l'algorisme en un context d'interacció amb objectes no s'adapta al treball, no té el fons discriminat. El dataset *ICL* [22] és molt complet, però no proporciona les etiquetes de píxels de mans. Ha estat utilitzat en molts de treballs anteriors *ie.* [22]. L'altre dataset que apareix a la taula és *UCI-EGO* [23], el qual té la propietat que les imatges són preses en primera persona, les mans són les del càmera. El seu contingut és molt complet, està format per un conjunt d'escenes independents, les quals contenen interacció amb objectes i diferents gestos de mans. És difícilment adaptable al treball al disposar d'un nombre reduït d'imatges i no tenir el fons segmentat, però és interessant considerar-lo com un punt de vista alternatiu de la metodologia.

Així que el treball es centra en dos datasets. El primer és `NYU Hand Dataset`, utilitzat a [3], i el DBG, elaborat amb dades capturades en un entorn no controlat. Ambdós conjunts de dades estan detallats en la secció 3.6.

1.3 La nostra contribució

La tècnica que es tracta en aquest document és la que està més estesa per a segmentar mans utilitzant només informació de profunditat. Ara bé, cap dels autors que la utilitzen [3, 20, 21] proporciona una justificació per a l'ús de la tècnica. Shotton *et al.* a l'article original de la metodologia [19] simplement especifica que la combinació de les característiques amb un RDF proporcionen la suficient precisió per discriminar totes les parts entrenades i que el disseny de les característiques està motivat per la

Conjunts de dades					
Dataset	Fons segmentat	Etiquetes	Punt de vista	Mida	Seqüències
HOI	No	Si	3ª persona	22174	No
ICL	Si	No	3ª persona	17604	Si
NYU	Si	Si	3ª persona	6716	Si
UCI-EGO	No	No	1ª persona	1457	No
DBG	Si	Si	3ª persona	6260	Si

Taula 1.1: Característiques més important dels datasets públics. La primera columna indica si les imatges disposen del fons segmentat. La segona si les imatges tenen els píxels de mans anotats. La tercera indica des d'on han estat presses les imatges. La quarta és el nombre d'imatges del dataset. I la darrera si el dataset té seqüències d'imatges o si són independents.

seva eficiència computacional. Tampoc s'ha pogut identificar a la literatura cap estudi profund que indiqui com funciona aquesta segmentació.

Aquest document conté un estudi exhaustiu del comportament de la tècnica. L'objectiu principal és identificar els seus punts forts i febles, centrant-se en la seva interpretació. Per a inspeccionar-lo s'analitza en el seu conjunt per identificar patrons i adquirir el coneixement per a entendre el seu comportament. Al mateix temps que es va inspeccionant el mètode se li aniran incorporant canvis per anar millorant el seu comportament.

El mètode utilitzat consisteix en la classificació de cada píxel de les imatges segons la seva probabilitat de ser mà. La probabilitat de cada píxel s'obté calculant una sèrie de característiques associades a ell, i realitzant la predicció amb un RDF.

En aquest treball, a més de presentar en detall el funcionament de la tècnica, s'identificaran tots els problemes que té el mètode. A cada problema identificat, s'estudiarà la seva causa i se li intentarà donar una solució. D'aquesta manera s'anirà millorant el mètode progressivament. Es prendrà com a punt de partida el treball de Tompson *et. al.* [3].

En els treballs anteriors tots els usos que es donava del mètode eren part d'un propòsit específic, per exemple identificar els gestos de les mans. A causa d'això les seves definicions només contemplen els casos en què les mans estan en un primer pla. Es pretén que la implementació resultant d'aquest treball sigui generalitzable al màxim nombre de casos possibles, això s'anirà aconseguint a mesura que es vagi millorant el mètode.

En aquest capítol s'ha introduït el tema tractat. S'ha començat el capítol definint l'estat de l'art del mètode treballat, on s'han exposat tots els treballs anteriors relacionats amb la recerca. Després s'han exposat els conjunts de dades disponibles públicament, especificant els que finalment s'han utilitzat. S'ha tancat el capítol resumint les contribucions aportades pel treball.

CAPÍTOL 2

PLANIFICACIÓ

Tot projecte necessita una definició de com serà dut a terme i del seu contingut per a no navegar a dins del caos. En aquest capítol es presenta com s'ha dut a terme i es detalla el seu contingut.

El projecte començarà amb una primera etapa de planificació. En la que es farà una recerca bibliogràfica sobre l'estat de l'art, es definirà a alt nivell les passes a seguir, l'arquitectura del software i el seu model de desenvolupament.

Una vegada finalitzada la planificació del projecte es començarà l'etapa de desenvolupament. Es durà a terme amb un model de desenvolupament evolutiu, que es basa en l'execució d'iteracions. El resultat de cada iteració és una versió d'un prototipus i un conjunt de resultats associats a aquest. En la darrera iteració es genera la versió final del software i els seus resultats, que seran els resultats final del projecte.

A l'acabar la darrera iteració del desenvolupament, es tancarà el projecte amb una fase de documentació del projecte i extracció de conclusions dels resultats. Els resultats obtinguts estan detallats en el capítol 5.

2.1 Model de desenvolupament

Per a l'execució del projecte s'ha utilitzat un model de desenvolupament de software per prototipus, també anomenat evolutiu. Consisteix en el desenvolupament d'un prototipus inicial que va evolucionant segons les necessitats que van sorgint. Aquesta tècnica és especialment útil quan no es tenen els requisits molt clars, que es van refinant en cada iteració generant una nova versió del prototipus. L'article [1] conté un altre exemple d'ús d'aquesta metodologia.

El desenvolupament es basa en iteracions, cada una comença amb una versió del software i un conjunt de requeriments, i acaben amb una nova versió, resultat de l'evolució de la de partida. L'excepció és la primera iteració, en la que no es fa evolucionar un prototipus, sinó que es desenvolupa la primera versió des de zero. La Figura 2.1 conté l'estructura de cada iteració del model de desenvolupament.

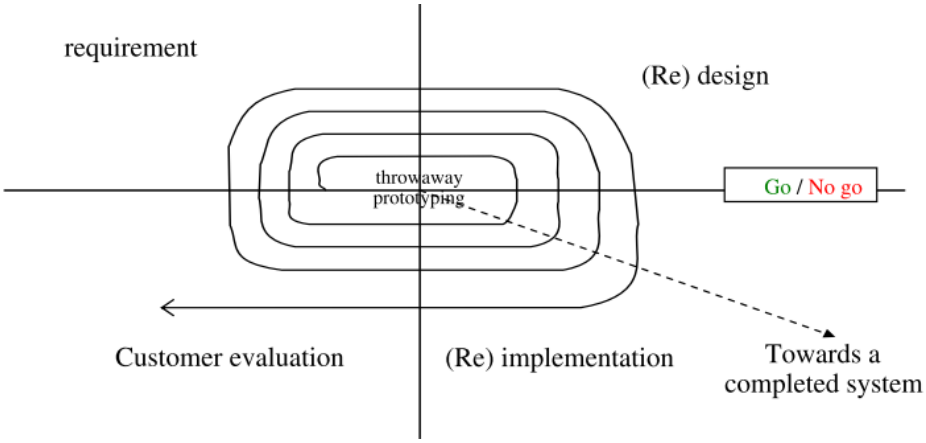


Figura 2.1: Esquema del procés de desenvolupament basat en prototipus [1].

La taula 2.1 conté les fases que composen cada iteració del desenvolupament d'un prototipus, que es corresponen amb les quatre que apareixen en la figura 2.1.

Després d'obtenir una bona base de l'estat de l'art i la planificació a alt nivell de les fases del projecte, s'executarà la primera iteració. En la que es desenvoluparà un primer prototipus del detector de mans. Partint d'aquesta primera versió, s'anirà iterant per fer-lo evolucionar fins que s'aconsegueixi una versió definitiva.

El desenvolupament començarà iniciant la primera iteració per la fase 1, i es donarà per finalitzat en la fase d'avaluació d'una iteració, en la que es decideixi que el prototipus ja compleix tots els requisits i és el definitiu. En aquest projecte, al tenir un temps limitat no s'arribarà a aquest punt, es limitarà l'execució a un nombre fixe d'iteracions.

L'aplicació d'aquesta metodologia en el desenvolupament del projecte està motivada per la falta coneixement dels problemes que apareixeran durant la seva execució. Aquesta incertesa fa que aquest model de desenvolupament sigui el que s'adapti millor a les necessitats del projecte.

2.2 Pla del projecte

Els distints apartats d'aquesta secció defineixen els detalls de com s'ha gestionat el projecte, del contingut del projecte, del treball realitzat al llarg del temps i dels riscos que afecten al projecte.

2.2.1 Guia de gestió del projecte

La gestió del projecte durant el seu cicle de vida ve marcat per dos elements: els rols dels membres participants i les estratègies que es duran a terme per a gestionar-lo.

Membres de l'equip del projecte

L'equip participant en el Treball Final de Màster (TFM) està format per l'autor del treball i per membres de la Universitat de les Illes Balears (UIB) i la Universitat Oberta de Catalunya (UOC).

Fases de cada iteració			
Fase	Descripció	Que inclou	Resultats
1. Anàlisi de la iteració	Selecció dels requisits que s'han d'implementar en la iteració.	Anàlisis de les conclusions i propostes extretes en la avaluació de la iteració immediatament anterior. Establiment dels problemes que s'han de solucionar en la iteració. Selecció de les propostes a implementar en la iteració actual.	Llista de requisits que s'han de complir en la iteració.
2. Disseny de la iteració	Definició de com i que s'ha d'implementar en la iteració per a complir amb els requisits.	Recerca bibliogràfica de solucions als requisits.	Coneixements. Fonts bibliogràfiques.
3. Implementació de la iteració	Implementar una versió del prototipus aplicant els coneixements adquirits en la tasca de disseny.	Construcció del prototipus.	Versió del prototipus.
4. Avaluació de la iteració	Obtenció de resultats que mostrin el comportament de la versió del prototipus. Mètriques, gràfics i imatges. Anàlisis d'aquests resultats per a identificar els punts febles de la versió i proposar possibles solucions a aquests.	Identificar problemes del prototipus.	Versió del prototipus validada

Taula 2.1: Fases que componen cada iteració del desenvolupament.

La Taula 2.2 conté el llistat dels diferents participants en el projecte i la seva funció. La primera columna indica el seu rol, la segona el nom dels membres, la tercera la institució a la qual pertanyen, i la darrera una descripció de la seva funció.

Rols			
Càrrec	Membres	Institució	Descripció
Autor	Bernat Galmés Rubert	UOC i UIB	Director del projecte i desenvolupador.
Responsable del TFM	Carles Ventura Royo	UOC	Professor de la UOC és el responsable darrer de l'execució del TFM.
Tutor del TFM	Gabriel Moyà Alcover	UOC i UIB	Tutela el treball. És el nexe d'unió entre les dues organitzacions universitàries.
Membres de la UIB	Xavier Varona Gómez Antoni Jaume Capó Gabriel Moyà Alcover	UIB	Donen el vist plau a l'execució d'algunes tasques. Són els membres de l'equip de recerca on s'engloba aquest TFM.

Taula 2.2: Taula amb els rols del projecte.

Estratègies de gestió

Per a gestionar el projecte durant la seva execució existeixen tres activitats principals: reunions setmanals, reunions al final de cada iteració i dos informes de seguiment.

Setmanalment, el tutor del TFM i l'autor es reuneixen per discutir l'estat del projecte. En aquestes reunions l'autor comunica al tutor com està avançant el projecte i si han tingut lloc incidències que dificultaran l'acompliment dels objectius de la iteració en el temps preestablert.

Al final de cada iteració, l'equip de la UIB es reuneix amb l'autor. En aquestes reunions l'autor presenta els canvis que s'han fet al nou prototipus i com han afectat als resultats. Entre tots decideixen com ha d'evolucionar el prototipus en la propera iteració.

Finalment, per a formalitzar el seguiment del projecte amb la UOC s'elaboraren dos informes de seguiment. Un a la meitat de l'etapa de desenvolupament, i l'altre al final de l'etapa. En aquests s'exposa l'estat de l'avanç del desenvolupament en aquests dos instants.

2.2.2 Definició detallada de l'abast

Aquesta secció conté la definició del contingut del projecte. Especifica tot el que es considera i tot el que queda exclòs.

Objectius

L'objectiu principal del projecte és obtenir un major coneixement del funcionament de la tècnica per a futurs treballs que requereixin la seva aplicació. A més, en base dels

nous coneixements s'aniran aplicant millores a la tècnica per a aconseguir un detector de mans que vagi millorant els resultats a mesura que avança el desenvolupament.

Requisits

Aquest projecte es caracteritza per a tenir uns requisits volàtils que poden variar al llarg del temps, de fet ho faran en cada iteració del desenvolupament.

Els requisits que es mantindran en tot el projecte són els següents:

- ***RS_001:*** El sistema ha de garantir la capacitat de processar una imatge en un temps inferior a $\frac{1}{60}$ segons. Per a poder ser aplicable a tots els fotogrames de vídeos a 60 fps.
- ***RS_002:*** El sistema ha de garantir una qualitat dels resultats, com a mínim, a l'altura de l'estat de l'art.
- ***RS_003:*** El sistema ha de ser adaptable per a poder garantir l'acompliment dels requisits per a diferents maquinaris. Ha de permetre la configuració dels paràmetres del sistema amb facilitat.

Criteris d'acceptació del projecte

Per avaluar si els resultats són els esperats, s'utilitzen els següents criteris:

- ***C_001:*** Cap imatge del dataset provoca un error en el prototipus.
- ***C_002:*** El prototipus proporciona uns resultats amb una exactitud igual o superior que el prototipus inicial de la iteració.
- ***C_003:*** Es disposa d'un llistat de mètriques i d'imatges amb resultats del prototipus que permeten validar el seu funcionament.
- ***C_004:*** Els membres de l'equip de la UIB han validat el prototipus.

Límits i restriccions del projecte

Hi ha una sèrie d'elements que no inclou el projecte, o seran proveïts externament.

- Les imatges per a comprovar el funcionament del sistema seran obtingudes i processades per l'*Application Programming Interface* (API) que proporciona el fabricant del sensor de profunditat usat.
- El prototipus final no estarà integrat directament amb l'API del sensor de profunditat. Farà falta un software per obtenir els fotogrames de la càmera i el software del prototipus per a obtenir els resultats.
- Únicament es garanteix que els resultats seran vàlids per a imatges obtingudes amb sensors de profunditat tipus *Time Of Flight* (TOF).
- Només s'avaluarà el sistema amb el sensor TOF de la *Kinect v2*.

- Els algorismes coneguts utilitzats provindran d'implementacions existents a la xarxa.

- La implementació descartarà tot el contingut de les imatges que estigui a una distància major de 6 metres.

Hipòtesi de partida

En l'elaboració del projecte s'han considerat els següents supòsits

- Si funciona amb càmera de la *Kinect v2*, el sistema hauria de funcionar amb qualsevol altre sensor TOF.

- S'hauria de poder desenvolupar el producte resultant amb una tecnologia compatible amb l'API del sensor de profunditat i integrar-li.

Activitats de producció del projecte

La taula 2.3 conté la descripció de les activitats a primer nivell del projecte: que fan i que inclouen. Resultats/lliurables de cada una.

2.2.3 Planificació temporal

Per qüestions logístiques de coordinació del seguiment del treball amb la UOC el desenvolupament del treball s'ha estructurat en dues fases. Tot i que la seva organització natural hauria estat en una fase de planificació i una d'iteracions, com està definit a la introducció del capítol. La Figura 2.2 conté l'estructura a alt nivell del projecte amb les seves fites principals, presenta els principals processos del projecte.

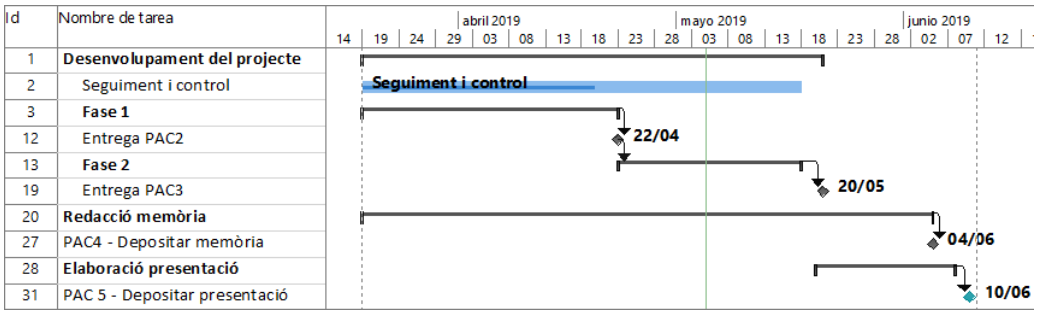


Figura 2.2: Diagrama a alt nivell de la planificació temporal del projecte.

A l'annex **??** es pot consultar el diagrama de Gantt complet amb totes les tasques i fites del projecte.

La Taula 2.4 conté totes les fites del projecte. Les que fan referència a una Prova d'avaluació contínua (PAC) corresponen a una entrega a la UOC. La resta fan referència a un lliurable del projecte.

