CLASSIFACIÓ DE LA POSICIÓ DEL CAP PER SEGUIMENT DE CARACTERISTIQUES EN TEMPS REAL

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per
Miguel Reyes Estany
i dirigit per
Xavier Baró, Sergio Escalera i Petia Radeva
Bellaterra, 22 de Juny de 2010

Universitat Autònoma de Barcelona

etse
Escola Tècnica Superior d'Enginyeria

El sotasignat, Xavier Baró Solé

Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en Miguel Reyes Estany

I per tal que consti firma la present.

Signat: Xavier Baró Solé

Bellaterra, 22 de Juny de 2010

Índex

1. INTRODUCCIO

L'anàlisi facial humà, degut a la seva àmplia autentificació biomètrica, és un camp molt actiu en la recerca de l'àrea de visió per computador. L'estimació de la posició del cap, en molts casos, és una component essencial per aplicacions d'interacció entre usuari-ordinador. Aquesta informació pot ser molt útil canalitzant-la a altres camps, com pot ser la comunicació humà-maquina a través d'una interfície molt natural per l'usuari, o altres casos com l'anàlisi del comportament humà.

1.1 MOTIVACIO

Durant els últims anys s'han desenvolupat diferents tècniques i mètodes que embarquen aquesta àrea. El problema és que les solucions donades només treballen sobre subjectes determinats en contexts específics. Això redueix la utilitat de la informació extreta i obre el repte de trobar mètodes més genèrics capaços d'extreure tota la informació sobre la pose facial.

A més, endinsar-me en un camp en plena expansió, en un món on dia rere dia s'intenta trobar una comunicació usuari-màquina més natural per l'home, i, sobretot, treballar sobre un camp on encara no s'han descobert mètodes universals ha estat una de les principals motivacions per l'elaboració d'aquest treball.

1.2 OBJECTIU

Aquest projecte és una introducció al camp de la detecció i classificació de les característiques facials, amb la intenció d'abordar el problema de detecció de la posició mitjançant un sistema detector-classificador.

Al llarg del treball, analitzarem els diferents algoritmes que més s'utilitzen tant en la detecció com classificació de característiques facials. En alguns casos caldrà certa formalització matemàtica per a comparar els diferents resultats amb les diferents tècniques.

Aquest sistema s'haurà d'enfrontar a les diferents condicions ambientals i contextuals, com per exemple, la variació de la il·luminació o la qualitat de la imatge entre d'altres, amb l'objectiu d'arribar a ser un robust sistema reconeixedor-classificador de posició.

A l 'hora d'aplicar les diferents tècniques suposarem que el nostre sistema s'aplica sobre subjectes amb certa continuïtat espaciotemporal, el que comportarà algunes restriccions i certs graus de llibertat al sistema.

1.3 ESTAT DE L'ART

El camp del reconeixement facial i pose és un dels problemes més tractats i estudiats per la visió per computador.
La posició d'un objecte pot ser descrita per mitjà d'una transformació de rotació i translació a partir d'un punt de referència per l'observador. Això comporta que determinar la posició d'un objecte sobre una imatge és un exercici complex, sobre el qual, existeixen principalment tres metodologies a l'hora de tractar el problema:

- **Models basats en l'anàlisi estadístic i geomètric:** són models on es coneix la forma o aparença de l'objecte del qual volem identificar la seva posició a l'espai. És a dir, que la seva projecció a l'espai és una funció ben coneguda. Una vegada hem captat la imatge a través del sistema sensor, hem de trobar un conjunt de punts mapejats sobre la imatge que indueixin a un objecte conegut. Aquesta inducció es fa a través de successives iteracions seguint la lògica de la forma i aparença de la imatge que es vol determinar. El mapejat de punts es realitza de manera estratègica en punts clau de la imatge, on aquesta ens proporciona informació caracteritzant i no ambigua, tals com cantonades, polígons parcialment regulars, o fronteres. Un dels exemples més comuns són el Active Shape Models[1] o el Active Appereance Models[2].

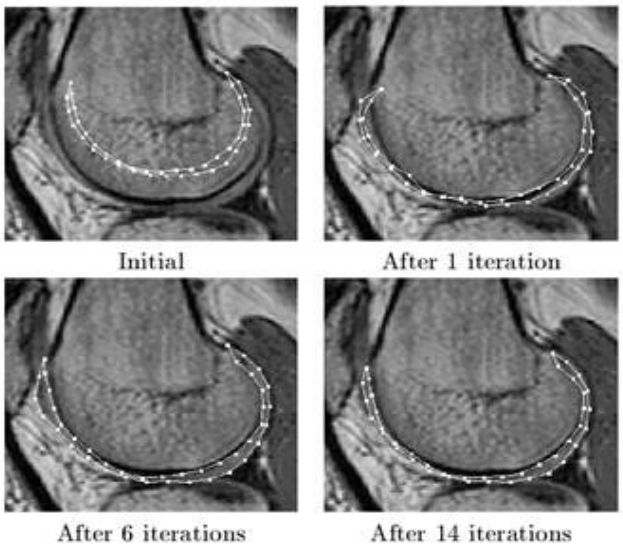


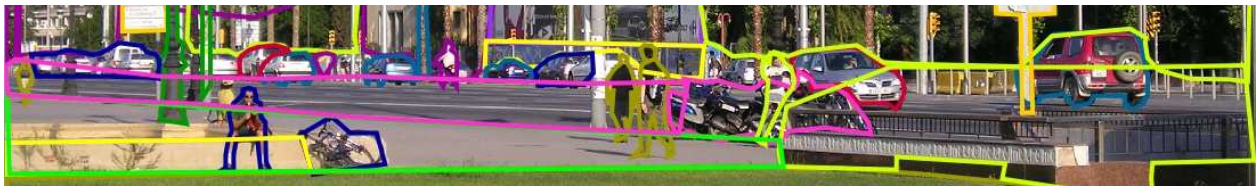
Figura 1.1

- **Models basats en aprenentatge:** són models on el reconeixement de la posició de l'objecte és totalment artificial, donant-li al sistema un tractament empíric del coneixement de les dades. Aquest model conté un alt volum d'exemples de la posició de l'objecte durant la fase d'aprenentatge. La determinació de la pose d'un objecte, vindrà donada per l'objecte après que més s'acosti a l'objecte a determinar[3], segons

uns paràmetres i funcions de semblança determinats pèl mètode.