Activitats del projecte			
Nom	Descripció	Que inclou	Resultats
Recerca bibliogràfica	Estudi de l'estat de l'art de tècniques de segmentació de mans.	Recerca de fonts bibliogràfiques relacionades amb la segmentació de mans. Identificació de la tècnica més adequada per el nostre cas d'ús. Recerca de les tècniques que usa la metodologia seleccionada.	Llistat de fonts bibliogràfiques.
Anàlisi i disseny del TFM.	Definició de com es durà a terme el treball.	Definició de les tecnologies a utilitzar. Selecció del model de desenvolupament. Definició detallada de l'abast.	Pla del projecte.
Desenvolupament	Iteracions del desenvolupament per a obtenir diferents versions del prototipus.	Execució de les 4 fases d'una iteració del model de desenvolupament evolutiu per a cada iteració que es dugui a terme.	Versió final del prototipus.

Taula 2.3: Taula amb les activitats a primer nivell del projecte.

2.2.4 Pla de gestió de riscos

Avaluació dels riscos

La taula 2.5 conté la definició dels riscos detectats i la seva avaluació.

Pla de contingència

Aquest pla conté com cal actuar davant els riscos identificats. L'estratègia principal que es seguirà serà de prevenció. En cas que tingui lloc algun risc es decidirà la resolució *ad-hoc*, això és possible gràcies al model evolutiu de desenvolupament, que potencia que la planificació sigui flexible.

Fites		
Codi	Descripció	Data
F00	PAC 2 – Desenvolupament – Fase 1	22/04/18
F01	Recerca bibliogràfica	07/04/19
F02	Anàlisi i disseny del TFM	22/03/19
F03	1ª iteració	29/03/19
F04	2ª iteració	12/04/19
F05	PAC 3 – Desenvolupament – Fase 2	20/05/19
F06	3ª iteració	01/05/19
F07	4ª iteració	10/05/19
F08	Publicació prototipus final	18/05/19
F09	PAC 4 – Memòria	04/06/19
F10	Redacció i correcció de la memòria	04/06/19
F11	PAC 5 – Presentació	12/06/19

Taula 2.4: Taula amb les fites del projecte.

Un exemple que mostra la particularitat d'aquest projecte és que en cas que tingui lloc el **R02** es modificaran els objectius del projecte, acció que seria impossible en la gran majoria de projectes de software. Per a prevenir aquest risc s'ha donat molta importància a la tasca de recerca bibliogràfica.

Per a la prevenció del **R03**, es planificaran les reunions de fi d'iteració amb una setmana d'antelació. Quan aquest risc tingui lloc s'aprofitarà el temps buit per a efectuar les tasques d'elaboració de la memòria.

Planificació

El monitoratge dels riscs es farà a la reunió setmanal entre l'autor del TFM i el seu tutor.

2.3 Desviacions respecte al pla inicial

En aquesta secció s'exposaran les desviacions principals que ha sofert el projecte durant les dues etapes principals en què s'ha dividit.

En la primera etapa es va desviar l'activitat de recerca bibliogràfica. Com que la part principal de la recerca bibliogràfica es va finalitzar a temps les fases posteriors varen començar abans d'acabar-la. L'execució de la implementació dels prototipus es va dur a terme des de les instal·lacions de la UIB, el temps que es tenia disponible fóra de les instal·lacions (principalment caps de setmana) es va continuar amb la recerca bibliogràfica. L'altra desviació va ser la finalització de la primera iteració en la meitat del temps, a causa d'una alta dedicació. Això va provocar que l'inici de la tercera iteració s'iniciés abans.

Durant la segona etapa, la tercera iteració es va allargar més del previst. Es va haver de desenvolupar una major quantitat de programari que el que s'havia pensat inicialment. Gràcies a les desviacions de la primera etapa, aquesta la iteració havia

Conjunts de dades						
Codi	Nom	Causa	Conseqüència	Impacte	Probabilitat	Nivell
R01	Iteracions més llargues del previst.	Mala previsió de la durada de les iteracions.	Aplaçament de les fites	Baix	Alta	Mitjà
R02	El mètode no es pot millorar.	El mètode és perfecte i no pot millorar-se.	El projecte acabarà abans de tenir un contingut el suficient complet com per a ser un TFM.	Alt	Mitjana	Alt
R03	Manca de disponibilitat dels membres de l'equip.	Manca de disponibilitat dels membres necessaris per a avaluar i començar la següent iteració.	Aplaçament de l'inici d'una iteració.	Mitjà	Alta	Alt

Taula 2.5: Taula amb els riscs projecte.

començat abans, pensant que així es podria afegir una altra iteració al desenvolupament. Així i tot, l'allargament de la iteració es va adaptar a la fita descrita inicialment. A causa d'això es va descartar la incorporació d'una quinta iteració.

Al final d'aquest document, a l'annex **??**, es pot trobar el cronograma del projecte actualitzat amb aquestes desviacions.

2.4 Arquitectura

El software que es desenvolupa utilitza com a entrada les dades d'un dataset separat en un conjunt d'entrenament i un de test. Per altra banda, com a entrada dels prototipus, s'utilitzen un conjunt d'imatges de profunditat obtingudes directament d'un sensor TOF. Sovint les imatges reals provindran del conjunt de test, però mai del d'entrenament.

Cada prototipus desenvolupat serà entrenat amb el conjunt d'entrenament i s'obtindran un conjunt de mètriques de la seva avaluació amb el conjunt de test. A partir de les imatges obtingudes directament del sensor TOF es generaran els resultats visuals

de la classificació, en forma d'imatges de probabilitat. Aquestes imatges són la sortida del RDF al classificar la imatge, on cada píxel representa la probabilitat que té aquest de pertànyer a una mà, en capítols posteriors sovint es farà ús d'aquestes imatges.

La Figura 2.3 conté l'esquema del funcionament del software. La primera passa sempre és entrenar el RDF amb el conjunt d'entrenament. Una vegada es té el classificador entrenat es classifica el conjunt de test i les imatges de probabilitat de la càmera TOF, amb el que s'obtenen els resultats del prototipus: les mètriques dels resultats i les imatges de probabilitat.

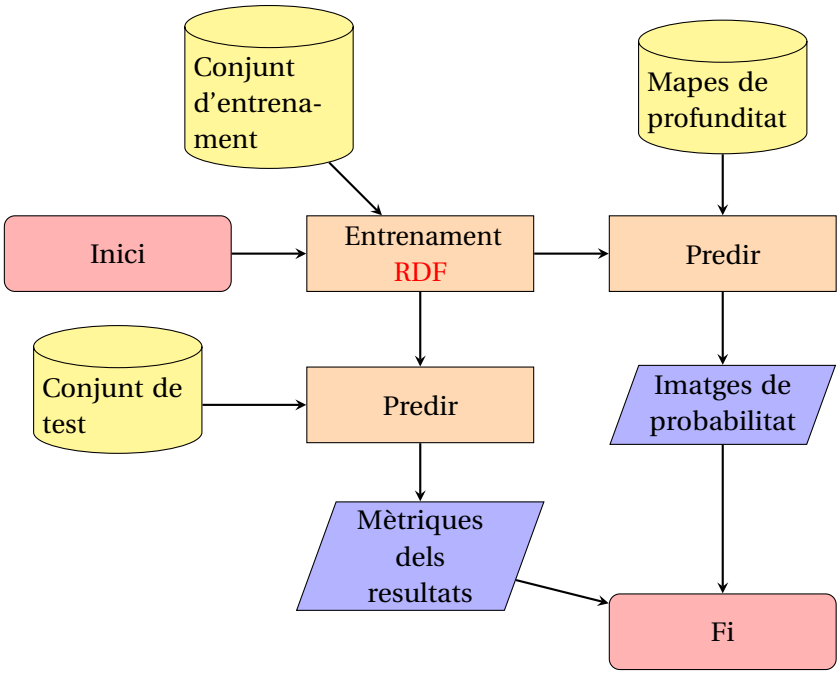


Figura 2.3: Esquema amb l'estructura del funcionament del software.

2.5 Tecnologies

L'execució del projecte requereix una sèrie d'elements tecnològics. L'obtenció de les imatges de profunditat depèn d'un dispositiu amb un sensor de profunditat. El desenvolupament de tota la funcionalitat associada requereix fer ús de diferents llenguatges de programació. Per evitar haver de desenvolupar parts genèriques dels prototipus també es farà ús de programari existent. Les eines seleccionades s'expliquen en els següents subapartats.

2.5.1 Maquinari

Pel desenvolupament del projecte es requereix bàsicament dos dispositius: un sensor de profunditat TOF i un ordinador.

El sensor TOF utilitzat és el de la *Kinect v2*, que és necessari per a obtenir els mapes de profunditat propis. Aquest dispositiu ofereix una API que permet accedir fàcilment a tots els seus sensors, i ofereix una gran quantitat d'utilitats per a processar imatges de

profunditat. S'ha seleccionat aquest dispositiu pel fet que estava disponible al laboratori on s'ha elaborat la major part del projecte.

Per a l'entrenament del RDF i obtenir els resultats de cada prototipus s'ha utilitzat un ordinador de sobretaula sense característiques tècniques rellevants.

2.5.2 Programari

Quant al programari necessari per a dur a terme el projecte hi ha dos components principals. El llenguatge de programació i les implementacions disponibles a la xarxa que poden ser útils, siguin en forma de llibreria o d'executables.

Llenguatge de programació

L'elecció del llenguatge de programació és un punt clau per a l'eficiència del desenvolupament i l'acompliment dels requisits del projecte. S'han d'analitzar els punts forts i els febles de cada llenguatge respecte al problema que s'està enfrontant.

El projecte bàsicament engloba dues disciplines: el processament d'imatges i l'aprenentatge automàtic. Existeixen una gran quantitat de llibreries en molts de llenguatges de programació per utilitzar com a *framework* d'aquestes disciplines. Una de les llibreries més populars de processament d'imatges és `OpenCV`, que disposa d'implementacions en `Python` i `C++`. Per a l'aprenentatge automàtic es poden trobar implementacions de caràcter públic en qualsevol llenguatge de programació a plataformes com `GitHub`, ara bé, `Python` disposa de la llibreria `scikit-learn` que implementa una gran varietat d'algorismes d'aprenentatge automàtic.

Per a la selecció del llenguatge també s'ha de tenir en compte el model de desenvolupament, hi ha llenguatges que s'adapten millor que altres a cada model de desenvolupament. En aquest cas es requereix un llenguatge de programació que permeti un desenvolupament àgil pel tractament d'imatges i dades. Els candidats més populars amb aquestes característiques són `Python` i `Matlab` o alternatives similars com `Octave`.

Finalment, hem de considerar la comunicació amb els sensors de profunditat. La *Kinect v2* disposa d'una API per accedir fàcilment a la informació dels seus sensors. Proporciona exemples i una documentació extensa. El problema és que només està implementada en `C++` i `C#`, cap dels dos llenguatges és adequat per a seguir un model de desenvolupament basat en prototipus.

D'acord amb la discussió dels paràgrafs anteriors, i del fort pes que té que el llenguatge sigui adequat per a un desenvolupament evolutiu, s'ha seleccionat `Python` en tot el que sigui possible. El punt d'inflexió d'aquesta elecció és l'experiència prèvia, i l'ampli ventall de llibreries de processament d'imatges i aprenentatge automàtic que disposa.

La interacció amb la *Kinect v2* es durà a terme amb `C++` que permet treballar amb la llibreria `OpenCV`. Aquest llenguatge, a més, proporciona una alta eficiència computacional. Té el defecte que la programació és més complexa: és tipat, s'ha de compilar, té una sintaxi més complexa i altres característiques que fan que sigui incompatible amb un model de desenvolupament basat en prototipus.

Així que, utilitzarem dos llenguatges de programació `Python` pel desenvolupament dels prototipus i `C++` per a la interacció amb la *Kinect v2*. Per aquest motiu, el resultat

del projecte estarà format per dos productes de software: el prototipus i el programa per interactuar amb la *Kinect v2*.

La solució amb `Python` únicament és un prototipus, el llenguatge és ineficient per a la gran quantitat de còmput de la implementació i els seus requisits d'eficiència en producció. Per a una posada en producció dels resultats del projecte el més òptim seria fer tota la implementació amb `C++`.

Llibreries

El desenvolupament dels prototipus s'ha recolzat en una sèrie de llibreries externes, que proporcionen un nivell d'abstracció al desenvolupament. Aquestes permeten utilitzar un software més eficient i de més qualitat que el que es podria arribar a desenvolupar amb el temps del projecte.

- *OpenCV*: Conté tota la funcionalitat necessària de processament d'imatges, i un gran ventall d'implementacions d'algorismes de visió per computador.
- *Numpy*: Llibreria d'abstracció de les operacions matricials. Implementa totes les operacions sobre matrius amb el llenguatge `C`. Les implementacions natives de `Python` d'aquestes estructures de dades són molt ineficients.
- *Matplotlib*: Llibreria de generació de gràfiques 2D. Utilitzada per visualitzar gràficament els resultats dels experiments.
- *scikit-learn*: Conté la implementació d'un gran nombre d'algorismes d'aprenentatge automàtic. També disposa d'utilitats relacionades com l'obtenció de mètriques dels resultats.
- *pandas*: Llibreria per a l'anàlisi i visualització de les dades. Permet treballar amb conjunts de dades d'una manera amigable, amb la contrapartida d'un major ús de recursos.
- *ovnImage*: Llibreria pròpia disponible públicament. Proporciona utilitats usades comunament al codi, mètodes de suport de processament d'imatges, visualització dels resultats, anàlisi de dades i *machine learning*.

Gestió del desenvolupament

A part de les eines anteriors, s'han utilitzat altres tecnologies per a gestionar els canvis en el progrés del desenvolupament. Aquestes eines permeten gestionar de manera robusta les distintes versions del software i faciliten la seva publicació.

- *Git*: Sistema de control de versions. En aquest projecte únicament s'utilitza per a gestionar les distintes versions per les quals va passant el software. Tot i que també és una eina molt potent per a la col·laboració entre distints desenvolupadors. S'han publicat els productes resultants al servei de *hosting* de repositoris `Git` `GitHub`.

- *Conda*: Software de gestió de les dependències del programari. Serveix per a gestionar les dependències de llibreries de cada versió del software. Permet construir un entorn virtual amb les versions de les llibreries que necessita una versió del software concreta.

En aquest capítol s'ha definit com s'ha duit a terme el projecte. S'ha començat en la secció 2.1 explicant el model de desenvolupament utilitzat. A la secció 2.2 s'ha detallat el pla de projecte i a la secció 2.3 les seves desviacions. Per tancar el capítol en la secció 2.4 s'ha definit l'arquitectura del software i a la secció 2.5 la tecnologia utilitzada.

CAPÍTOL 3

MÈTODE

En aquest capítol es defineix el mètode per a segmentar mans en temps real utilitzant imatges de profunditat. La tècnica es sosté sobre un algorisme d'aprenentatge automàtic que proporciona uns resultats precisos en un temps reduït, un RDF [24]. Aquest classificador, combinat amb unes característiques simples que consisteixen en la comparació de dos píxels, permet realitzar la segmentació en els quadres d'un vídeo en temps real.

El mètode parteix d'una imatge de profunditat i acaba amb una imatge amb les probabilitats de cada píxel de pertànyer a una mà. El procés s'organitza en tres fases principals. En la primera passa es seleccionen els píxels a classificar amb un algorisme de segmentació del fons, comunament anomenat en anglès *background substraction* [16]. En la segona passa, es calcula un conjunt de característiques per a cada píxel a classificar. Finalment, usant un RDF s'obtenen les probabilitats de tots aquests píxels de pertànyer a una mà.

Les seccions d'aquest capítol exposen els elements més importants de la metodologia. En la secció 3.1 es defineix el procés seguit per a obtenir la classificació dels píxels d'una imatge. En la secció 3.2 s'especifica el format de les imatges que utilitza el mètode. La secció, la 3.3, especifica el format de les característiques que utilitza la tècnica. La secció 3.4 detalla el funcionament del RDF. I per acabar el capítol, la secció 3.5, conté els detalls de la implementació del mètode.

3.1 Procés

Aquesta secció conté els detalls del procés de funcionament del mètode. Es defineix el format de les imatges d'entrada, com són processades i com finalment s'obté la classificació de tots els píxels de la imatge.

Les imatges d'entrada del mètode provenen d'un sensor de profunditat de tipus TOF [25], a partir de les quals, s'obtindrà una imatge on cada píxel indica la seva distància del sensor.

El diagrama de flux de la Figura 3.1 il·lustra el procés per a obtenir la classificació dels píxels de la imatge. El procés parteix d'una imatge de profunditat, a la qual s'aplica un algorisme de *Background Substraction* per a seleccionar els píxels a classificar. Després es calcula un conjunt de característiques per cada un d'aquests píxels perquè puguin ser usats com a exemples d'entrada d'un classificador. Per acabar, s'utilitza un RDF per a obtenir les probabilitats dels exemples de ser mà.

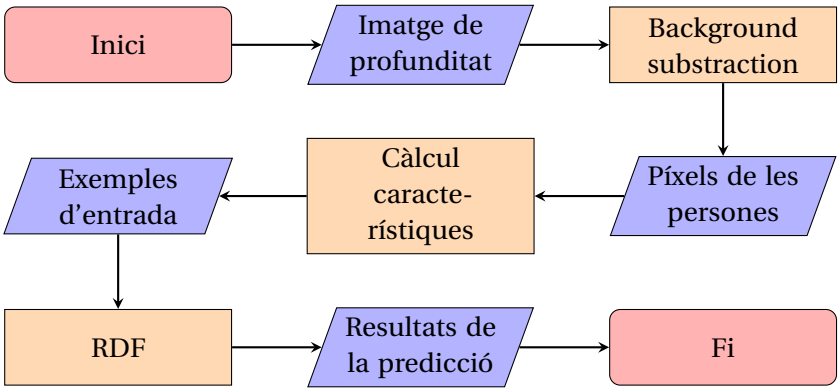


Figura 3.1: Esquema del procés usat per a obtenir la probabilitats dels píxels de pertànyer a una mà.

3.2 Imatges d'entrada

El sistema per a funcionar requereix d'una font de dades, imatges de profunditat. Totes les dades utilitzades provenen de datasets públics o han estat capturades amb un sensor TOF.

S'ha utilitzat el sensor TOF del dispositiu *Kinect v2 de Microsoft*. El qual també proporciona una càmera *Red, Green, Blue* (RGB) i una càmera d'infrarojos, gràcies a la càmera de color podem transformar les imatges de profunditat en imatges RGBD. El projecte ha ignorat l'existència de la càmera d'infrarojos.

Les úniques dades que s'han capturat han estat el color i la profunditat. Com que el sensor de profunditat i el de color no són el mateix, estan ubicats a llocs distints i proporcionen una perspectiva diferent de les imatges. Per això, s'ha de mapejar la informació d'una de les dues matrius a sobre de l'altre. Com que hi ha posicions que no es poden mapejar, píxels no visibles des de l'altra perspectiva, la imatge traslladada tendra valors buits. Per això s'ha mapejat el color a sobre de les matrius amb la profunditat, així es pot treballar amb les imatges de profunditat originals. La Figura 3.2 visualitza la imatge de color traslladada a les coordenades de la profunditat. En la secció 4.2 es defineix com s'aconsegueix la translació. En les imatges de profunditat cada píxel indica la distància en que es troba de la càmera.

3.3 Característiques

Les característiques per a classificar els píxels són les definides per Shotton *et al.* [19], que es varen inspirar en les de Lepetit *et al.* a [26]. Cada característica consisteix en

Figura 3.2: Imatge de color traslladada a les coordenades de la imatge de profunditat. Els píxels no pintats són els que no s'ha pogut obtenir el seu valor.

la diferència de la profunditat de dues posicions. Vénen definides per dos *offsets*, els quals són dos vectors que defineixen la distància i la direcció dels dos píxels a comparar respecte al píxel objectiu.

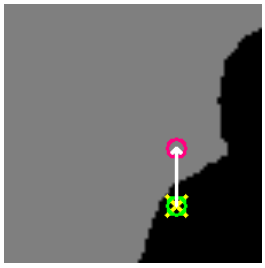
L'equació 3.1 conté el còmput de la característica θ del pixel x, on $d_I(x)$ és el valor de la profunditat del píxel x en la imatge I, i u i v són els *offsets* de la característica θ. Gràcies a la divisió dels *offsets* entre la profunditat del píxel ($\frac{1}{d_I(x)}$) es normalitza la posició dels *offsets* de manera que són invariants a la distància entre l'objecte i la càmera. És a dir, un *offset* defineix una distància real al píxel, comprova el valor del píxel que està a una distància en metres del píxel, no a un determinat nombre de píxels.

$$f_\theta(I,x) = d_I\left(x + \frac{u}{d_I(x)}\right) - d_I\left(x + \frac{v}{d_I(x)}\right) \tag{3.1}$$

Així que, $d_I\left(x + \frac{u}{d_I(x)}\right)$ fa referència al valor de la profunditat d'un píxel relatiu a x definit pel *offset* u. L'altra part de l'equació $d_I\left(x + \frac{v}{d_I(x)}\right)$ referència al valor de la profunditat en un altre píxel relatiu a x, aquest cop definit per v.

La figura 3.3 il·lustra el funcionament de les característiques. Mostra que depenent de l'estructura dels *offsets*, aquestes prenen diferents valors segons la seva ubicació o la forma de l'objecte al qual pertanyen. Les imatges 3.3a, 3.3b i 3.3c mostren uns tipus de característiques que donen informació sobre la posició del píxel a sobre de l'objecte. La que apareix a la Figura 3.3a pren valors elevats quan el píxel pertany a la part superior d'un objecte. Les altres dues 3.3d i 3.3e proporcionen informació sobre la forma de l'objecte al qual pertanyen, en lloc de sobre la posició del píxel a sobre de l'objecte.

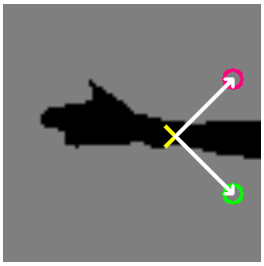
La motivació de l'ús d'aquestes característiques, com està definit a [19], és que encara que individualment proporcionen poca informació, usades conjuntament i combinades amb un RDF, són el suficient precises per discriminar totes les parts entrenades. Una altra justificació és la seva eficiència computacional: no necessiten



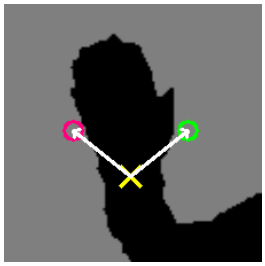
(a) Característiques que prenen valors alts en píxels superiors de l'objecte.

(b) Característiques que prenen valors alts en píxels de la part esquerra de l'objecte.

(c) Característiques que prenen valors alts en píxels de la part dreta de l'objecte.



(d) Característiques que ajuden a trobar estructures horitzontals.



(e) Característiques que ajuden a trobar estructures verticals.

Figura 3.3: Exemples d'estructura de característiques que ajuden a discriminar una part concreta del cos. La creu groga indica el píxel objectiu, x. El cercle groc fa referència a la posició a comparar definida pel *offset* u, $d_I\left(x+\frac{u}{d_I(x)}\right)$; i el cercle violeta a la posició definida pel *offset* v, $d_I\left(x+\frac{v}{d_I(x)}\right)$. Les figures mostren un tros d'una imatge de profunditat amb una persona segmentada del fons, l'ombra negre és la persona.

preprocessament, cada característica únicament necessita llegir com a molt 3 píxels i efectuar 5 operacions aritmètiques. A més, els càlculs de les característiques es pot implementar fàcilment a sobre d'una GPU. Teòricament, característiques més potents basades en, per exemple, integrals profundes sobre regions, curvatura o descriptors locals e.g. [27] oferirien millors resultats, però impossibilitarien la seva execució en temps real [19].

3.4 Randomized Decision Forests

Un RDF és un conjunt de T arbres de decisió, formats per nodes de decisió i nodes fulla [24]. La Figura 3.4 mostra gràficament la seva estructura interna, els nodes blaus són els de decisió i els verds les fulles. Els nodes de decisió estan formats per una característica f_θ i un llindar τ. Quan es classifica un píxel viatge en cada arbre des del seu node arrel a un node fulla. A cada node de decisió es bifurca cap al seu fill dret o esquerre segons el resultat de la comparació del valor de la seva característica f_θ amb el seu valor de τ. Els nodes fulla tenen una probabilitat associada per cada classe del conjunt d'entrenament, que són les probabilitats que té un exemple que ha acabat en aquest node de pertànyer a cada classe. Els resultats de la classificació en un arbre de decisió són les probabilitats

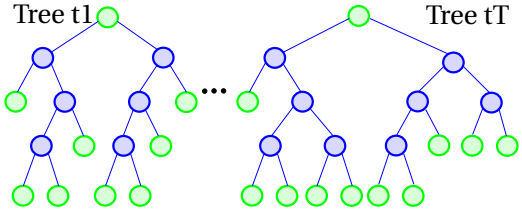


Figura 3.4: Randomized Decision Forest, RDF. Estructura interna.

que hi ha associades al node fulla on ha acabat l'exemple. En el cas del RDF es disposa de múltiples arbres de decisió, el resultat final de la classificació és la distribució de probabilitat de cada classe amb els resultats de cada arbre (P_t), eq. 3.2.

$$P(I,x) = \frac{1}{T}\sum_{t=1}^{T} P_t(I,x) \tag{3.2}$$

El RDF usat en els experiments d'aquest treball usa el criteri d'"entropia" per a mesurar la qualitat de les separacions de les dades en cada node. Durant l'entrenament, l'objectiu és identificar el valor dels paràmetres f_θ i τ que millor separin el conjunt d'entrenament. Això s'aconsegueix aplicant un algorisme d'optimització sobre l'entropia [28].

Una propietat del RDF és que permet entrenar un conjunt d'arbres de manera independent i incorporar-los al RDF. A causa de la gran quantitat de dades disponibles s'ha aprofitat aquesta propietat per entrenar el classificador per blocs. D'aquesta manera s'ha separat el conjunt d'entrenament en una sèrie de subconjunts, els quals han entrenat un nombre concret d'arbres independents dels altres, aquest procés es detalla en la secció 3.5.2.

Durant d'execució de diferents experiments que s'exposen en aquest treball s'analitzen les propietats de l'algorisme per extreure conclusions sobre el comportament de les característiques. Una propietat interessant del RDF és que disposa d'alguns paràmetres que permeten configurar-lo. Un dels més importants és la profunditat màxima que es permet que tinguin els arbres de decisió. Amb aquest paràmetre es pot jugar amb la mida dels arbres que formen el bosc, el que permet evitar el *over-fitting*, i descartar parts dels arbres que no ajuden a obtenir bons resultats. El *over-fitting*, o sobre-entrenament, és l'efecte que es produeix quan l'algorisme en lloc d'aprendre el comportament de les dades s'aprèn el conjunt d'entrenament, de manera que és incapaç de generalitzar a altres dades. Així quan s'usa amb dades que divergeixen una mica és incapaç de classificar-les correctament. La figura 3.5 conté un exemple il·lustratiu d'aquest efecte.

3.5 Implementació

En les seccions anteriors s'han definit els fonaments teòrics sobre els quals es basa la metodologia. Per a la implementació de la tècnica hi ha molts d'aspectes que s'han de considerar. Els requisits i restriccions del projecte exigeixen una sèrie de criteris que ha de complir el treball que s'assoleixen amb una implementació concreta.

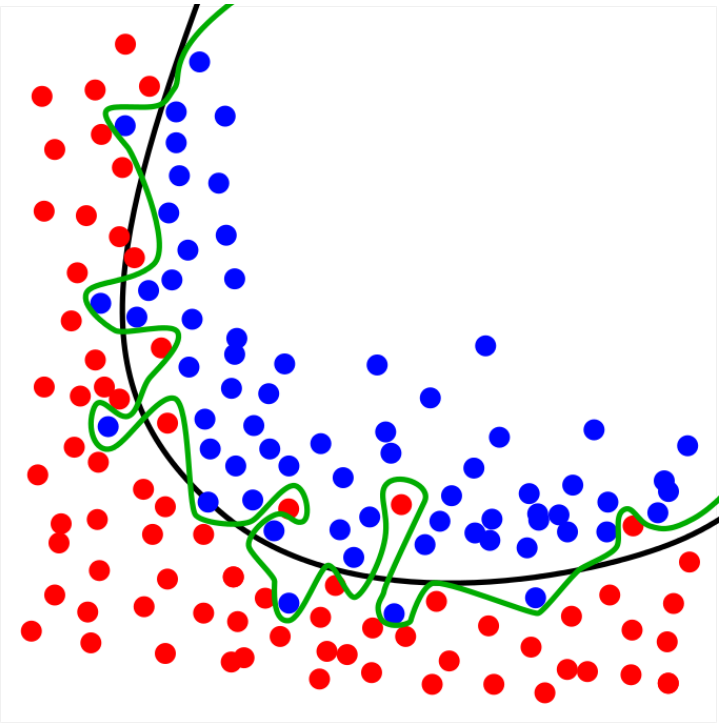


Figura 3.5: Exemple d'*over-fitting*, els punts simulen els exemples d'entrenament on el seu color simbolitza la seva classe. La línia verda simbolitza la separació de les dades amb *over-fitting* i la negra la separació en que es generalitza el comportament de les dades. Imatge proporcionada per Chabacano – Treball propi, CC BY-SA 4.0, Enllaç.

3.5.1 Característiques

Un detall de la implementació de les característiques és com s'obté la seva estructura, els *offsets* que les componen. Aquests són definits aleatòriament a partir d'una distribució uniforme, que pren valors en un rang definit experimentalment, del qual es parlarà en la secció 6.1.2. A la meitat de les característiques es posa un dels dos *offsets* a zero, a sobre del píxel objectiu, com les que es mostren a les figures: 3.3a, 3.3b i 3.3c, que són el tipus de característiques que més ajuden a trobar la mà [3].

La implementació que es considera utilitza 1000 característiques per cada píxel. Amb aquesta gran quantitat de característiques s'assegura una alta exactitud, amb la contrapartida que s'incrementa el temps de computació i l'ús de memòria. Com es discutirà en el capítol 6 es pot reduir el nombre de característiques fins a unes 100 mantenint una exactitud acceptable, en el cas que necessités implementar l'algorisme a una màquina amb menys recursos seria el que s'hauria de fer.

Un requisit del mètode és que ha de poder ser executat en temps real. Per això tant el còmput de les característiques com la predicció s'han d'executar de forma eficient.

Es podria millorar la paral·lelització efectuant el càlcul de cada característica en cada node de decisió dels arbres. Això surt de l'abast d'aquest treball, ja que suposaria implementar l'algorisme del RDF des de zero.

3.5.2 Entrenament

Una de les propietats més problemàtiques del mètode és l'espai que ocupen les característiques computades en memòria i la quantitat de dades que s'usen. Així, utilitzant les característiques com a nombres en coma flotant de 64 bits, i 1000 característiques per cada píxel, cada exemple d'entrenament ocuparà el següent espai a memòria: $64 \cdot 1000 = 64000$ bits = 8 KB.

Ja es pot intuir que no és possible entrenar el RDF amb tots els píxels disponibles de totes les imatges. Per això es selecciona aleatòriament un nombre fixe de píxels de cada classe en cada imatge, aquest nombre s'identifica com $nSamples$. El nombre de bits que es necessiten per cada imatge d'entrenament es calcularà com: $bitsFeat \cdot nSamples \cdot nFeats$. On $bitsFeat$ és el nombre de bits que ocupa cada característica, i $nFeats$ és el nombre de característiques utilitzat.

Així, per $nSamples = 2000$, que no arriba a un 1% dels píxels de la imatge, ja que totes les imatges de la base de dades usada tenen una resolució de $424x512$, 217088 píxels. L'espai que ocuparà la seva informació serà de: $8 \frac{\text{KB}}{\text{píxel}} \cdot 2000$ píxels = 16000 KB = 16 MB.

Per a mitigar aquest problema el que es fa és reduir la precisió de les característiques eliminant les seves dues xifres amb menys pes, dividir-les per 100 i convertir-les a enters de 8 bits. Amb aquesta nova configuració l'espai que ocupen els exemples d'entrenament de cada imatge és: 1 byte $\cdot$ 1000 característiques $\cdot$ 2000 píxels = 2000000 bytes = 2 MB.

Tot i que s'ha reduït dràsticament la quantitat de memòria utilitzada pel conjunt d'entrenament, és impossible emmagatzemar totes les dades del conjunt d'entrenament a memòria per a entrenar el classificador. Com s'ha introduït a la secció anterior el RDF permet entrenar subconjunts d'arbres de decisió de manera independent, per a posteriorment incorporar-los al model complet. S'ha aprofitat aquesta propietat dividint el dataset d'imatges D en n subconjunts D_i, on $\bigcup_{i=1}^{n} D_i = D$ i $\bigcap_{i=1}^{n} D_i = \emptyset$. Així, entrenant un nombre reduït d'arbres amb cada D_i, per a després integrar-los dins el RDF, es pot entrenar l'algorisme complet utilitzant un gran nombre d'imatges. A més, aquesta estratègia ajuda a accelerar el temps d'entrenament i redueix el *over-fitting* [29]. La mida de cada subconjunt es decideix d'acord al nombre d'exemples d'entrenament que es prenen de cada imatge i del nombre d'imatges que formen cada subconjunt.

3.6 Conjunts de dades

L'altre element clau de la metodologia són els conjunts de dades utilitzats per a entrenar i testejar el classificador. Aquests són els que defineixen com es comportarà el model.

Aquests conjunts de dades contenen dos tipus d'informació: les imatges de profunditat i les etiquetes dels píxels de les mans.

S'han utilitzat dos datasets, l'utilitzat en el treball de Tompson *et al.* a [3], que es referencia com el dataset de la NYU. I un dataset propi, elaborat amb imatges preses en un entorn no controlat, el DBG. Les següents seccions detallen els dos datasets.

3.6.1 Dataset NYU

El primer dataset usat és el de la NYU, aquest dataset ha estat utilitzat en el treball previ de Tompson *et al.* a [3], el qual és el punt de partida d'aquest projecte.

Els seus punts forts són que té un nombre d'imatges adequat i que proveeix un etiquetatge de les mans de qualitat. Està centrat en la segmentació de mans per a l'anàlisi de gestos, per això, totes les imatges tenen les mans en primer pla, apareixen a davant del cos dels actors. El dataset està compost per 6716 imatges amb una única persona en primer pla fent gestos amb les mans, i a més té el fons segmentat.