- **Mètodes basats en optimització:** aquests mètodes donen una visió robusta al problema, especialment quan les imatges no han estat alineades. La posició de l'objecte inicial es representa mitjançant un mapejat aleatori de punts sobre la imatge. La diferència o error es controla a través d'una funció d'aptitud. El mapejat dels punts inicials s'aniran alterant de forma aleatòria però controlada dins d'un cert número d'iteracions o fins que l'error de la posició de l'objecte estigui dins d'un límit acceptat com a correcte. El principal problema d'aquest mètode es que en certs contexts no es pot implementar en temps real, ja que el temps de càlcul no està controlat. Un exemple és l'ús dels algorismes genètics[4].

Fins ara, només s'ha esmenat el problema de tractar la posició d'un objecte, però tal i com es veurà al llarg del treball, aquest problema inclou tot un sistema de visió per computador on hi intervenen elements d'adquisició de la imatge, extracció de característiques, preparació i segmentació de las imatges, i processat de la informació per a la classificació.



1.4 PLANIFICACIÓ I COST:

Descrivim quins recursos i temps de dedicació s'ha necessitat per dur a terme aquest projecte. S'ha fet una divisió de les tasques principals per fer un millor seguiment del projecte.

Id	Descripció	Inici	Finalització	Duració
1	Mètodes bàsics extracció d'informació sobre imatges	23/03/09	28/04/09	26
2	Mètodes bàsics de reconeixement objectes	29/04/09	29/05/09	23
3	Familiarització i realització d'aplicacions amb OpenCv	25/03/09	28/05/09	45
4	Creació de models d'aprenentatge basats en posició del cap	01/06/09	30/07/09	40

5	Aplicació de les diferents tècniques de reconeixement	03/08/09	17/09/09	30
6	Aplicacions de detecció i reconeixement de posició del cap en temps real	02/11/09	30/11/09	18
7	Disseny i implementació software detector de pose	02/06/09	30/11/09	120
8	Introducció mètodes estadístics basats en forma	14/12/09	15/01/10	25
9	Introducció mètodes estadístics basats en aparença	18/01/10	12/02/10	20
10	Utilització software classificador de pose basat en AAM	15/02/10	22/03/10	25
11	preparació de paper per a Workshop on Analysis and Modeling of Faces and Gestures CVPR10	08/02/10	19/03/10	30
12	Elaboració de la documentació	07/12/09	01/06/10	125

Observem la projecció al llarg del temps d'aquestes tasques:

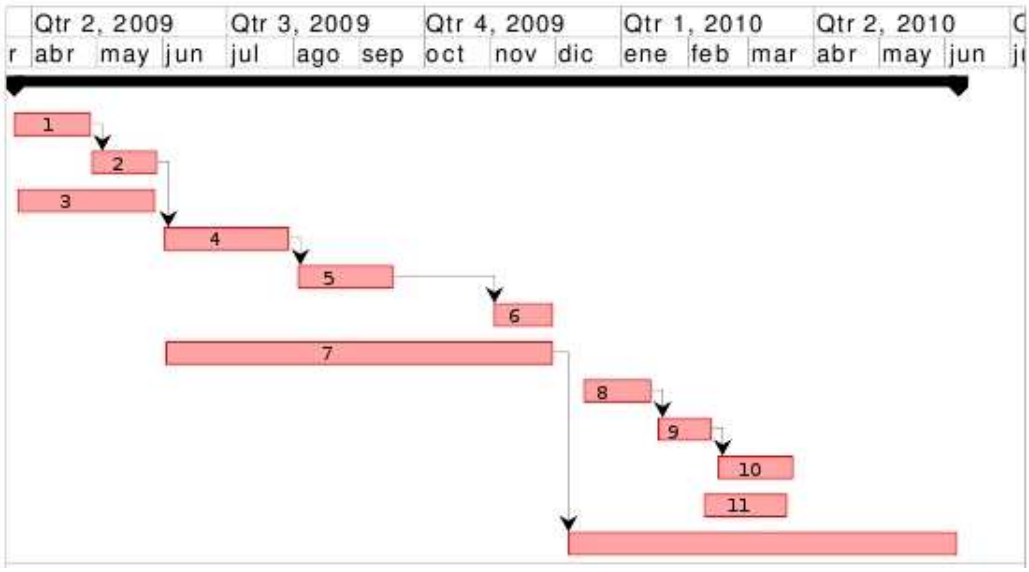


Figura 1.2

Analitzant la planificació de les tasques s'observa que el treball s'ha realitzat principalment en tres fases.

- **Fase d'iniciació:** Una primera fase d'iniciació en la visió per computador, englobada per les tres primeres tasques. Aquesta fase comporta el desenvolupament de diferents proves sobre l'adquisició i processament de les imatges, on es van implementar diferents aplicacions en OpenCV per familiaritzar-se amb el llenguatge i la visió per computador. Aquesta iniciació va ser essencial i necessària ja que encara no es tenien prou coneixements sobre el camp on s'anava a treballar. Aquesta fase que hem anomenat d'iniciació consta de 46 dies.

-**Tractament del problema:** una vegada es va obtenir certa fluïdesa, es va a passar a tractar el problema específic de la posició del cap amb eines i mecanismes propis de visió per computador. És un període d'anàlisi del problema i examinació dels diferents mètodes utilitzats per altres autors. Juntament s'inicia la tasca de "Disseny i implementació de software detector de pose", on sempre la mantindrem paral·lelament, ja que aquest tasca ens servirà com a banc de proves dels mètodes a examinar i per l'anàlisi del nostre. Observem que la tasca "Creació de models d'aprenentatge basats en posició del cap" ocupa un considerable espai de temps, tal i com veure en els següents capítols, és una tasca molt important alhora d'obtenir resultats. Finalment aquesta segona fase acaba amb "Aplicacions de detecció i reconeixement de posició del cap en temps real" on donem per finalitzat un mètode per a la classificació de la posició del cap. Tota aquest fase produeix un cost de 120 dies.

-Sorgiment d'un mètode alternatiu: una vegada analitzats els resultats obtinguts a la segona fase, es va proposar un altre mètode al problema del tractament de la posició del cap. Aquesta fase consta d'un període més curt, ja que els plantejaments del problema a tractar, tals com context, objectius, etc. van ser resolts a la fase anterior. La durada total és de 70 dies.

Com a tasca final, ens queda "Elaboració de la documentació " del projecte.