La figura 3.6 il·lustra uns quants exemples de les imatges que disposa el dataset. Les imatges contenen la informació dels 16 bits de profunditat separats en el canal blau i verd de la imatge, i al canal vermell conté les etiquetes de les mans. Es pot veure que totes les imatges tenen el fons de la imatge segmentat amb un valor concret, amb el qual és fàcil identificar els píxels que pertanyen a les persones que apareixen en les escenes.

Aquest dataset està enfocat en la detecció de les mans per a identificar la seva postura. Per aquest motiu totes les imatges contenen les mans en el primer pla de l'escena, el que provoca que el dataset disposi molt poca variabilitat entre les imatges.

Figura 3.6: Imatges d'exemple del dataset de la NYU.

3.6.2 Dataset propi

Com es veurà a capítols posteriors, els resultats amb el dataset de la NYU fallen en moltes situacions, no contempla molts de casos que tenen lloc en un ambient no controlat. Amb el fi de solucionar aquests problemes es va crear aquest nou dataset, el DBG. Aquest permet analitzar la tècnica en un entorn real, incloent-hi la major quantitat de variabilitat possible entre les imatges. El dataset inclou una persona efectuant activitats davant la càmera, tant asseguda com dreta, intentant incorporar la major quantitat possible de diferents posicions de les mans i dels braços. Les motivacions per a la creació d'aquest dataset es detallen en la secció 4.3.

El nou dataset està format per 6331 imatges preses en un ambient no controlat. Incorpora un ampli rang de noves escenes que no apareixien en el NYU, amb el propòsit d'intentar generalitzar el màxim els resultats obtinguts. Aquestes noves escenes són:

- Una persona caminant amb les mans en una posició natural.
- Una persona dreta amb les mans i els braços en distintes posicions. S'intenta cobrir el major nombre de casos possibles.
- Una persona asseguda amb les mans i els braços en distintes posicions. S'intenta cobrir el major nombre de casos possibles.

- Una persona caminant interactuant amb objectes.
- Una persona dreta interactuant amb objectes.
- Una persona asseguda interactuant amb objectes.

El contingut està format per parts més o menys iguales amb les escenes d'una persona dreta, d'una persona asseguda i d'una persona caminant. Una gran aportació del dataset que no contemplava el de la NYU, són les imatges que incorporen interacció amb objectes.

En la figura 3.7 es mostren una sèrie d'exemples de les imatges que conté el dataset. En aquestes imatges hi ha combinades la informació de la profunditat, de la segmentació del fons i les etiquetes de les mans.

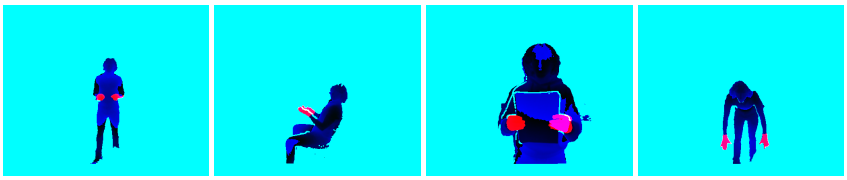


Figura 3.7: Imatges d'exemple del DBG.

A part, el dataset disposa de la informació del color mapejada a sobre de la imatge de profunditat, tal com es mostra a la figura 3.8.

Figura 3.8: Informació del color del DBG.

En aquest capítol s'han definit tots els fonaments teòrics sobre els quals es basa el mètode. S'ha començat detallant el procés necessari per a obtenir els resultats de la predicció d'una imatge i el format que han de tenir aquestes per a ser processades correctament. Després s'ha especificat en què consisteixen les característiques necessàries per a fer la predicció i s'han definit els detalls del classificador utilitzat, el RDF. S'ha continuat explicant els detalls tècnics rellevants de la implementació, i s'ha tancat el capítol exposant els detalls dels conjunts de dades utilitzats. La informació presentada en aquest capítol són tots els detalls del mètode.

CAPÍTOL 4

EXPERIMENTS

Per adquirir un major coneixement del mètode i poder millorar la tècnica s'han dut a terme un seguit d'experiments. Els resultats no sempre han estat satisfactoris, en alguns casos no s'han aconseguit els objectius, així i tot, tots ells han estat fructífers pels resultats del projecte. Els fracassos han incrementat el coneixement de les limitacions del mètode, i els èxits han permès millorar la tècnica.

En aquest capítol s'exposen les diferents activitats que s'han agut de dur a terme per a desenvolupar el projecte. En la secció 4.1 es discuteix la definició de la implementació del mètode, on es detallen els seus paràmetres. La resta de seccions aniran presentant com s'han dut a terme els experiments efectuats per millorar el mètode. Cada secció s'estructura de la següent manera: primer s'estableixen els problemes que es volen solucionar, en segon lloc les tècniques que s'apliquen al mètode i per acabar es discuteix en quina mesura s'han assolit els objectius.

4.1 Ajustament de paràmetres

Una vegada implementada la primera versió del prototipus, el primer problema és identificar una configuració adequada pel mètode. Es disposa d'una gran quantitat de paràmetres als quals s'ha d'assignar un valor. Per exemple, la implementació del RDF de la llibreria *scikit-learn* disposa de 17 paràmetres, la majoria dels quals no es mencionaran aquí al ser irrellevants per l'anàlisi (*ie.*: el nombre de processos a executar en paral·lel durant l'entrenament). D'altres estan clarament definits pels autors originals del mètode, com el criteri per a decidir la millor separació dels exemples durant l'entrenament del RDF: la "entropia".

4.1.1 Llistat dels paràmetres

Els paràmetres que componen el mètode es poden dividir en dos grups principals segons l'àmbit on afecten. Per això, s'ha separat la seva definició en dues llistes. La primera presenta els que afecten al funcionament global de l'algorisme: format de les

dades i de les característiques. La segona llista conté els que afecten l'entrenament i l'estructura interna del RDF. Els paràmetres del funcionament global són:

- La distància màxima dels *offsets* de les característiques respecte al píxel objectiu.
- El nombre de característiques.
- El nombre d'imatges que conté cada subconjunt d'entrenament.
- El nombre d'exemples de cada classe que es prenen de cada imatge per entrenar el RDF.

A més d'aquests paràmetres el mètode en disposa d'uns altres que afecten a com es construirà el RDF i quina estructura tindrà. Dels 17 paràmetres que té la implementació utilitzada només se'n tractaran tres. La resta, o s'han considerats irrellevants per a la problemàtica, o el seu valor no afecta als resultats obtinguts. Els utilitzats són:

- La profunditat màxima dels arbres de decisió.
- El nombre mínim d'exemples que pot tenir un node per a ser una fulla.
- El nombre d'arbres de decisió a entrenar per a cada subconjunt del conjunt d'entrenament.

No es pretén identificar el valor òptim dels paràmetres. En l'etapa inicial no tindria sentit, ja que a mesura que evoluciona el prototipus els valors òptims variaran. Amb aquest anàlisi inicial es pretén identificar els valors que han de rondar els paràmetres.

4.2 Captura d'imatges

En aquesta secció es tractarà la problemàtica associada amb la captura d'imatges directament amb un sensor TOF. Per a capturar-les i processar-les adequadament s'ha definit un procediment que s'encarrega de capturar les imatges del dispositiu i retornar-les amb el format necessari pel mètode.

4.2.1 Sortida de la *Kinect v2*

Del dispositiu utilitzat, la *Kinect v2*, es recopilen les dades de dos sensors: el sensor TOF i un càmera convencional. En la secció 3.2 es definia la problemàtica de tenir sincronitzada la informació del color i de la profunditat de cada *frame*. Per a sincronitzar les informacions l'API de la *Kinect v2* proporciona un mètode per a traslladar cada píxel d'una imatge a les coordenades de l'altre. A causa de la problemàtica dels valors nuls, es traslladen els píxels de la imatge de color a la seva posició en la imatge de profunditat, amb el que permet treballar amb tota la informació de la profunditat, que és la que usa el RDF.

La Figura 4.1 mostra l'estructura de la sortida de la *Kinect v2* de forma esquemàtica. A partir de les imatges de profunditat ("depthmap") i les de color ("Color image") la API del dispositiu calcula el núvol de punts de la imatge de color a sobre de la de profunditat. El resultat serà la imatge de color projectada a damunt de la imatge de profunditat ("Colorized Point Cloud"), en la qual es pot apreciar tots els píxels que no s'han pogut

projectar en color blanc. Un exemple més clar d'aquest efecte es pot apreciar a la figura 3.2.

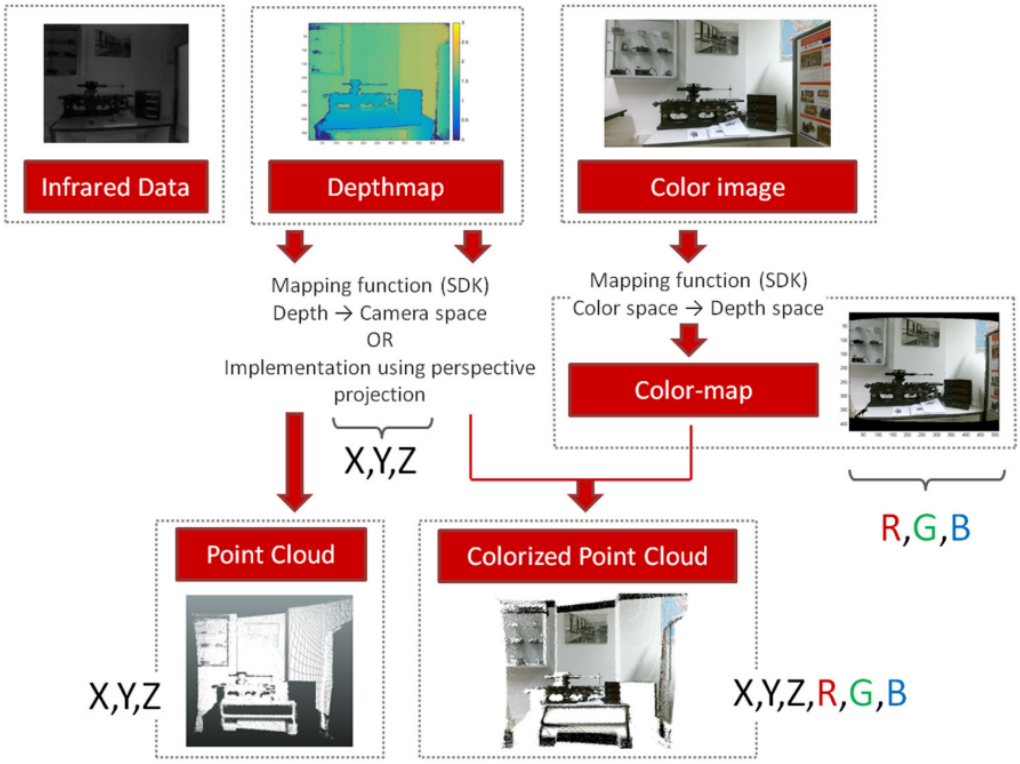


Figura 4.1: Representació esquemàtica de la sortida de la *Kinect v2* i del núvol de punts oferida a [2].

4.2.2 Processament de les imatges

Les imatges capturades d'aquesta manera encara no són compatibles amb les del dataset. Es necessita separar el fons de la imatge de les persones que apareixen en l'escena, això s'aconsegueix amb un algorisme de *background substraction*. En una primera aproximació es va aplicar una implementació d'un projecte anterior realitzat pel mateix grup de recerca [16]. Degut a la necessitat d'automatitzar al màxim el procés finalment es va decantar per a realitzar la segmentació del fons mitjançant les utilitats de l'API del dispositiu. La informació de l'àmpliament conegut *skeleton tracking* de la *Kinect v2* es pot consultar amb la seva API. La figura 4.2 mostra la sortida del *skeleton tracking*, a més de la informació de les articulacions indica quins píxels pertanyen a les persones, aquesta informació és utilitzada per a segmentar-les de la resta de la imatge.

Les imatges processades amb el procés descrit són compatibles amb les del dataset usat pel sistema inicial, el de la NYU. Això permet tractar de la mateixa manera les imatges capturades directament amb el dispositiu i les del dataset.

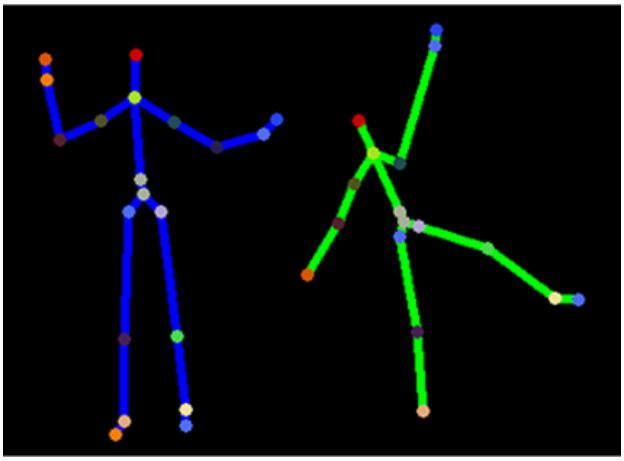


Figura 4.2: Exemple de la sortida del *skeleton tracking* de la *Kinect v2*.

4.3 Dataset propi

L'execució de l'algorisme amb imatges del món real no funcionava com es tenia previst, com es veurà a la secció 5.1.1, en la majoria de postures naturals de les persones l'algorisme no funcionava. Per exemple, l'algorisme no estava pensat per a detectar les mans en repòs d'una persona que creua l'escena caminant. Al dataset de la NYU totes les imatges eren d'una persona en primer pla realitzant gestos. Amb la motivació d'aconseguir un algorisme capaç de funcionar en la resta de casos, és a dir per a generalitzar al màxim els resultats del mètode, es va crear el DBG, detallat en la secció 3.6. Gràcies al procés definit en la secció anterior les imatges del nou conjunt són compatibles amb les imatges utilitzades prèviament. Amb aquest nou conjunt de dades es va construir un nou detector de mans amb el fi generalitzar els resultats per a qualsevol escena.

L'objectiu de la creació del nou dataset és obtenir un RDF més generalista del que s'aconseguia amb el dataset de la NYU. La implementació amb aquell primer conjunt de dades havia demostrat que en les imatges reals no detectava les mans en tots els casos. Per exemple, les mans de persones que apareixen caminant per l'escena no són detectades. En el capítol 5 es poden consultar els resultats obtinguts amb la primera versió, i els seus problemes, també s'hi trobarà com varien els resultats utilitzant el DBG en el procés d'entrenament.

4.3.1 Obtenció de les etiquetes

A la secció 4.2 s'ha definit com es capturen les dades des del dispositiu. La informació de les imatges no és suficient per entrenar i testejar el RDF, fa falta obtenir les etiquetes de les mans, per aconseguir-ho els actors apareixen usant uns guants roigs en totes les imatges. Posteriorment amb un procés semiautomàtic es segmenten les mans que apareixen en l'escena usant la seva colorimetria. Aquí és on entra en joc la utilitat de disposar de les imatges de color mapejades a sobre les de profunditat.

A la figura 4.3 es mostra la informació del color utilitzada. La figura 4.3a és la imatge capturada per la *Kinect v2*. En la figura 4.3b hi ha la imatge que s'ha utilitzat per obtenir les etiquetes de les mans. Aquesta és la imatge de color mapejada a sobre de la de

profunditat, com la que apareix a la figura 3.2, on a més se li han aplicat el resultat del *background substraction* per a descartar els possibles components del fons amb colors similars als guants.

(a) Imatge de color capturada amb el dispositiu.

(b) Imatge de color transformada a les coordenades de la profunditat i filtrada amb el *backgroud substraction*.

Figura 4.3: Transformació de la imatge de color capturada a les coordenades de la profunditat.

Per a aconseguir les etiquetes s'ha utilitzat l'espai de color *Hue, Saturation and Value* (HSV). Aquesta codificació del color facilita la detecció dels píxels vermells, els que els seus canals prenen valors entre $H \in [170, 190]$, $S \in [150, 255]$ i $V \in [50, 255]$ són els que comunament pertanyen als guants. Tot i així, aquests valors varien segons la il·luminació o les ombres de l'escena. Per obtenir uns resultats amb una qualitat acceptable, l'etiquetatge es du a terme de forma semiautomàtica, es permet variar el rang de valors de HSV a segmentar en cada imatge. Una vegada obtinguda la segmentació s'eliminen tots els contorns que ocupen una àrea inferior a 100 píxels. En la secció 5.2.3 es detalla el software utilitzat per a construir el dataset.

4.4 *Hard-negatives*

Un problema heretat en la implementació amb el DBG és que els píxels de les cares sovint són confosos amb mans. La Figura 4.4 mostra com es confonen els píxels de cara amb els de les mans. Aquest problema és degut al fet que el valor que prenen les característiques en els caps i en les mans són molt semblants. Per ajudar al classificador a separar més eficaçment ambdues estructures existeix una tècnica que consisteix en la incorporació d'un percentatge més gran dels exemples negatius que tenen tendència a classificar-se malament al conjunt d'entrenament. A aquests exemples se'ls coneix com a *hard-negatives* [30].

Per a implementar la tècnica el primer pas és recopilar els *hard-negatives* de les imatges, procés que s'ha dut a terme en tres passes. Primer, s'ha aplicat a totes les imatges tres detectors de cares seqüencialment: els detectors clàssics `Haar` i `lbf` [31], i també una implementació basada en xarxes neuronal convolucionals definida a [32]. De cada cara detectada s'han recopilat 100 exemples aleatoris, tots ells són els que s'afegiran al conjunt d'entrenament com a *hard-negatives*. Finalment, aquests exemples s'han separat a parts iguales per a afegir el mateix nombre d'exemples a dins de cada subconjunt d'entrenament. A l'hora de fusionar els nous exemples a

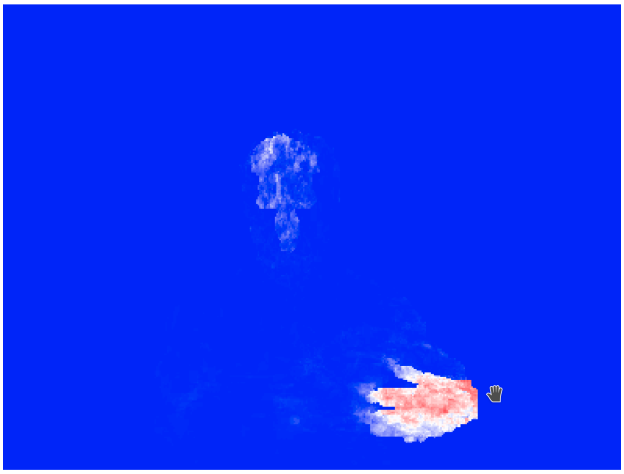


Figura 4.4: Imatge de probabilitat dels píxels. El color vermell indica una alta de probabilitat de ser mà, el blanc els píxels confusos i el blau indica els que no tenen cap probabilitat de ser d'una mà, tal com s'explica en la secció B.2.2.

dins el conjunt d'entrenament s'ha aprofitat per a mesclar aleatòriament les dades de tots els subconjunts, garantint que el nombre d'exemples positius i negatius de cada subconjunt estigui balancejat. Com a resultat es tenen una sèrie de subconjunts d'entrenament, on cada un conté exemples de totes les imatges i de totes les cares que apareixen en el dataset. Fins al moment cada un d'aquests subconjunts estava format per les dades d'unes imatges concretes.

La passa final ha estat entrenar i testejar el RDF amb els nous subconjunts d'entrenament, amb el que s'ha validat que els resultats augmentaven lleugerament amb l'aplicació d'aquesta tècnica.

En aquest capítol s'han definit les tasques principals que s'han dut a terme durant l'execució del projecte, cada una s'ha dut a terme en una iteració del desenvolupament. En la secció 4.1 s'han definit la implementació inicial, detallant els paràmetres del mètode. En la secció 4.2 s'ha exposat com s'obté la informació a processar directament de la *Kinect v2*. En la secció 4.3 s'han exposat les motivacions que han dut a crear el nou dataset, i com s'ha creat. Per acabar, la secció 4.4 parla del darrer experiment realitzat, la incorporació d'exemples negatius de cares al conjunt d'entrenament. Als pròxims capítols entrarem en els detalls d'aquests experiments, al capítol 5 es mostren els resultats obtinguts amb cada un, i en el capítol 6 es discuteixen els seus resultats.

CAPÍTOL 5

RESULTATS

De l'execució del projecte s'han obtingut dos tipus de resultats. Els principals són els obtinguts de l'evolució del detector de mans. Per a cada versió s'ha obtingut una sèrie de mètriques i imatges de probabilitat que serveixen per a avaluar l'evolució del mètode. Els altres resultats són una sèrie de productes elaborats per a executar el projecte, tots aquests productes estan disponibles públicament. Entre aquests hi ha una sèrie d'elements de programari, llibreries i programes executables; una plana web que presenta els resultats del projecte, i el DBG. Per una altre part, s'ha iniciat una línia de recerca sobre la qual es pot continuar treballant.

En aquest capítol es presenten els resultats obtinguts, a l'annex B es poden trobar els detalls sobre la presentació dels resultats. Aquest s'estructura en dues seccions principals, la primera secció, la 5.1, mostra com van variant els resultats amb l'evolució del prototipus. En la secció 5.2 es presenten tots els productes resultants del projecte.

5.1 Resultats prototipus

Començarem aquesta secció presentant els resultats de la versió original del mètode, la reproducció del definit per Tompson *et al.* [3]. Els següents apartats mostren com ha evolucionat el model en les distintes iteracions de prototipatge. La secció 5.1.1 presenta els resultats al provar la versió original amb imatges capturades en un entorn no controlat. La secció 5.1.2 il·lustra com varien els resultats a l'entrenar i testejar el mètode amb imatges en un entorn no controlat. Finalment, la secció 5.1.3 presenta els resultats de la versió final del prototipus, a la qual se li ha aplicat una tècnica per evitar que detecti les cares com a mans.

Els resultats de la primera versió del model són prou bons. La taula 5.1 mostra les mètriques obtingudes de l'avaluació amb el conjunt de test.

En la figura 5.1 es mostren uns quants exemples dels resultats visuals obtinguts amb aquesta implementació, les imatges mostrades no han estat utilitzades per entrenar el classificador. Les dues primeres columnes mostren els resultats que es repeteixen

Primer prototipus				
	Precision	*Recall*	F_1-*score*	N. exemples
Cos	0.95	0.98	0.96	1946000
Mà	0.98	0.95	0.96	1943337
macro avg	0.96	0.96	0.96	3889337
weighted avg	0.96	0.96	0.96	3889337

Taula 5.1: Mètriques obtingudes de l'avaluació del primer prototipus amb el conjunt de test.

en la gran majoria del conjunt de test, les quals tenen uns resultats excel·lents. Per exemplificar que l'algorisme no sempre dóna uns resultats tan bons, les dues següents columnes exemplifiquen dos casos en què es detecten les mans però no s'obtenen uns resultats tan remarcats. El darrer exemple mostra una imatge on es confon la cara de l'actor amb les mans, una problemàtica que es tracta en el darrer apartat d'aquesta secció, el 5.1.3.

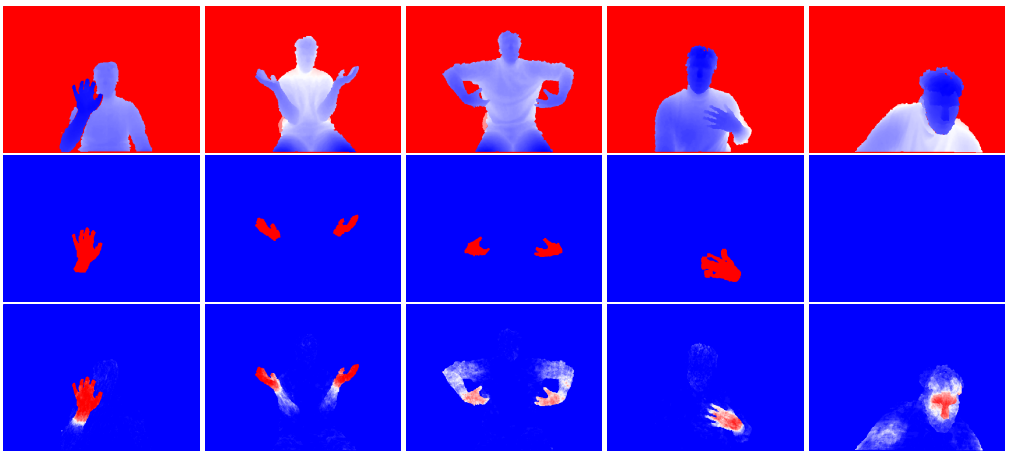


Figura 5.1: Resultats amb la rèplica del mètode de Tompson *et. al.* [3].

Tot i que les mètriques presenten uns resultats excel·lents, a la pràctica els resultats disposen d'una sèrie de problemàtiques. S'han identificat tres escenaris en el mètode en què aquest és incapaç de funcionar, aquests escenaris estan representats en la figura 5.2 i es detallen a continuació.

El primer problema detectat, representat en la primera fila de la figura 5.2, és la incapacitat de la implementació de detectar amb precisió el límit entre la mà i el braç. Sharp *et al.* al seu treball [20] mitiga aquest problema etiquetant l'avantbraç i utilitzant-lo com una tercera classe.

El segon problema és la confusió entre els colzes i les mans, aquesta problemàtica és de difícil solvència, amb el braç aplegat els colzes tenen una estructura similar a les mans en les imatges de profunditat. Està representat en la segona fila de la figura 5.2.

Finalment, el darrer problema, darrera fila de la figura 5.2, és la dificultat que té l'algorisme de detectar les mans quan no es troben al davant del cos. Aquest problema

(a) Imatges d'entrada. (b) Etiquetatge del dataset. (c) Píxels amb una probabilitat major del 50 % de ser mà.

Figura 5.2: Problemes detectats en el model.

és deu al fet que la majoria d'imatges amb les quals s'ha entrenat el RDF contenen les mans en primer pla, quan es troba amb un element que no està al davant el cos no el sap classificar com una mà.

5.1.1 Imatges món real

Per a validar la implementació del RDF es va provar el funcionament de l'algorisme amb imatges captades directament amb un dispositiu TOF, el de la *Kinect v2*. Les mètriques de l'avaluació amb aquestes imatges es mostren a la taula 5.2, extretes amb el conjunt de test del DBG del que es parlarà a la següent secció. Es pot veure com el valor de les mètriques ha disminuït considerablement, el que significa que l'algorisme no té el mateix comportament en un entorn no controlat.

La figura 5.3 mostra els resultats obtinguts en les prediccions de les imatges del món real. Les dues primeres columnes il·lustren casos en els quals es tenen les mans en primer pla, on l'algorisme funciona perfectament, són com les utilitzades per entrenar el RDF. Les altres tres columnes mostren casos en els quals no funciona, ja que la implementació no hi està preparada.

Segon prototipus				
	Precision	*Recall*	F_1-*score*	N. exemples
Cos	0.79	0.87	0.83	772000
Mà	0.83	0.73	0.78	667358
macro avg	0.81	0.80	0.81	1439358
weighted avg	0.81	0.81	0.81	1439358

Taula 5.2: Mètriques obtingudes de l'avaluació del primer prototipus amb el conjunt de test del DBG.

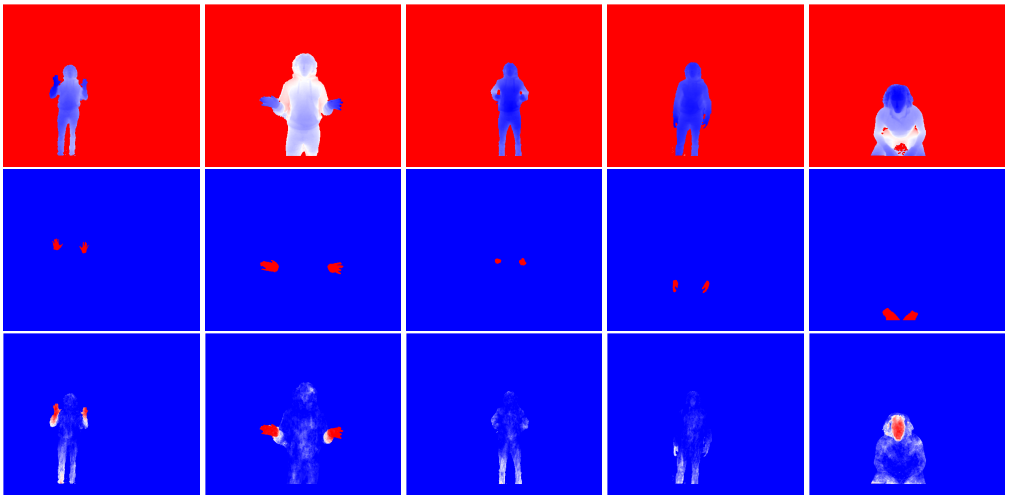


Figura 5.3: Resultats amb imatges capturades directament amb la *Kinect v2*.

5.1.2 Dataset propi

Després d'observar que el comportament no era l'esperat amb les imatges capturades directament es va construir el DBG, especificat a la secció 3.6. Les mètriques obtingudes entrenant i testejant amb el dataset es mostren a la taula 5.3. Es pot veure que els resultats són considerablement més bons que amb l'entrenament amb el dataset de la NYU, concretament les tres mètriques tenen un increment del 13 % del seu valor.

Tercer prototipus				
	Precision	*Recall*	F_1-*score*	N. exemples
Cos	0.92	0.98	0.95	772000
Mà	0.97	0.90	0.93	667358
macro avg	0.95	0.94	0.94	1439358
weighted avg	0.94	0.94	0.94	1439358

Taula 5.3: Mètriques obtingudes de l'avaluació del prototipus entrenat i testejat amb els conjunts d'entrenament i de test del DBG.

La figura 5.4 mostra els resultats obtinguts amb la nova implementació, usant els mateixos exemples que en la secció anterior. Es pot apreciar com els resultats han millorat considerablement, als exemples de la tercera i quarta columna ja es detecten les mans. I al darrer exemple ja no es detecta la cara d'una forma tan descarada.

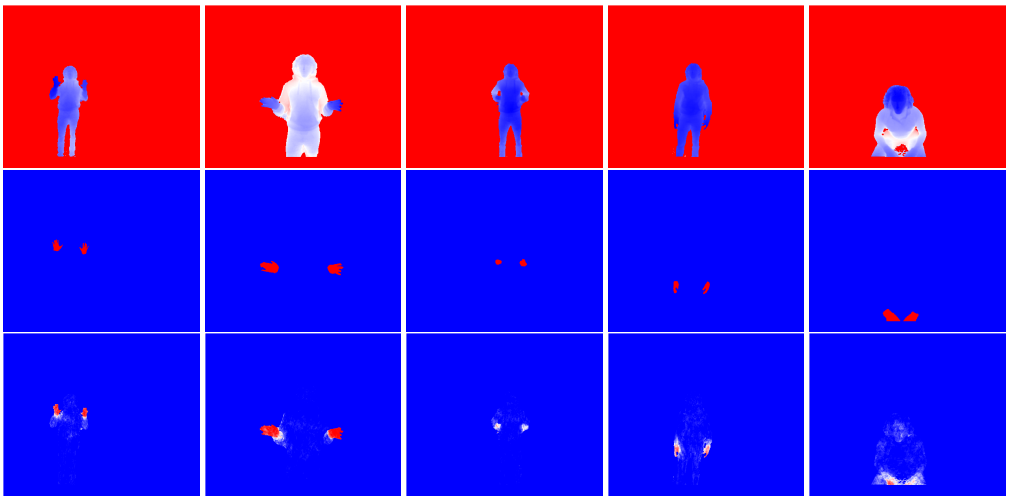


Figura 5.4: Resultats obtinguts amb el RDF entrenat amb el DBG.

En la secció 6.2 s'analitzarà com varia el comportament al processar imatges en un entorn no controlat segons el dataset utilitzat.

5.1.3 Resultats finals: Hard-negatives i conjunt de dades propi

En la darrera versió del prototipus es segueix utilitzant el DBG per a crear el RDF, aquest cop afegint els *hard-negatives* de les cares, tal com es definia al capítol 4. Es coneixen com a *hard-negatives* els exemples negatius tenen tendència a classificar-se erròniament. Als resultats presentats anteriorment els píxels de les cares són un exemple d'aquest tipus de mostres.

La taula 5.4 mostra les mètriques obtingudes entrenant el mètode afegint els exemples *hard-negatives*. Es pot veure com la variació dels resultats no és tan significativa com amb el canvi del dataset d'entrenament.

Quart prototipus				
	Precision	*Recall*	F_1-*score*	N. exemples
Cos	0.94	0.98	0.96	772000
Mà	0.97	0.93	0.95	667358
macro avg	0.96	0.95	0.95	1439358
weighted avg	0.95	0.95	0.95	1439358

Taula 5.4: Mètriques obtingudes de l'avaluació del prototipus entrenat i testejat amb el DBG afegint exemples *hard-negatives* als conjunts d'entrenament i test.

La figura 5.5 mostra els resultats obtinguts en aquesta versió. Els resultats han millorat molt lleugerament respecte a l'exemple anterior, ara es troben les mans identificades marcades més fortament. La cara que apareix al darrer exemple ara pràcticament no es detecta com a mà.

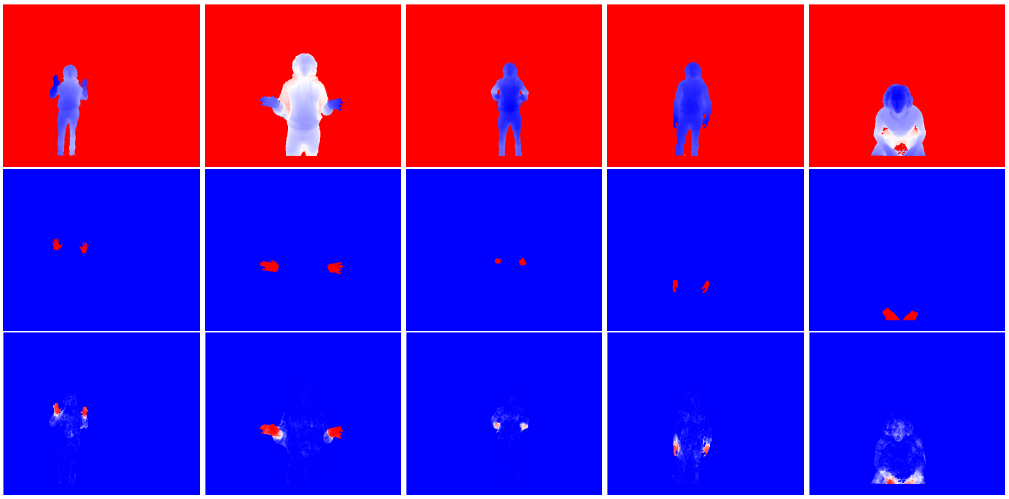


Figura 5.5: Resultats augmentant el nombre de *hard-negatives* al DBG.