Es calcula que el projecte ha produït un cost de 270 dies de treball. Com a mitjana d'hores de treball dedicades cada un d'aquest dies suposarem 2,5 hores, per tant el projecte consta de 675 hores. Davant aquesta planificació el projecte exigeix el següent cost pressupostari:

	Quantitat	Preu (amb IVA)	Preu Final
Llicència Microsoft Visual Studio 2005	1	542,00 €	542,00 €
Llicència Matlab Student Version 2008	1	51,20 €	51,20 €
Equip informàtic	1	600,00 €	600,00 €
Hores treballades	675	13€/h	8.775,00 €
		Total	9.968,00 €

1.5 Estructura de la memòria

A continuació es comenta breument l'estructuració d'aquesta memòria:

- Anàlisi: En aquest apartat s'estudien les diferents tècniques d'adquisició i processament de la informació a través de les imatges. S'enumeraran els diferents mètodes utilitzats per diferents autors, i entrarem en més detall en aquells que hem utilitzat i testejat. Es contrastaran diferents mètodes per al mateix fi amb la intenció d'analitzar avantatges i desavantatges. Sempre que calgui, l'anàlisi d'una tècnica vindrà donat de certa formalització matemàtica.

- Disseny: Dins d'aquest capítol posarem en pràctica els diferents mètodes analitzats a l'apartat anterior, enumerant els diferents problemes sorgits i mostrant els resultats obtinguts. Es comentaran els algoritmes utilitzats i les modificacions adients que hagin sorgit per a produir els diferents resultats. No s'entrarà en detall en la implementació però sí es comentaran certes primitives d'OpenCV que ens han semblat interessants.

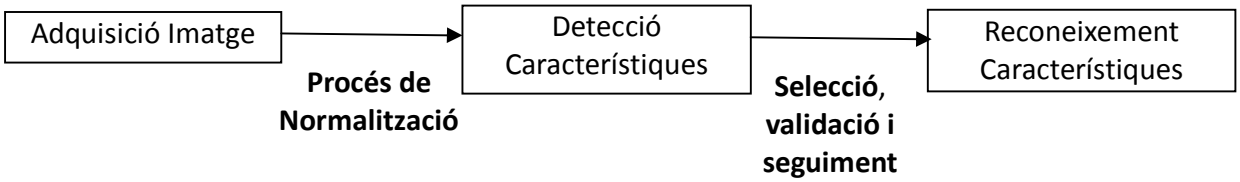
- Resultats: Es mostraran les diferents sortides als diferents mètodes implementats, utilitzant els diagrames més adients per l'anàlisi dels resultats. S'explicaran tant els resultats donats pels tests com les condicions en les quals s'han processat.

- Conclusió: En aquest últim capítol es fa la reflexió final. Explicarem el grau d'encert o fracàs al treball realitzat mitjançant l'anàlisi de resultats així com futures línies de continuació.

1. ANÀLISI DE TECNIQUES I METODOLOGIES DE DETECCIÓ DE CARACETERISTIQUES

A continuació realitzarem un exercici d'anàlisi i observació dels diferents mètodes de reconeixement i seguiment de característiques facials en el camp de la visió per computador.

Abans d'entrar a cada un dels mètodes mostrem el sistema que volem crear per a determinar la posició del cap sobre una imatge:



L'Adquisició de la imatge serà l'element inicial del sistema, també anomenat element sensor, encarregat d'extreure la imatge d'un medi físic. Aquesta subtracció ha de ser enviada de forma llegible per els altres elements a través d'una codificació comprensible per els altres elements del sistema. Aquesta part física del sistema serà obviada en la documentació.

Una vegada hem obtingut la imatge del medi, ja sigui a través d'un vídeo o amb una webcam, s'executa un procés de normalització. Aquest procés no és cap mètode de detecció, sinó una preparació de les dades per a que els mètodes de detecció funcionin de manera òptima. Com veurem, aquest procés és determinant a l'hora de l'obtenció de les característiques, i per tant, molt determinant en els resultats finals.

La part central del sistema, detecció de característiques, és on trobem la varietat de mètodes i tècniques de detecció de regions d'interès (ROI), les quals analitzarem. L'objectiu es fer un anàlisi exhaustiu de les sortides produïdes per aquest bloc, amb la intenció de seleccionar aquell mètode més adient pel nostre objectiu, el reconeixement de la posició del cap en temps real.

De sortides possibles del bloc de "Detecció de característiques" podem obtenir vàries, encara que totes no poden ser vàlides. Llavors en el nostre cas entrarà en joc certa lògica espaciotemporal capaç tant de descartar possibles casos de falsos positius com verificar les deteccions correctes. En aquesta part del sistema, ens centrarem en comentar com i per què hem utilitzat la lògica espaciotemporal. Podríem dir que aquest procediment actua com a filtre.
La part final del sistema serà l'encarregada de donar-li coherència a tota la informació extreta al sistema. Aquest bloc té com a finalitat proporcionar una classificació discreta al conjunt de

característiques, extretes pel procés anterior. Com a sortida final del sistema obtindrem la posició del cap respecte a un punt d'origen.

2.1 PROCÉS DE NORMALITZACIÓ

Aquest processament previ que es realitza sobre les imatges pretén lliurar un alt grau d'uniformitat a les imatges. Aquest procediment comporta certa independència a les propietats de la imatge. Les principals propietats en les que treballarem són:

- Brillantor
- Contrast
- Nivell de gris
- Mida horitzontal i vertical

La normalització d'imatges ha de ser capaç d'eliminar soroll i ressaltar els aspectes més importants. Aquesta normalització també ha de corregir efectes mediambientals, per exemple la llum, tals que poden alterar la sortida del detector de característiques[5].

2.1.1 FILTRES TRANSFORMADORS

Aquesta transformació treballa amb l' histograma de la imatge monocroma. Pretén obtenir una distribució uniforme de l', és a dir, donar el mateix nombre de píxels a cada nivell de gris de la imatge.

Imatge Original	Imatge Equalitzada
	

Figura 2.1

Observem que amb aquesta transformació augmenta el contrast de la imatge i la brillantor és uniforme.

Per a millorar la visibilitat dels objectes de la imatge ens interessa eliminar soroll. Amb aquest fi utilitzarem una matriu de convolució. Per aplicar aquesta transformació, definirem la imatge a tractar composada per una col·lecció bidimensional de píxels en coordenades rectangulars. Aquesta matriu serà multiplicada per un altra anomenada "kernel". El kernel utilitzat depèn de l'efecte desitjat.

Mostrant un exemple, si la nostra imatge es representés mitjançant la següent matriu:

35	40	41	45	50
40	40	42	46	52
42	46	50	55	55
48	52	56	58	60
56	60	65	70	75

I la nostra matriu 3x3 **kernel** és:

0	1	0
0	0	0
0	0	0

Aquesta transformació, també anomenada filtratge, examina successivament cada píxel de la

imatge. Per a cada píxel, que anomenarem "píxels inicials", es multiplica el valor d'aquest píxel i el valor dels 8 circumdants pel valor corresponent al kernel. Llavors s'afegeix el resultat, i el píxel inicial es regula amb aquest valor resultat[6].