En el següent capítol, secció 6.3, es discuteix l'efecte d'aquests canvis en l'estructura de les dades en el comportament del model.

5.2 Productes

Per a obtenir els resultats presentats en la secció anterior s'ha generat una sèrie de software. A part, s'ha elaborat el DBG i una plana web amb els resultats. Tots els productes s'han publicat obertament a la xarxa. Els distints apartats d'aquesta secció detallen aquests productes.

La taula 5.5 conté un resum dels productes obtinguts. Es pot accedir al contingut clicant sobre el nom del producte.

5.2.1 Llibreria de detecció de mans en imatges de profunditat

El producte més important del projecte és la llibreria *Hands rdf*. Conté la funcionalitat per a detectar mans en imatges de profunditat de forma transparent per a l'usuari. Està format per dues classes principals: una que serveix de farcell, *wrapper*, per al RDF i una altra per a calcular les característiques dels píxels indicats d'una imatge. També inclou un mòdul que facilita l'accés i la processament de les dades dels datasets. A part, disposa d'una classe que agrupa tots els paràmetres del mètode.

Contingut

La llibreria està formada per tres classes principals: `RDF`, `Features` i `Config`. També disposa d'una sèrie d'utilitats en forma de classes i funcions que faciliten l'entrenament del RDF.

Productes resultants	
Producte	Descripció
`Hands rdf`	Llibreria amb les utilitats per a detectar mans en imatges de profunditat en temps real.
`Hands cv`	Scripts que fan ús de la llibrería `Hands rdf` per a construïr els distints prototipus del projecte. També conté els scripts per obtenir els resultats de cada versió (mètriques i imatges de probabilitat).
`kinect frame reader`	S'encarrega de capturar frames en temps real de la Kinect v2, aplica el *background substraction* a cada frame capturat. Registra la imatge de profunditat, la imatge de color mapejada a sobre de la de profunditat i els resultats del *background substraction*.
`kinect dataset builder`	Converteix les imatges capturades amb el `kinect frame reader` al format usat per `Hands rdf`. A més, és el software encarregat d'assignar les etiquetes a les mans de forma semiautomàtica.
DBG	Dataset d'imatges de profunditat on totes les persones apareixen amb les mans etiquetades. S'ha fabricat amb els productes anteriors. Els detalls del seu contingut estan definits a la secció 3.6.
Web amb els resultats	Plana web que presenta els resultats de cada iteració del desenvolupament.

Taula 5.5

La classe `RDF` és hereva de la classe `RandomForestClassifier` de la llibreria *scikit-learn*, per tant comparteix totes les seves funcions i atributs. Quan s'instancia un objecte d'aquesta classe es carrega una instància de l'objecte ja entrenat, que es pot usar directament per a fer les prediccions de noves imatges de la mateixa manera que s'usaria qualsevol algorisme d'aprenentatge automàtic supervisat de la llibreria *scikit-learn*.

Per a fer les prediccions de les imatges és necessari calcular un conjunt de característiques de cada píxel que es vol predir, d'aquesta tasca s'encarrega la classe `Features`. La seva funció principal és `get_image_features` que rep per paràmetre una imatge de profunditat i una llista de posicions de la imatge, retorna una matriu amb el valor de les característiques dels píxels de les posicions indicades. El resultat de la funció posteriorment es pot utilitzar directament per a realitzar les prediccions amb la classe `RDF`. Per una altra banda, també disposa d'una sèrie de funcions per a depurar el funcionament de l'algorisme i per accedir al valor dels *offsets*, les quals no es detallen perquè no tenen importància per un usuari extern.

Al capítol 4 es definia el conjunt de paràmetres que requereix el mètode. Tots ells estan agrupats com atributs d'una classe a dins de la llibreria, la classe `Config`. Aquesta serveix com a nucli de l'aplicació, i és accedida des de la resta de components de la

llibreria per a accedir a qualsevol constant o variable global que utilitzin. L'estructura de la classe segueix l'estructura *singleton*, només es permet tenir una sola instància d'un objecte d'aquest tipus, qualsevol objecte de la classe que s'inicialitzi serà la mateixa instància. Això significa que per modificar qualsevol paràmetre del mètode hem de modificar un atribut de la classe que afectarà al conjunt de la llibreria.

A part dels tres elements mencionats anteriorment la llibreria disposa d'una sèrie d'utilitats per a manipular els conjunts de dades. La finalitat d'aquestes funcions és ocultar el codi d'accés a les dades per a fer l'entrenament i les prediccions del conjunt de test amb el RDF.

Ús

A l'annex A es pot trobar un exemple d'ús de la llibreria. On s'exemplifica com s'usa la llibreria i com es pot instal·lar amb un RDF ja entrenat.

5.2.2 Scripts experiments

Amb el fi de disposar del codi genèric de la llibreria separat dels seus usos s'ha creat un repositori a part, el *hands cv*, amb tots els scripts que fan ús de la llibreria. Conté dos executables principals, un per construir un conjunt de dades, i un altre per a obtenir les mètriques i visualitzar els resultats.

Per a construir els conjunts de dades utilitzats pel RDF, l'script *retrieve_data.py* converteix totes les imatges del dataset en el format que requereix l'algorisme per ser entrenat. A l'acabar l'execució es tenen les dades convertides en una sèrie de fitxers, que són els subconjunts de les dades amb què s'entrena el RDF. Cada fitxer conté una matriu `numpy` on cada fila és d'una mostra i les columnes són les seves característiques.

Una vegada es tenen els diferents conjunts de dades construïts, el proper pas és entrenar el RDF amb el conjunt d'entrenament i obtenir les mètriques del seu rendiment amb el de test. D'aquesta tasca s'encarrega l'script executable *propab_maps.py*, que una vegada té el RDF entrenat genera les mètriques i tots els mapes de probabilitat de les imatges del conjunt de test. Incorpora l'opció de generar un vídeo amb totes les imatges de probabilitat generades.

Per a la construcció del dataset amb els *hard-negatives* disposa de tres scripts auxiliars. Un per a obtenir les posicions de les cares en totes les imatges. Un altre per a calcular els fitxers d'entrenament dels píxels de les cares. I un darrer script que s'encarrega de mesclar les dades obtingudes amb *retrieve_data.py* amb les dades de les cares, amb el que s'obté les dades utilitzades per a entrenar la darrera versió del prototipus.

A part, al repositori s'inclouen un conjunt de funcions auxiliars útils per a simplificar l'accés a la llibreria i a les imatges.

Per a executar el procés complet de l'obtenció dels resultats d'un prototipus s'ha d'executar com a mínim *retrieve_data.py* i *propab_maps.py*. Per no haver d'executar manualment els scripts un rere l'altre el projecte disposa de dos scripts `bash`, un per a les primeres versions del prototipus i un altre per a la darrera versió.

5.2.3 Programari per a crear el dataset

La creació del DBG ha requerit dues funcionalitats auxiliars: una per a la captació de les imatges del dispositiu TOF i l'altre per a processar-les per a obtenir les etiquetes de les mans.

Per a la captació dels quadres de la *Kinect v2* s'ha implementat el *kinect frame reader*. El programa accedeix a l'API del dispositiu per a fer captura dels quadres cada segon. Al fer una captura d'un quadre emmagatzema en carpetes independents la següent informació: la imatge de profunditat, la imatge de color, la imatge de color mapejada a sobre de la de profunditat i els resultats del *backgroud substraction* del quadre. A l'acabar l'execució es tenen quatre carpetes, una per cada tipus d'informació capturada amb totes les captures.

Encara fa falta obtenir les etiquetes dels píxels de mà, amb aquest fi s'ha implementat una GUI que permet seleccionar un rang dels valors a segmentar i transforma les imatges en el mateix format que les del dataset de la NYU. Aquesta GUI és el *kinect dataset builder*. El software permet iterar sobre els frames capturats amb la *Kinect v2*. A cada un permet seleccionar els llindars que s'aplicaran als canals HSV per a obtenir les etiquetes. El llindar seleccionat es conservarà pel següent frame, facilitant així la segmentació, ja que dos frames consecutius tindran unes condicions semblants. La figura 5.6 conté una captura de pantalla de la GUI desenvolupada. A la part inferior de la finestra hi ha les barres que permeten seleccionar els llindars superiors i inferiors de cada canal per a filtrar cada imatge. Pressionant la tecla "w" es guarden les imatges amb el mateix format que tenien les del dataset de la NYU i la interfície automàticament s'actualitza amb el següent frame.

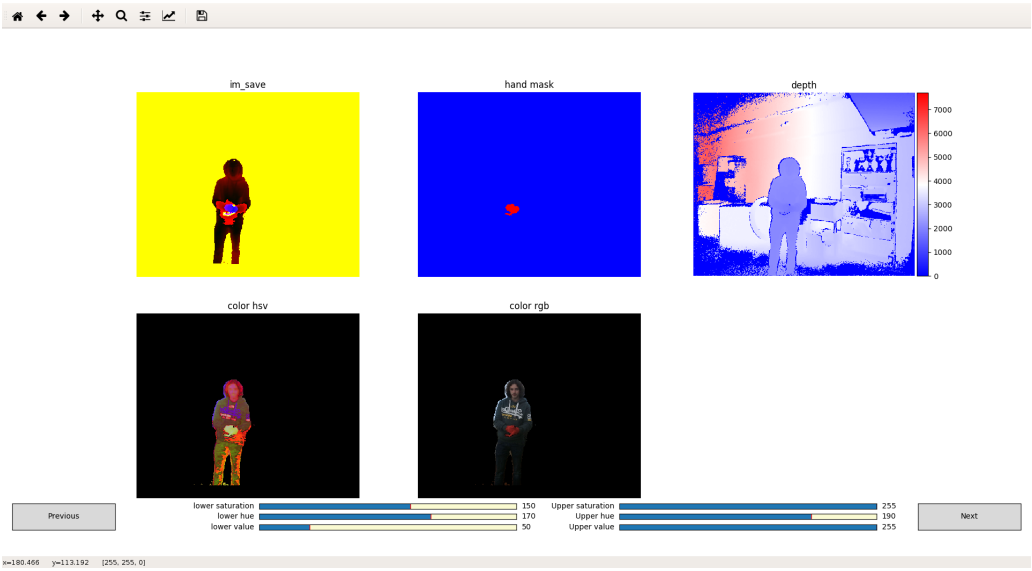


Figura 5.6: Captura de pantalla de la GUI implementada per etiquetar el dataset.

5.2.4 Conjunt de dades

El dataset construït durant el desenvolupament del projecte, el DBG, és una aportació important del projecte. Està disponible públicament associat a la llibreria de detecció

de mans.

A la secció 3.6 s'han definit els detalls del contingut del dataset. I a la secció 4.3 les motivacions que han dut a crear-lo.

5.2.5 Plana web

No tots els productes resultants han estat elements de programari. Els resultats més importants del projecte són com ha anat evolucionant el comportament del prototipus al llarg de les iteracions, el contingut de la secció 5.1. Tot aquest contingut s'ha incorporat en una plana web associada al repositori de la llibreria.

En aquest capítol s'han presentat tots els resultats derivats de l'execució del projecte. Se n'han generat dos tipus: mètriques dels distints prototipus i una sèrie de productes. En la secció 5.1 es defineixen les mètriques obtingudes amb cada un dels prototipus generats i també es mostren exemples de les imatges de probabilitat obtingudes amb cada un. L'altra secció del capítol, la 5.2, presenta tots els productes generats durant el projecte.

CAPÍTOL 6

DISCUSSIÓ

Aquest capítol pretén analitzar el comportament del model segons les propietats que s'han anat estudiant durant el desenvolupament del projecte: paràmetres, conjunts de dades i estructura de les dades (*hard-negatives*). La finalitat del capítol és il·lustrar als lectors com afecten al comportament global del sistema i que puguin conèixer que s'ha de modificar per a obtenir un comportament desitjat.

El capítol s'estructura de la següent manera. En la secció 6.1 es discuteix com afecta al model el valor dels diferents paràmetres del sistema. En la secció 6.2 es presenta com afecta el dataset usat als resultats. En la secció 6.3 es discuteix en quina mesura afecta als resultats la incorporació d'exemples *hard-negatives* als conjunts d'entrenament. Per acabar, la secció 6.4 presenta les hipotètiques possibles futures iteracions del desenvolupament o noves línies de recerca per on es pot continuar.

6.1 Afectes de la parametrització

El comportament del model es pot ajustar amb una sèrie de paràmetres. En aquesta secció es mostra com varia el comportament del model alterant el valor dels paràmetres definits en la secció 4.1.

6.1.1 Paràmetres RDF

Els paràmetres del RDF són els que defineixen com es construirà el classificador. Per tant, la seva importància és crucial per a obtenir uns bons resultats.

Un dels seus paràmetres més importants és la profunditat màxima dels arbres de decisió. Aquest paràmetre limita el nombre de nivells que poden tenir els arbres que formen el bosc. Jugant amb aquest valor es pot controlar el que aprèn l'algorisme. Un valor petit d'aquest paràmetre construirà arbres amb massa pocs nivells per a ajustar-se a les dades d'entrenament, efecte que es coneix com a *under-fitting*. En canvi, un valor massa alt farà que els arbres entrin en massa detall i s'aprenguin el conjunt d'entrenament, *over-fitting*, efecte del qual ja s'ha parlat a la secció 3.4.

La figura 6.1 mostra com afecta l'evolució de la profunditat màxima dels arbres al comportament del sistema. Es pot veure com a partir dels 15 nivells assoleix una cota màxima i afegint profunditat els resultats es mantenen estables. El que significa que la profunditat màxima òptima està sobre els 15 nivells. Si se n'utilitzen més, a més d'augmentar la complexitat del model, al testejar amb un altre conjunt de dades es podria caure en l'efecte del *over-fitting*.

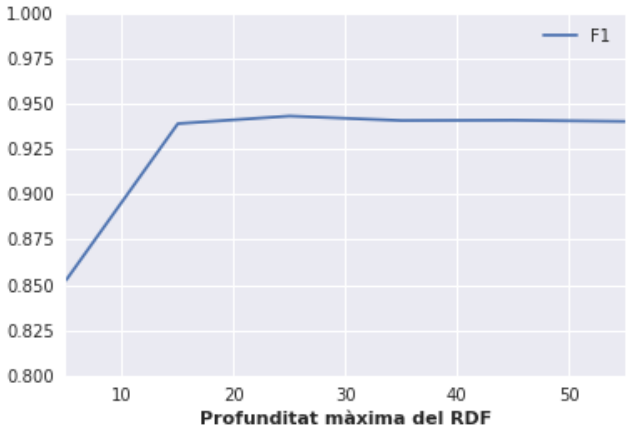


Figura 6.1: Evolució del F_1 -*score* del model a mesura que augmentem la profunditat màxima dels arbres.

Un altre paràmetre considerat és el nombre mínim d'exemples que pot tenir un node per a poder ser una fulla. La seva funció és la mateixa que la profunditat màxima dels arbres, limitar la mida dels arbres, i per tant serveix per controlar el *under-fitting* i el *over-fitting*. Quan s'ha de dividir un node, si algun dels dos fills té un nombre d'exemples inferior a l'especificat pel paràmetre no es dividirà i serà una fulla. Tot i que aquest paràmetre és important per a controlar el comportament del model, ha estat suficient analitzar la profunditat màxima dels arbres per controlar la mida dels arbres.

Finalment, el darrer paràmetre estudiat és el nombre d'arbres de decisió entrenats per a cada subconjunt d'entrenament. L'entrenament del RDF es feia per blocs, es dividia el conjunt d'entrenament en n subconjunts i s'entrenava un determinat nombre d'arbres amb cada un. La figura 6.2 mostra com varia el comportament del model segons el nombre d'arbres entrenats amb cada subconjunt. Es pot veure com el nombre d'arbres utilitzat no és un factor rellevant, la variació del F_1 -*score* entre usar un arbre a usar-ne quinze va de 0.86 a 0.94, a partir dels 15 l'increment del nombre d'arbres no s'aconsegueix millorar les classificacions. Per tant, el nombre òptim d'arbres per a la implementació són 15 per cada subconjunt, incrementant el seu nombre només s'aconsegueix incrementar la complexitat del model.

6.1.2 Configuració característiques

Un factor clau del funcionament del model és l'estructura de les característiques que es prenen dels píxels. Aquestes defineixen la informació que s'aprendrà per a cada mostra. Per tant, la seva qualitat és clau pel funcionament de l'algorisme, com millor

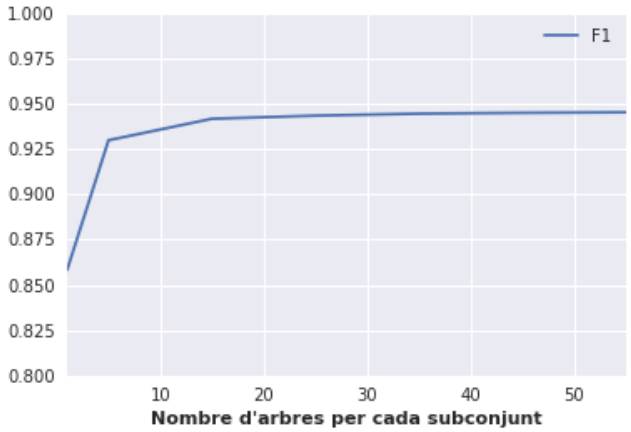


Figura 6.2: Evolució del F_1 -*score* del model segons el nombre d'arbres entrenat per cada subconjunt d'entrenament.

informació proporcionin l'algorisme podrà realitzar unes millors prediccions. En els següents apartats es discuteix l'efecte dels paràmetres que les afecten.

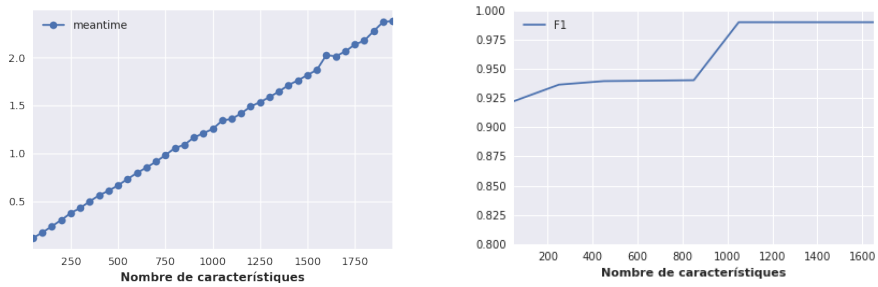
Nombre de característiques

El nombre de característiques que es calcularan per a cada exemple afecta als resultats. Quantes més característiques es tinguin més informació de les mostres tindrà el RDF. Ara bé l'increment del nombre de característiques té la contrapartida que amb ella també augmenten la quantitat de recursos computacionals necessaris, tant la memòria necessària per emmagatzemar les característiques com l'increment del nombre d'operacions aritmètiques.

A la figura 6.3 es mostra com evoluciona el comportament del model segons el nombre de característiques utilitzat. En la figura 6.3a es pot apreciar com el temps de computació creix linealment amb l'increment del seu nombre al executar el codi 1. Encara que es podria modificar la implementació perquè fos més eficient, per exemple, paral·lelitzant els càlculs de les característiques en una GPU, és interessant notar que té cost lineal per a seleccionar el nombre òptim de característiques. En la figura 6.3b es mostra com evoluciona el F_1 -*score* del model amb el nombre de característiques, es pot apreciar un bot important entre les 900 característiques i les 1100, per tant un valor entre aquests seria adequat. En una aplicació en que es disposàs de pocs recursos computacionals es pot veure com es podria reduir el nombre de característiques a 100 i es seguiria tenint una F_1 -*score* major que 0.9.

Distància màxima dels *offsets*

Quan aquest paràmetre té un valor en el rang adequat els resultats varien dèbilment. El procés d'entrenament del RDF s'encarrega de descartar les característiques amb una longitud massa llarga o massa petita. Per tant a no ser que es pretengui utilitzar el nombre mínim de característiques possibles el paràmetre no té un gran impacte.



(a) Evolució del temps de càlcul d'una imatge de probabilitat segons el nombre de característiques.

(b) Evolució del F_1 -*score* del model segons el nombre de característiques.

Figura 6.3: Comportament del sistema amb l'augment del nombre de característiques.

Així i tot, un valor inadequat del paràmetre afectarà considerablement als resultats. Amb un valor massa gran moltes característiques seran inútils, comprovaran píxels que surten de la imatge, mentre que amb poques característiques les no descartades poden no bastar per fer la predicció amb exactitud. Quan se n'utilitzen moltes l'entrenament del RDF descartarà les inútils, però tindrem una implementació ineficient. En canvi, amb un valor massa petit no es rastrejaran píxels que són rellevants per a fer la predicció amb exactitud.

6.2 Conjunts de dades

Un dels factors més importants del comportament del sistema són els conjunts de dades utilitzats per a entrenar i per testejar el model. Les dades d'entrenament són les que defineixen que aprendrà el model i les de test contenen les dades per a les quals es verifica el seu funcionament.

A la secció 3.6 es detallava el contingut dels dos datasets utilitzats en el projecte. El conjunt de dades de la NYU tenia el defecte que totes les imatges contenien les mans en un primer pla, i per tant l'algorisme no detectava les mans ubicades en qualsevol altra ubicació. Amb l'objectiu de solucionar aquest problema es va elaborar el DBG.

Per avaluar el grau d'assoliment dels objectius amb el nou conjunt de dades, i comprovar si realment és més generalista, es comparen les mètriques del comportament de l'avaluació del conjunt de test del dataset propi amb l'entrenament amb els dos datasets utilitzats. És important destacar que l'entrenament amb el DBG no ha utilitzat cap imatge del seu conjunt de test.

La taula 6.1 il·lustra el comportament dels dos RDF (sense *hard-negatives*) al processar imatges d'un entorn no controlat, les columnes indiquen el dataset amb el qual s'ha entrenat el RDF. Es pot veure com els resultats són bastant més favorables amb l'entrenament amb el DBG. A més, a la figura 5.3 s'ha vist visualment que els resultats en aquests entorns de l'entrenament amb el NYU no eren el suficient precisos, no era capaç d'inferir els casos pels quals no havia estat entrenat.

Per a mostrar la diferència entre el dataset utilitzat per a entrenar la figura 6.4 mostra les imatges de probabilitat amb els dos models, on es pot apreciar una gran evolució

Mètriques en un entorn no controlat						
	NYU			DBG		
	Precision	*Recall*	F_1-*score*	*Precision*	*Recall*	F_1-*score*
Cos	0.79	0.87	0.83	0.75	0.98	0.95
Mà	0.83	0.73	0.78	0.97	0.90	0.93
macro avg	0.81	0.80	0.81	0.95	0.94	0.94
weighted avg	0.81	0.81	0.81	0.94	0.94	0.94

Taula 6.1: Comparació dels resultats obtinguts amb l'avaluació del conjunt de test del DBG.

entre els dos models. A la figura es veu clarament que amb l'entrenament amb el NYU la més ben identificada és a l'exemple de la quinta fila, que és l'única mà en primer pla. En la resta d'exemples en els quals apareixen mans en repòs o amb altres posicions es veu com el DBG les detecta correctament i l'altre no.

La conclusió és que el dataset de la NYU està pensat per un propòsit específic, trobar les mans en primer pla per a identificar els gestos. En canvi el DBG és més generalista.

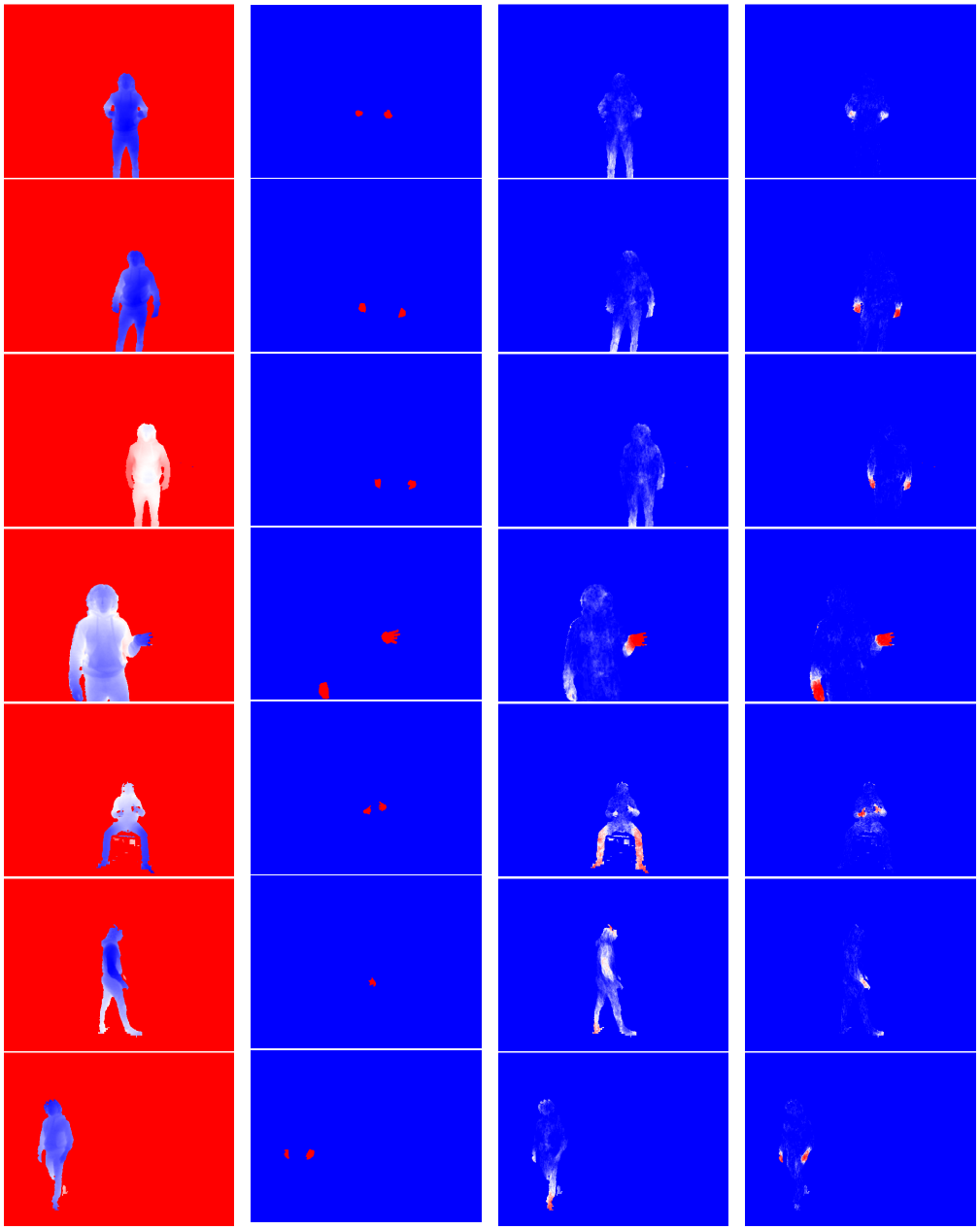
6.3 Estructura de les dades

L'estructura de les dades d'entrenament condiciona el que aprendrà el classificador. Un exemple d'això és el nombre d'exemples de cada classe que composen el conjunt d'entrenament, si es disposa de més exemples d'una classe que d'una altra l'algorisme aprendrà millor la classe que conté més exemples i ignorarà l'altre. Per a evitar això les dades que s'utilitzen per entrenar sempre tenen el mateix nombre d'exemples positius que negatius. Recordem que en la darrera versió del model s'han efectuat dos canvis en les dades utilitzades. Per una part s'ha afegit un percentatge elevat d'exemples de píxels de cares al conjunt, i per una altra s'han mesclat tots els exemples de manera que cada subconjunt tingui exemples de totes les imatges.

En aquesta secció es discutirà com afecta l'estructura de les dades al comportament global del sistema. Es començarà exposant com afecta la mida dels subconjunts de les dades que s'utilitzen per entrenar. Finalment es discutirà com afecta la incorporació d'exemples de cares i la mescla de les dades dels subconjunts.

6.3.1 Mida dels subconjunts

En la secció 3.5.2 es definia que l'entrenament es feia dividint les dades d'entrenament en subconjunts. La mida d'aquests subconjunts es defineix pel nombre d'imatges de les quals tenen dades. Com que el nombre d'imatges del dataset es fixe, a mesura que augmentem la seva mida estem disminuint el nombre de subconjunts que conformen el conjunt d'entrenament. Segons Amit *et al.* [29] en incrementar el nombre de conjunts d'entrenament, a més d'accelerar l'entrenament, es redueix el *over-fitting*. A la figura 6.5 es pot veure com la mida dels subconjunts, i el seu nombre, no afecta d'una manera notable als resultats. Tot i que es pot observar un petit pic màxim a sobre de les 2000 imatges per subconjunt.



(a) Imatge d'entrada. (b) Etiquetatge ideal. (c) Prediccions amb el NYU. (d) Prediccions amb el DBG.

Figura 6.4: Prediccions en un entorn no controlat segons el dataset utilitzat per entrenar.

6.3.2 Mescla de les dades

En les tres primeres versions del prototipus, cada subconjunt d'entrenament estava format per un subconjunt concret de les imatges. És a dir, cada un estava format pels exemples d'unes imatges concretes. Així, es tenien les imatges del dataset separades en n subconjunts disjunts, amb cada grup d'imatges es construïa un subconjunt d'entrenament. Amb aquesta tècnica cada subconjunt d'arbres aprenia les dades de les seves

Figura 6.5: Evolució del F_1-*score* del model a mesura que augmentem el nombre d'imatges en cada subconjunt d'entrenament.

imatges ignorant els casos que no apareixen en les imatges del seu grup.

Per intentar que tots els subconjunts continguin dades de tots els casos que apareixen en el dataset, en el darrer experiment es mesclen aleatòriament les dades de cada subconjunt. El resultat és que els exemples que formen cada grup d'entrenament estan formats per exemples aleatoris de tot el dataset.

A la taula 6.2 es pot veure com varien les mètriques del comportament del classificador amb la mescla dels exemples en tots els subconjunts. Les mètriques de l'esquerra contenen les de la implementació amb un conjunt concret d'imatges en cada subconjunt, les de la dreta les obtingudes amb les dades mesclades. Es pot veure com els resultats són pràcticament iguals. Com que la configuració està optimitzada, en certa manera, per a funcionar amb els subconjunts concrets, l'únic que es pot extreure és que amb ambdues tècniques s'obtenen pràcticament els mateixos resultats. Per tant, amb aquesta tècnica no s'ha aconseguit millorar el prototipus. S'ha observat que variant els paràmetres els resultats tenen una tendència a donar uns millors resultats amb les dades barrejades. Per consistència amb la resta del document s'han presentat els resultats amb la mateixa configuració que en la resta del document, la que es defineix en l'annex B.

Comparativa amb els exemples barrejats						
	DBG			Mescla		
	Precision	*Recall*	F_1-*score*	*Precision*	*Recall*	F_1-*score*
Cos	0.95	0.98	0.96	0.94	0.98	0.96
Mà	0.98	0.95	0.96	0.97	0.93	0.95
macro avg	0.96	0.96	0.96	0.96	0.95	0.95
weighted avg	0.96	0.96	0.96	0.95	0.95	0.95

Taula 6.2: Comparació del comportament amb les dades normals i les dades barrejades.

6.3.3 Exemples hard-negatives

La darrera modificació del conjunt de dades va ser la incorporació d'exemples negatius de cares. Com s'ha vist en el capítol 5 l'algorisme té una tendència a confondre aquestes dues estructures. Amb aquests exemples dins el conjunt de dades es pretén que l'algorisme aprengui millor a diferenciar entre les dues estructures.

La taula 6.3 mostra la variació del comportament del model amb aquesta modificació del conjunt de dades. Es pot apreciar que la qualitat del model és la mateixa, les mètriques prenen els mateixos valors, varien en el quart decimal.

Comparativa de l'efecte dels exemples *hard negatives*						
	Mescla			Mescla + *hard negatives*		
	Precision	*Recall*	F_1-*score*	*Precision*	*Recall*	F_1-*score*
Cos	0.94	0.98	0.96	0.94	0.98	0.96
Mà	0.97	0.93	0.95	0.97	0.93	0.95
macro avg	0.96	0.95	0.95	0.96	0.95	0.95
weighted avg	0.95	0.95	0.95	0.95	0.95	0.95

Taula 6.3: Comparació del comportament amb les dades barrejades i amb els exemples *hard negatives*.

Amb la modificació de l'estructura de les dades no s'ha obtingut cap millora rellevant. S'ha de tenir en compte que s'està treballant amb un model quasi perfecte. Els casos que li fallen per a ser un predictor ideal són els problemes 1 i 2 que es mostren en la figura 5.2. Aquests problemes són a causa de la informació que proporcionen les característiques, i no es solucionen canviant l'estructura de les dades o amb un altre dataset.

6.4 Treball futur

L'execució del projecte ha servit per adquirir un coneixement detallat del funcionament de l'algorisme. Aquest és pràcticament immillorable sense substituir elements bàsics de la implementació, com el RDF o les característiques. L'alt grau de comprensió del model ha permès definir possibles treballs futurs a sobre de la tècnica.

6.4.1 Tracking

Els resultats de la versió final encara conserven la majoria de problemes elementals del mètode, els quals s'ha arribat a la conclusió que són inherents del mètode i no es poden solucionar. Són casos en què les característiques dels píxels dels elements confosos prenen els mateixos valors, com els colzes.

Una manera de solucionar aquest problema seria afegir informació temporal i de la colorimetria de l'escena. De manera que l'algorisme aprofités la informació de la ubicació de la mà en el frame anterior per a identificar la posició de la mà en l'actual. Tècniques com el *MeanShift* [33], el *particle filter* [34] o el *optical flow* [35] podrien suposar un bon punt de partida.