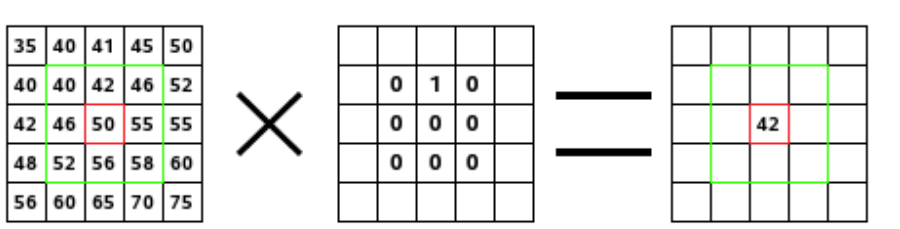


Figura 2.2

Detallant l'acció, el valor del píxel central a la imatge resultant fa les següents operacions:

40*0 + 42*1 + 46*0 + 46*0 + 50*0 + 55*0 + 52*0 + 56*0 + 58*0 = 42

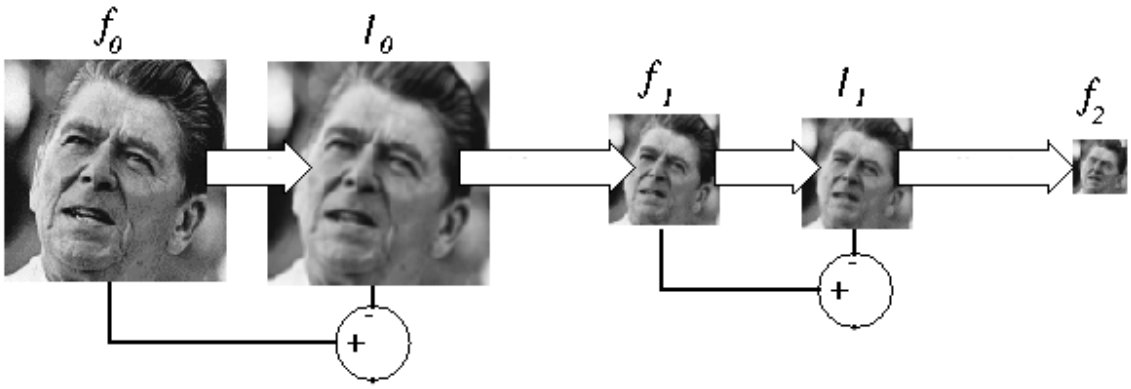
Aquest kernel, o filtre, produirà un efecte de desplaçament vertical sobre una imatge.

Figura 2.3

Per al nostre projecte utilitzarem un filtre de suavitzat, de mida 5x5, amb l'objectiu d'eliminar soroll. La mascara del **kernel** és la següent:

1	1	1	1	1
1	3	3	3	1
1	3	9	3	1
1	3	3	3	1
1	1	1	1	1

Amb aquest filtre suavitzem l'escena, és a dir, s'eliminen els petits canvis bruscos de color que existeixin. Aquesta transformació es realitza repetidament sobre una imatge a diferents escales, i de cada una estimem un promig, obtenint una transformació final. D'aquest procés final anomenem **Gaussian Pyramid**, on existeixen dues variants, **down** quan les imatges s'escalen a mida inferior i ***up*** quan les imatges s'escalen a mida superior de l'original. Per a l'eliminació del soroll és apropiat el ***down***, per tant utilitzarem aquest.



Arriba el punt d'examinar els mètodes de detecció de característiques, o objectes, que existeixen en el camp de la visió per computador. En aquest apartat no ens centrarem en un objecte en concret, sinó analitzarem la tècnica per a detectar qualsevol objecte sobre una imatge.

Adquisició Imatge → **Procés de Normalització** → Detecció Característiques → **Selecció, validació i seguiment** → Reconeixement Característiques

2.2 DETECTOR D'OBJECTES VIOLA JONES

És un dels mètodes que més bons resultats dona per aplicacions en temps real, degut al seu escàs temps de càlcul, amb una taxa de falsos positius baixa. Va ser desenvolupat a l'any 2001 per Paul Viola i Michael Jones[7], a través de la fundació investigadora Mitsubishi Elèctrics. Aquest detector es basa fonamentalment en tres conceptes, els quals passarem a comentar:

- Haar like features
- Integral Image
- Classificador en cascada

2.2.1 HAAR LIKE FEATURES

Durant finals de la dècada dels 80, quan s'estava gestant la tècnica ara coneguda com visió per computador, a l'hora de detectar qualsevol característica sobre una imatge, utilitzem la seva component d'intensitat. Es treballava sobre el canal d'il·luminància en imatges monocromàtiques, o elaborant una cerca sobre certs rangs de valors RGB en imatges a color. Ja a la dècada dels 90, es va pensar un mètode alternatiu a l'hora d'examinar la imatge. Es va proposar fer subdivisions de la imatge, englobades en regions, les quals es categoritzaven per la suma del valor dels píxels que conformaven l'àrea de la regió. Aquesta idea va motivar a crear el mètode de ***Viola-Jones***. La principal aportació del seu mètode era la forma de fer les subdivisions de la imatge. Quan examinen la imatge s'utilitza el que anomenem les "***Haar-like features***", que són rectangles de mida arbitrària, que col·locats sobre la imatge, tornen a subdividir l'espai de la imatge en regions.
Observem els tipus de "***Haar-like features***" existents:

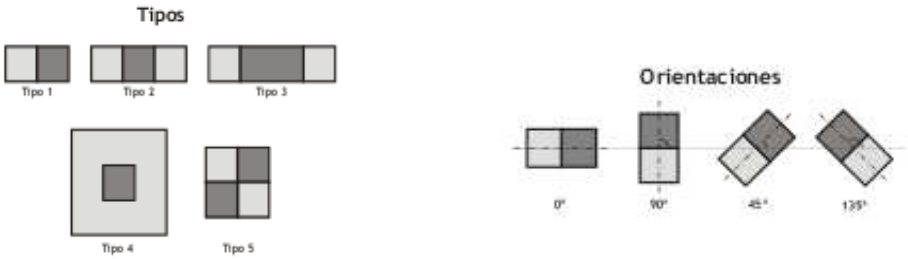


Figura 2.5

El valor de cada regió ve donat per la diferencia de la suma de tots els píxels d'un rectangle. Obtenint aquests valors, guardem informació molt caracteritzant, tals com fronteres o canvis de textura. Un altre avantatge del mètode, es que treballar en un sistema basat en característiques opera molt més ràpid que un sistema basat en píxel.

Freqüentment els objectes o patrons que estem buscant sobre una imatge poden estar rotats o escalats o qualsevol posició de la imatge. Això comporta una utilització exhaustiva de les "***Haar-like features***" sobre la imatge, que a priori produiria un elevat cost computacional. Una de les grans aportacions de ***Viola-Jones*** és el seu mètode per a calcular el valor de cada regió rectangular, es aquí quan entra el concepte de "***integral image***".

2.2.2 INTEGRAL IMAGE

La "***integral image***" és valor calculat sobre els píxels d'un rectangle utilitzant els "***Haar-like feature***". El valor de la "***integral image***" en la posició *x,y* conté la suma dels píxels de esquerra a dreta i de dalt a baix que conforma el rectangle.

$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

Observem el mètode del càlcul d'una característica mitjançant un exemple. Suposem que apliquem la següent "***Haar-like feature***" sobre una regió de la imatge:

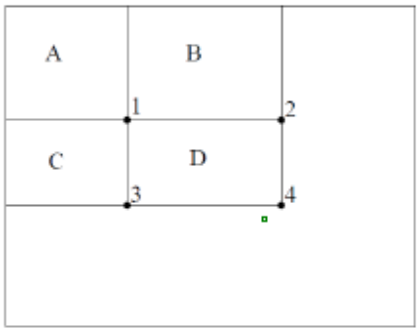


Figura 2.6

Analitzant la imatge, podem dir que el valor de la "***Integral Image***" en el punt 1 es la suma dels píxels del rectangle A. El valor de "***integral image***" en el punt 2 es la suma dels píxels de A i B, en el punt 3 A i C, i finalment, en el punt 4 esta format per A, B,C i D. Llavors podem calcular el valor de la "***Haar-like feature***" amb només 4 referències a un vector. Es aquest fet el que fa que ***Viola-Jones*** sigui un mètode utilitzat en aplicacions a temps real.

2.2.3 CLASSIFICADOR CASCADA

El plantejament ideal seria que apliquéssim les característiques ***Haar*** sobre la totalitat de la imatge, en diferents posicions, i obtenir els millors resultats. Aquest fet comportaria un alt cost computacional tant d'espai com de temps. Davant aquest problema es va desenvolupar el classificador ***Adaboost.*** El mètode ***Viola-Jones*** utilitza la variant ***Gentle Adaboost***. Un classificador ***Adaboost*** esta format per un conjunt de funcions classificació combinades associades a un pes. Aquest pes es el grau de fiabilitat de la funció classificadora. Diguem que una funció classificadora amb un alt pes filtrarà més falsos positius i que una amb un valor menor. Aquest pes es associat en la fase d'entrenament del classificador, normalment a través d'un procés d'obtenció i ordenació de la informació invariant i característica, aplicant

procediments com ***PCA***.
S'anomena classificador en cascada pel motiu de que es van aplicant funcions classificadores de menys a més pes, el que comporta, certa inducció a trobar certs positius i rebutjant possibles negatius en funcions classificadores prèvies. Aquest fet, comporta una disminució en el temps computacional, i per tant un altre cop, apropament a les aplicacions en temps real.

2.2.4 TEST VIOLA-JONES

Detecció cara frontal base de dades [8]

Nombre de Cares	507
Certs positius	71,94% 364
Falsos positius	11,83% 60
Falsos negatius	16,23% 83

Figura 2.7

Detecció de característiques facials, de la mateix base de dades amb 137 cares

	Ulls	Nas	Boca
Detecció correcta	87,30%	78,30%	82,50%
Detecció incorrecta	9,60%	20,40%	11,20%
No detecció	1,10%	1.30%	6,30%

2.3 TEMPLATE MATCHING

Aquesta tècnica va ser històricament, la primera emprada a l'hora de trobar un objecte sobre una imatge. El seu funcionament és molt simple i instintiu. Es basa en examinar diferents parts de la imatge on aplicarem un patró, o varis, de l'objecte el qual estem buscant. On hi existeixi una taxa d'encert suficientment fiable, confiem a trobar l'objecte. La metodologia es força senzilla però s'han elaborat diferents metodologies, més complexes i eficients, que proporcionen més robustesa i fiabilitat al procés de **matching**.

2.3.1 TÈCNIQUES DE TEMPLATE MATCHING

El patró es projecta exhaustivament , sobre la imatge que volem examinar, des de el marc superior esquerra fins al inferior dreta, alhora es guarden les regions on més encert s'ha aconseguit. Aquest patró haurà de ser de mida inferior o igual a l'objecte que busquem.

- Mètodes que proporcionen el grau de **matching** *Anomenem T al patró, I a la imatge i R a la raó d'encert***:**

1. **Correlació :** Multiplicació del patró per a cada regió de la imatge. Això produeix que un patró molt similar a la regió de la imatge serà proper a la unitat, mentre que una diferencia del patró i la imatge tendeix a zero.

$$\boldsymbol{R_{ccorr}(x,y) = \sum_{x',y'}[T(x',y').I(x+x',y+y')]^2}$$

2. **Mètode de la diferència dels quadrats:** En aquest cas un **matching** perfecte tendeix a zero mentre que una gran entre patró i imatge comporta la unitat.

$$\boldsymbol{R_{qdiff}(x,y) = \sum_{x',y'}[T(x',y') - I(x+x',y+y')]^2}$$

3. **Coeficient de correlació:** en aquest cas el grau d'encert es donat en relació una mitjana. Ara un bon **matching** tendirà a **+1** mentre que un dolent cap a **-1**