6.4.2 Detecció de formes

De l'observació del comportament del mètode s'ha inferit que aquesta mateixa tècnica és capaç de detectar formes dels objectes. Així, aquest mètode, entrenat amb un dataset en el qual en lloc d'estar etiquetades les mans estiguin etiquetats distints tipus d'objectes, serà capaç de detectar a quin objecte pertany cada píxel segons la forma de l'objecte.

Una extensió d'aquest treball consisteix a usar un RDF i les mateixes característiques per a comprovar que l'algorisme és capaç de discriminar objectes segons la seva forma.

En aquest capítol s'ha analitzat el comportament del mètode d'acord a una sèrie de factors. En la secció 6.1 s'ha discutit el comportament del mètode segons els valors dels seus paràmetres. En la secció 6.2 s'ha analitzat el comportament de la tècnica segons el conjunt d'entrenament usat, que s'ha vist que afecta considerablement als resultats. En la secció 6.3 s'ha discutit l'efecte de la modificació de l'estructura de les dades del dataset, s'ha vist que la modificació de la seva estructura no té uns resultats considerables. Per acabar, s'ha tancat el capítol en la secció 6.4 exposant els possibles treballs futurs derivats d'aquest.

CAPÍTOL 7

CONCLUSIONS

La detecció de mans en imatges és un camp que ha estat treballat per molts d'autors. La majoria ho treballen en imatges en les quals únicament disposen de la informació del color, ja que són les més comunes. Amb l'aparició dels sensors de profunditat aparegueren un nou grup de metodologies, de les quals destaca la de Shotton *et al.* [19]. La seva tècnica tot i no estar enfocada amb la detecció de mans va ser estesa per Tompson *et al.* [3] amb aquesta funció, el treball presentat en aquest document parteix del seu.

La tècnica presentada consisteix en la classificació dels píxels de la imatge segons si pertanyen a una mà mitjançant aprenentatge automàtic. S'aconsegueix calculant una sèrie de característiques dels píxels d'un conjunt d'entrenament, les quals consisteixen en la diferència entre dues posicions relatives a ell, per a després entrenar un RDF.

Durant el treball s'ha intentat millorar la comprensió en detall del mètode i intentar millorar-lo. Amb aquest objectiu, s'han realitzat una sèrie de tasques i experiments que han anat millorant el mètode en més o menys mesura.

La primera tasca que s'ha dut a terme ha estat la implementació inicial del mètode amb la qual s'ha observat que el funcionament d'aquest depenia de dos factors: una sèrie de paràmetres i el conjunt d'entrenament. La segona tasca ha estat el disseny d'un nou conjunt de dades per a entrenar i testejar el mètode, en aquesta tasca s'ha observat com el comportament del mètode varia notablement amb el conjunt d'entrenament utilitzat. S'ha passat de tenir un mètode centrat en la detecció de mans en primer pla a un mètode generalitzable a mans que apareixen a qualsevol banda de l'escena. La darrera tasca ha consistit en la modificació de les dades d'entrenament de manera que incorporessin un major nombre dels exemples negatius que normalment es classifiquen malament, amb aquest darrer experiment pràcticament no hem aconseguit millorar el mètode.

Els resultats aconseguits amb els diferents prototipus són excel·lents, obtenen una F_1*-score* major que 0.9 en tots ells. La majoria dels casos en els quals no funciona són a causa de problemes inherents del mètode, on les característiques utilitzades no proporcionen suficient informació, per exemple les característiques dels píxels

d'un colze arreplegat en primer pla prenen els mateixos valors que els de les mans en molts de casos. Si s'inspecciona les imatges de probabilitat dels resultats (estan tots disponibles al web del projecte), s'observa que la gran majoria dels casos en els quals falla coincideixen amb els problemes 1 i 2 de la figura 5.2. Això significa que s'ha arribat als resultats màxims que es poden obtenir amb la tècnica, per això per molt que es modifiqui l'estructura de les dades o es manipulin els paràmetres s'aconsegueix augmentar la qualitat del sistema.

APÈNDIX A

PROGRAMARI HANDS RDF

Un dels productes resultants més importants d'aquest projecte ha estat una llibreria amb les funcionalitats per a detectar mans en imatges de profunditat, detallada en la secció 5.2.1. En aquest apèndix es presenta com s'utilitza la llibreria.

A.1 Instal·lació

La llibreria es pot instal·lar en un entorn *python* amb el programa *pip*. Inclou el RDF entrenat amb el dataset DBG i amb la configuració que es mostra a l'apèndix B. La instal·lació es pot fer executant la comanda:

```
pip install hands-rdf
```

A.2 Ús

El codi 1 mostra un exemple de l'ús de la llibreria, donada una imatge obtenir les probabilitats dels píxels de les persones que hi apareixen de ser d'una mà. L'exemple fa ús de les tres classes principals de la llibreria.

L'exemple comença important el codi de les tres classes de la llibreria utilitzades, línies 1-3. Després crea una instància del classificador i una instància de la classe per calcular les característiques, línies 5 i 6. A la línia 8 es carrega una imatge de profunditat a dins de la variable `depth_image`, que ha de ser una matriu d'enters de 16 bits.

Una vegada es tenen els principals elements de l'exemple carregats a dins variables. A la línia 10 es calculen totes les posicions de la imatge que no pertanyen al fons de la imatge. Notar que s'ha fet ús de la classe `Config` per a saber el valor que tenen el fons de les imatges, recordem que aquesta classe conté totes les constants i paràmetres necessàries per a fer ús de la llibreria. Utilitzant la imatge de profunditat i les posicions candidates a ser d'una mà a la línia 11 es calculen les característiques dels candidats. Com a resultat, a la variable `features` es té una matriu en la qual cada fila conté les característiques del píxel de la mateixa posició en la llista de posicions.

El darrer pas consisteix en l'obtenció de les prediccions de les posicions. A la línia 13 es crea una imatge amb tots els seus valors a zero, que serà una imatge de probabilitats. Després, utilitzant la matriu *features* directament com a paràmetre del mètode `predict_proba` del *RDF* a la línia 15 s'obtenen les probabilitats de cada píxel de ser d'una mà. Per acabar a la línia 16 es consulten les probabilitats obtingudes en la classificació i es copien a les posicions de la imatge de probabilitat.

```
1  from hands_rdf.RDF import RDF
2  from hands_rdf.features import Features
3  from hands_rdf.Model.Config import config
4
5  clf = RDF()
6  f = Features()
7
8  depth_image = "Càrrega una imatge de profunditat de enters de 16 bits"
9
10 indexs = np.nonzero(depth_image < config.BG_DEPTH_VALUE)
11 positions, features = f.get_image_features(depth_image, indexs)
12
13 proba_mask = np.zeros((depth_image.shape[0], depth_image.shape[1]),
14                        dtype=np.float32)
15 predicted = clf.predict_proba(features)
16 proba_mask[positions] = predicted[:, 1]
```

Codi 1: Exemple d'ús de la llibreria.

APÈNDIX B

FORMAT DELS RESULTATS

Aquest apèndix conté tota la informació sobre els resultats presentats en el document. En la primera secció es detalla la configuració per defecte dels paràmetres utilitzada en tots els experiments. La segona secció defineix les mètriques utilitzades per avaluar el comportament del model. En la tercera secció es detalla el format de les imatges de probabilitat que es mostren al llarg del document.

B.1 Configuració execució

Amb el fi de mostrar uns resultats consistents, s'ha utilitzat la mateixa configuració per a extreure tots els resultats mostrats en el document. La taula B.1 mostra el valor que prenen els paràmetres del sistema en aquesta configuració.

Paràmetres	
Paràmetre	**Valor**
Valors dels *offests*	de -120000 a 120000
Nombre de característiques	1000
Nombre d'imatges en cada subconjunt	2000
Nombre de píxels de cada classe pressos en cada imatge per a crear el conjunt de dades	500
Nombre d'arbres entrenats per a cada subconjunt de les dades	15
Profunditat màxima dels arbres	30
Nombre d'exemples mínim en un node per a poder esser fulla	1

Taula B.1: Valors dels paràmetres utilitzats per defecte per a l'obtenció dels resultats.

B.2 Presentació del resultats

Per a presentar els resultats del projecte s'utilitzen una sèrie de mètriques per a comprovar el funcionament del model. I unes imatges que mostren la probabilitat que té cada píxel de ser una mà, les imatges de probabilitat.

B.2.1 Mètriques

Les mètriques utilitzades són extretes amb un RDF entrenat amb el conjunt d'entrenament, i s'obtenen a partir dels resultats de les prediccions del conjunt de test. En el document s'utilitzen tres mètriques: la *Precision*, el *Recall* i el $F1-score$.

El *Recall*, també anomenat *True Positive Rate* (TPR), és la proporció de píxels positius classificats correctament del total de píxels positius. Essent P el nombre d'exemples positius i TP el nombre de píxels positius classificats correctament es correspon a la fórmula:

$$Recall = \frac{TP}{P} \tag{B.1}$$

La *Precision* és el percentatge de píxels positius classificats correctament de tots els classificats com a positius. Essent FP el nombre d'exemples negatius classificats com a positius es correspon a la fórmula:

$$Precision = \frac{TP}{TP+FP} \tag{B.2}$$

Finalment, la F_1*-score* és la mitjana harmònica de la *Precision* i el *Recall*:

$$\text{F}_1\text{-score} = 2 \cdot \frac{Precisió \cdot TPR}{Precisió + TPR} \tag{B.3}$$

Per a presentar el valor mitjà de les mètriques en totes les classes s'utilitzen les següents expressions:

- *macro avg*: Mitja de les mètriques per a cada classe, donant el mateix pes a totes les classes.
- *weighted avg*: Mitja de les mètriques de cada classe ponderada pel nombre d'exemples de cada classe en el conjunt de test.

B.2.2 Imatges de profunditat

A part de les mètriques del model que presenten com es comporta el sistema en conjunt al document també es presenten imatges de probabilitat. Aquestes imatges mostren la probabilitat que té cada píxel de pertànyer a una mà, amb un mapa de color divergent. La figura B.1 mostra el mapa de color d'aquestes imatges on el color blau significa que el píxel té una probabilitat del 0 % de ser mà, el blanc un 50 % i el vermell un 100 %.

Figura B.1: Mapa de color de les imatges de probabilitat.

BIBLIOGRAFIA

[1] X. Zhang, Z. Fu, W. Cai, D. Tian, and J. Zhang, "Expert Systems with Applications Applying evolutionary prototyping model in developing FIDSS : An intelligent decision support system for fish disease / health management," *Expert Systems With Applications*, vol. 36, no. 2, pp. 3901–3913, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2008.02.049 (document), 2.1, 2.1

[2] E. Lachat, H. Macher, T. Landes, and P. Grussenmeyer, "Assessment and calibration of a RGB-D camera (Kinect v2 Sensor) towards a potential use for close-range 3D modeling," *Remote Sensing*, vol. 7, no. 10, pp. 13 070–13 097, 2015. (document), 4.1

[3] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Transactions on Graphics (ToG)*, vol. 33, no. 5, p. 169, 2014. (document), 1, 1.1, 1.2, 1.2, 1.3, 3.5.1, 3.6, 3.6.1, 5.1, 5.1, 7

[4] C. Zimmermann and T. Brox, "Learning to estimate 3d hand pose from single rgb images," in *International Conference on Computer Vision*, vol. 1, 2017, p. 3. 1, 1.1

[5] Y. Wen, C. Hu, G. Yu, and C. Wang, "A robust method of detecting hand gestures using depth sensors," in *Haptic Audio Visual Environments and Games (HAVE), 2012 IEEE International Workshop on*. IEEE, 2012, pp. 72–77. 1, 1.1

[6] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun, "Hand pose estimation and hand shape classification using multi-layered randomized decision forests," in *European Conference on Computer Vision*. Springer, 2012, pp. 852–863. 1

[7] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," *Communications of the ACM*, vol. 54, no. 2, pp. 60–71, 2011. 1

[8] I. Ayed, B. Moyà-Alcover, P. Martínez-Bueso, J. Varona, A. Ghazel, and A. Jaume-I-Capó, "Validación de dispositivos RGBD para medir terapéuticamente el equilibrio: El test de alcance funcional con Microsoft Kinect," *RIAI - Revista Iberoamericana de Automatica e Informatica Industrial*, vol. 14, no. 1, pp. 115–120, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.riai.2016.07.007 1

[9] F. Xu, "Human Detection Using Depth and Gray Images Kikuo Fujimura 800 California St , Mountain View CA," *Proceedings of the IEEE*, 2003. 1

[10] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *IEEE transactions on cybernetics*, vol. 43, no. 5, pp. 1318–1334, 2013. 1

[11] J. S. Supančič, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan, "Depth-Based Hand Pose Estimation: Methods, Data, and Challenges," 2018. 1

[12] Z. Qiu-yu, L. Jun-chi, Z. Mo-Yi, D. Hong-xiang, and L. Lu, "Hand gesture segmentation method based on ycbcr color space and k-means clustering," *Interaction*, vol. 8, pp. 106–116, 2015. 1.1

[13] A. Mittal, A. Zisserman, and P. H. Torr, "Hand detection using multiple proposals." in *BMVC*. Citeseer, 2011, pp. 1–11. 1.1

[14] M. Van den Bergh and L. Van Gool, "Combining rgb and tof cameras for real-time 3d hand gesture interaction," in *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*. IEEE, 2011, pp. 66–72. 1.1

[15] J. Lin, X. Ruan, N. Yu, and J. Cai, "Multi-cue based moving hand segmentation for gesture recognition," *Automatic Control and Computer Sciences*, vol. 51, no. 3, pp. 193–203, 2017. 1.1

[16] G. Moyà-Alcover, A. Elgammal, A. Jaume-i Capó, and J. Varona, "Modeling depth for nonparametric foreground segmentation using rgbd devices," *Pattern Recognition Letters*, vol. 96, pp. 76–85, 2017. 1.1, 1.2, 3, 4.2.2

[17] H. Liang, J. Yuan, and D. Thalmann, "3d fingertip and palm tracking in depth image sequences," in *Proceedings of the 20th ACM international conference on Multimedia*. ACM, 2012, pp. 785–788. 1.1

[18] R. Tara, P. Santosa, and T. Adji, "Hand segmentation from depth image using anthropometric approach in natural interface development," *Int. J. Sci. Eng. Res*, vol. 3, no. 5, pp. 1–4, 2012. 1.1

[19] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. Ieee, 2011, pp. 1297–1304. 1.1, 1.3, 3.3, 3.3, 7

[20] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei *et al.*, "Accurate, robust, and flexible real-time hand tracking," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 3633–3642. 1.1, 1.3, 5.1

[21] B. Kang, K.-H. Tan, N. Jiang, H.-S. Tai, D. Treffer, and T. Nguyen, "Hand segmentation for hand-object interaction from depth map," in *Signal and Information Processing (GlobalSIP), 2017 IEEE Global Conference on*. IEEE, 2017, pp. 259–263. 1.1, 1.2, 1.3

[22] D. Tang, H. J. Chang, and A. Tejani, "Latent Regression Forest: Structured Estimation of 3D Articulated Hand Posture," Imperial College London, London, UK, Tech. Rep., 2014. 1.2

[23] G. Rogez, M. Khademi, J. S. Supančič, J. M. Montiel, and D. Ramanan, "3D hand pose detection in egocentric RGB-D images," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015. 1.2

[24] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001. 3, 3.4

[25] S. B. Gokturk, H. Yalcin, and C. Bamji, "A time-of-flight depth sensor - System description, issues and solutions," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2004-January, no. January, 2004. 3.1

[26] V. Lepetit, P. Lagger, and P. Fua, "Randomized trees for real-time keypoint recognition," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2. IEEE, 2005, pp. 775–781. 3.3

[27] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," California univ San Diego la Jolla dept of computer science and engineering, Tech. Rep., 2002. 3.3

[28] A. Criminisi, "Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning," *Foundations and Trends® in Computer Graphics and Vision*, vol. 7, no. 2-3, pp. 81–227, 2012. 3.4

[29] Y. Amit and D. Geman, "Shape Quantization and Recognition with Randomized Trees," *Neural Computation*, vol. 9, no. 7, pp. 1545–1588, 1997. 3.5.2, 6.3.1

[30] D. M. Pedro F. Felzenszwalb, Ross B. Girshick and D. Ramanan, "Object detection with discriminatively trained part-based models," *Computer*, vol. 47, no. 2, pp. 6–7, 2008. 4.4

[31] A. Kumar, A. Kaur, and M. Kumar, "Face detection techniques: a review," *Artificial Intelligence Review*, 2018. [Online]. Available: https://doi.org/10.1007/s10462-018-9650-2 4.4

[32] M. Patacchiola and A. Cangelosi, "Head pose estimation in the wild using Convolutional Neural Networks and adaptive gradient methods," *Pattern Recognition*, vol. 71, pp. 132–143, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2017.06.009 4.4

[33] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, vol. 2. IEEE, 2000, pp. 142–149. 6.4.1

[34] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image and Vision Computing*, vol. 21, no. 1, pp. 99–110, 2003. 6.4.1

[35] C. Tan, F. Sun, and W. Zhang, "Deep Transfer Learning for EEG-Based Brain Computer Interface," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2018-April, pp. 916–920, 2018. 6.4.1