$$R_{ccoef}(x,y) = \sum_{x',y'}[T'(x',y') \cdot I'(x+x',y+y')]^2$$

$$T'(x',y') = T(x',y') - \frac{1}{(w.h)\sum_{x',y'} T(x',y')}$$

$$I'(x+x',y+y') = I(x+x',y+y') - \frac{1}{(w.h)\sum_{x',y'} I(x+x',y+y')}$$

2.3.2 APLICACIONS I PROVES

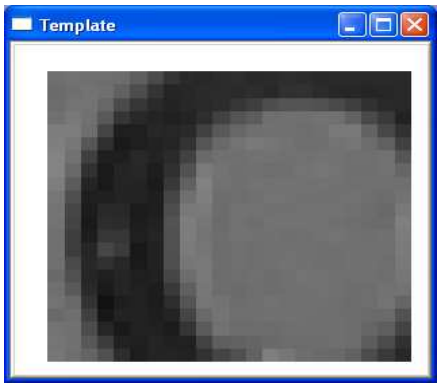
Imatge Original	Patró
	

Figura 2.8

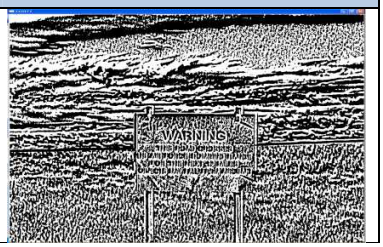
CORRELACIÓ	DIFERÈNCIA QUADRATS	COEFICIENT CORRELACIÓ
		

Figura 2.9

A primera vista ja podríem dir que els resultats no son gens bons. En cap de les tres imatges resultants de cada mètode, som capaç de distingir un patró. Això és degut a que sobre les imatges existeix soroll per part del medi, en aquest cas diferencies de llum que fan no existeixi una correspondència prou exacta entre la imatge i el patró. Per corregir aquest efecte

al canal de la il·luminància exercirem un procés de normalització a les imatges.

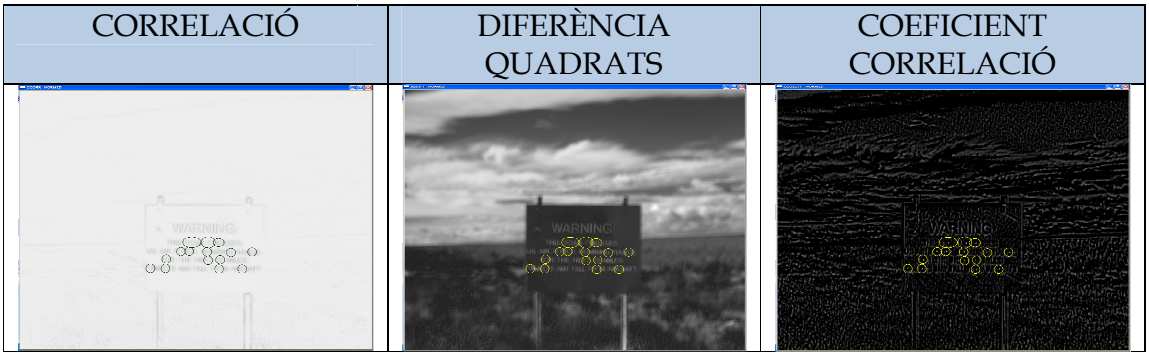


Figura 2.10

Els resultats milloren i amb cert coneixement de la situació dels objectes podríem identificar algun patró. Tot així podem dir que no es mètode prou robust sobre imatges en entorns no controlats, on hi ha o pot haver molt soroll. El **matching template** podem dir que és un problema clàssic de cerca de màxims o mínims globals, depenent del mètode emprat.
Com hem dit, aquest tècnica no es prou eficient en entorns no controlats i inesperats, observem ara un experiment en entorns totalment controlats. En quant a controlats ens referim a que la il·luminació és mante en nivells constants, els patrons contenen la mida exacta i no es troben rotats.

Aplicació del **matching template** a la industria, utilitzant la correlació normalitzada:

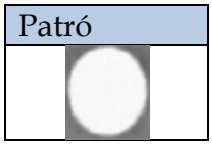


Figura 2.11

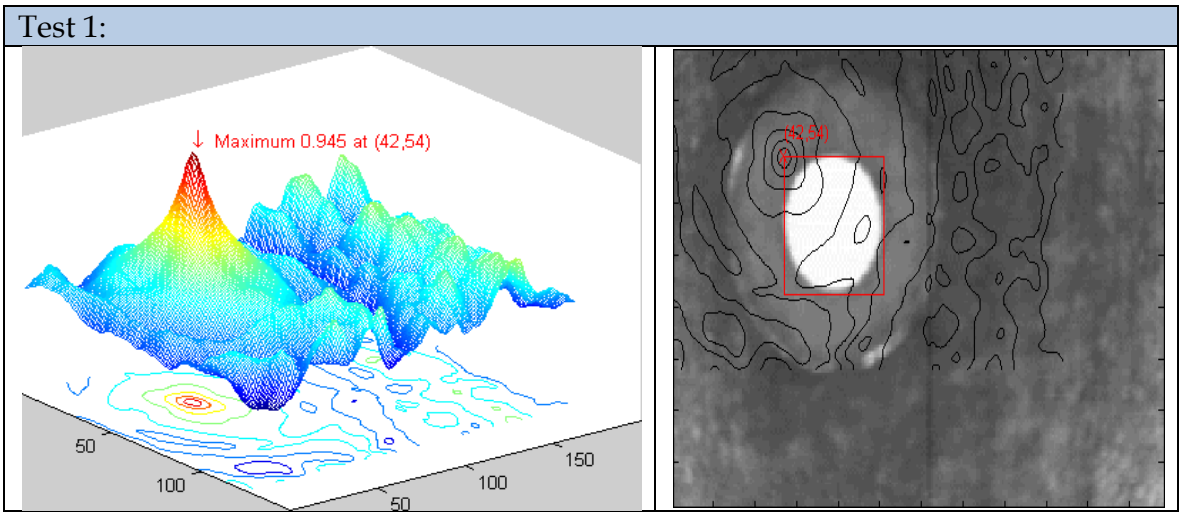


Figura 2.12

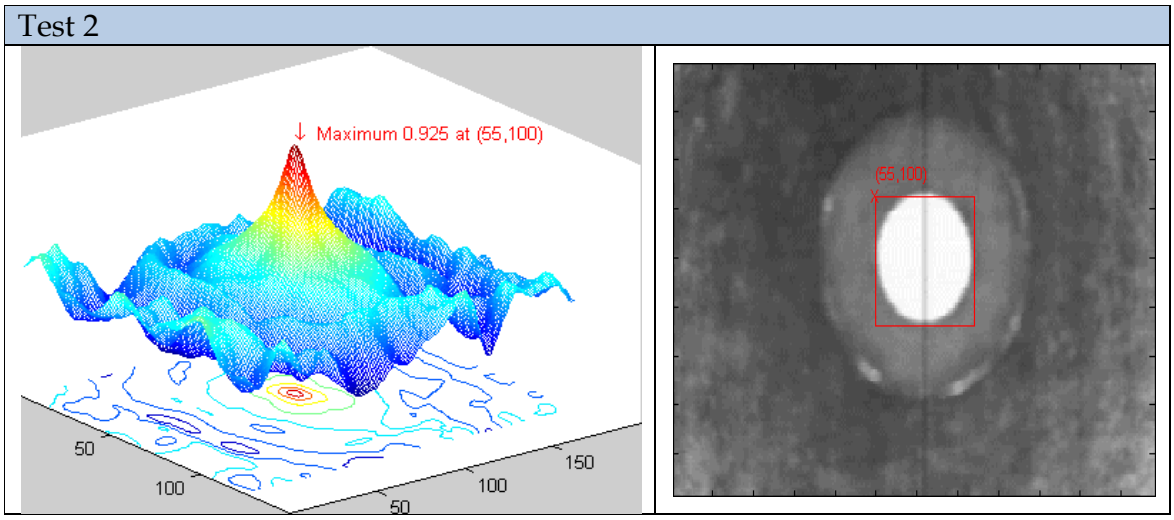


Figura 2.13

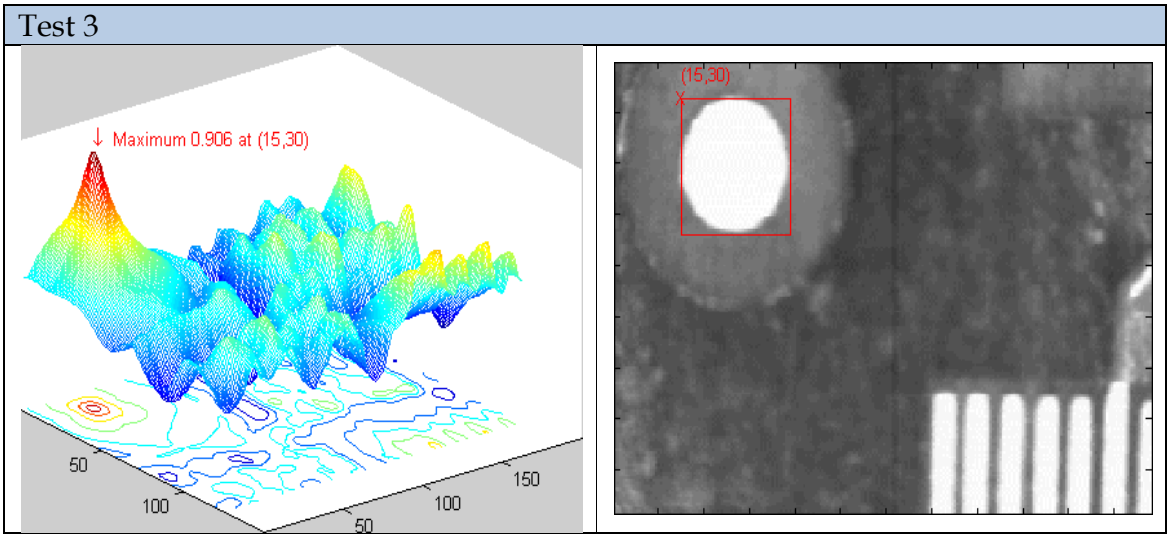


Figura 2.14

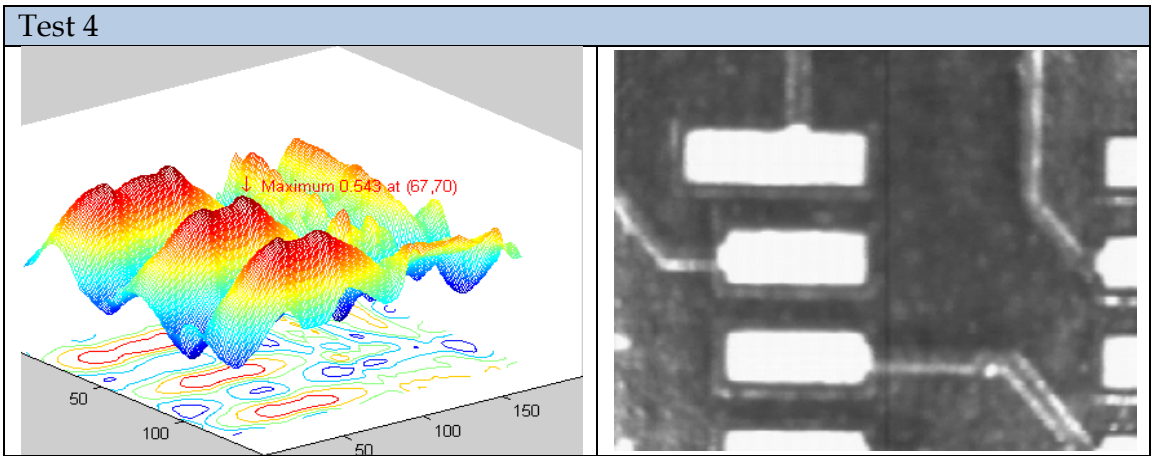


Figura 2.15

Amb les gràfiques es pretén constatar que el **template matching**, es redueix a un problema de cerca de màxims i mínims globals, treballant sobre la informació proporcionada per les imatges. Amb aquest últim experiment es veu que els resultats milloren en entorns controlats. Per tant, aquest mètode serà descartat a l'hora d'afrontar el problema de la localització de característiques facials, ja que l'aplicació vol ser prou robusta i treballar en entorns amb alta probabilitat de soroll.

3. ANALISI DE TECNIQUES I METODOLOGIES DE SEGUIIMENT DE CARACTERISTIQUES

Fins ara s'ha descrit mètodes per a la detecció de les característiques. Ara entrem en el procés de seguiment o **tracking** d'aquestes característiques. Tal i com es va comentar d'inici, el seguiment de qualsevol objecte sempre mantindrà les premisses de la lògica espaciotemporal.

3.1 LOGICA ESPACIOTEMPORAL

La lògica **espaciotemporal** es basa en:

- **Localitat espacial**: Una vegada detectat l'objecte a l'instant **t**, si obtenim la posició espacial d'aquest a l'instant **t+1,** la seva distancia neta respecte a l'instant **t** no pot superar cert valor llindar. Aquest valor llindar serà relatiu a la resolució de les imatges sobre les que es treballa, i la unitat de càlcul seran píxels.

- **Localitat temporal:** Els objectes no desapareixen de l'escena de forma immediata. Si hem detectat un objecte a l'instant **t**, donem per cert que també existirà a l'instant **t+1.** Si el nostre sistema detector no ha sigut capaç de trobar-ho a l'instant **t+1,** la posició de l'objecte vindrà donada per la premissa anterior, posicionant l'objecte a la posició de l'instant **t.**

El bloc de detecció de característiques comporta una segona fase, que és el seguiment o **tracking** d'aquestes. Aquesta segona fase és fortament depenent de les tècniques de detecció i inclús el sistema pot tornar enrere, si el sistema de seguiment no troba l'objecte. Observem-ho millor amb un diagrama (figura 3.1)

Seguidament entrem en els sistemes de seguiment o **tracking** de característiques. Una vegada
l'objecte ha sigut detectat,aquests sistemes han de ser capaços de conèixer la posició a
l'escena de l'objecte detectat.

Els mètodes analitzats com a sistemes de seguiment són els següents:

- **Sift i Surf**
- **Optical Flow de Lukas Kanade**
- **Mean Shift – Cam Shift**
- **Blob Tracking**

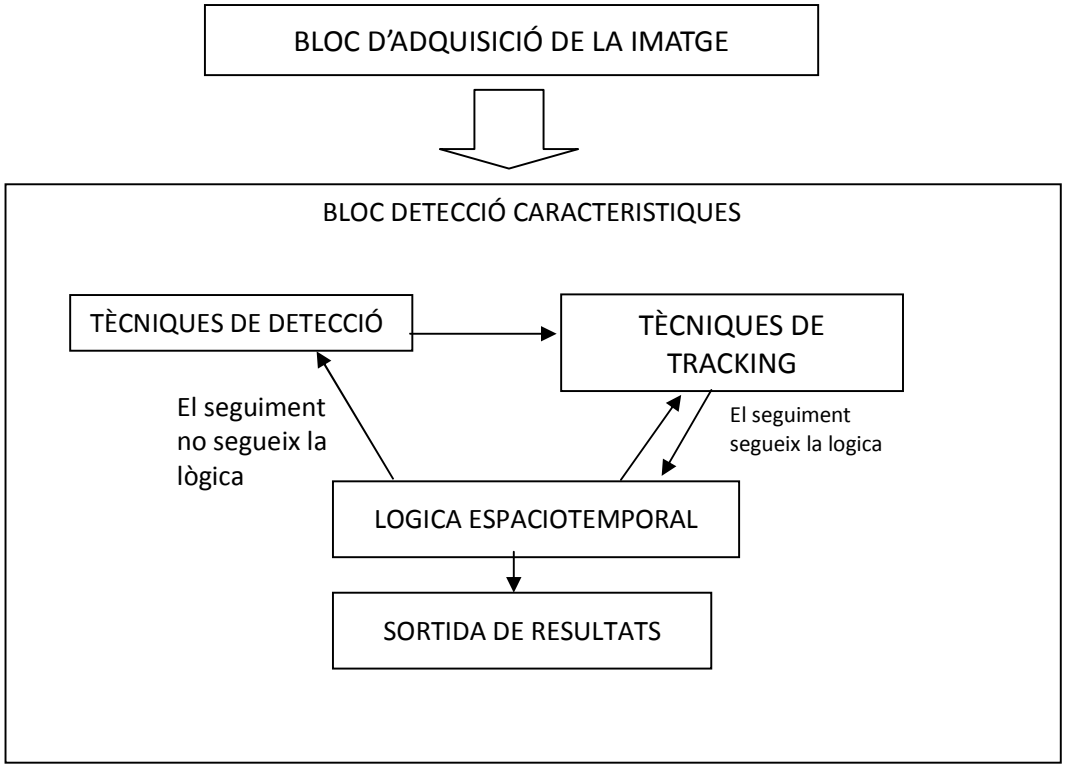


Figura 3.1

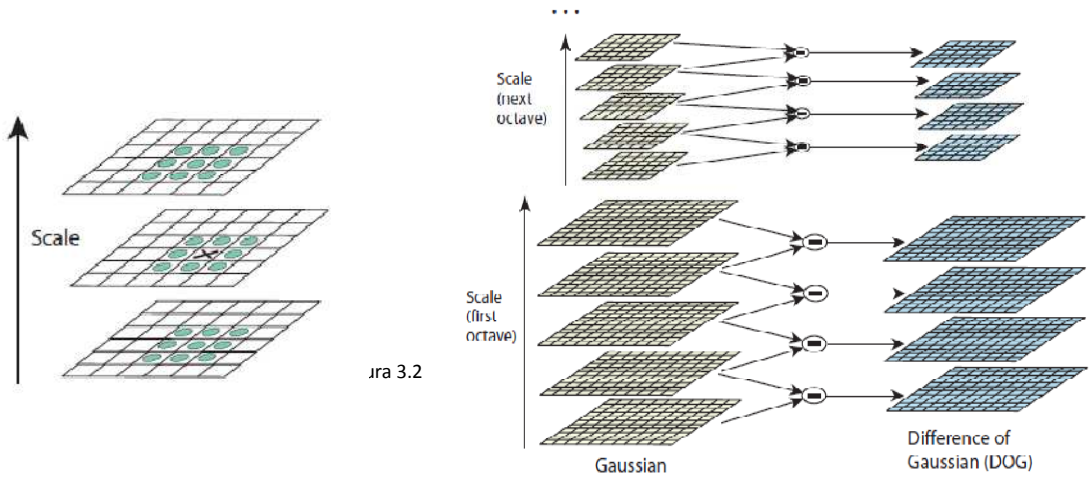
3.2 SIFT (Scale-invariant feature transform)

SIFT és un algorisme capaç de detectar i descriure els trets més característics de la regió d'interès d'una imatge. Una vegada extrets un conjunt de trets característics, té l'habilitat trobar la correspondència d'aquests mateixos sobre la imatge, a la qual se li han aplicat certes transformacions, tals com translació, rotació o escalat. La principal avantatge d'aquesta tècnica és la seva robustesa davant les transformacions sobre la imatge, el soroll, canvis d'il·luminació i inclús, en alguns casos, la oclusió parcial.

Per portar a terme aquesta extracció s'ha d'examinar exhaustivament l a imatge. Si no féssim servir una metodologia prou eficient aquesta extracció ens podria consumir un alt cost computacional, amb aquest motiu analitzarem la tècnica emprada per **Lowe[8]**, la qual es capaç combinar robustesa amb rapidesa.

Fonaments de l'algorisme de **Lowe**:

- **Trobar l'espai-escala**: S'estableixen com a trets característics (o **keypoints)** els màxims i mínims del resultat de la diferència de la funció Gaussiana. Aquestes diferències provenen del resultat d'aplicar un filtre de convolució de suavitzat a una col·lecció d'imatges amb resolució decreixent . Anteriorment es va definir com a piràmide gaussiana. Amb aquesta metodologia els petits detalls desapareixen de l'escena i els que perduren els considerem com a **keypoints.** Aquests trets són invariants a l'escalatge i a la rotació.

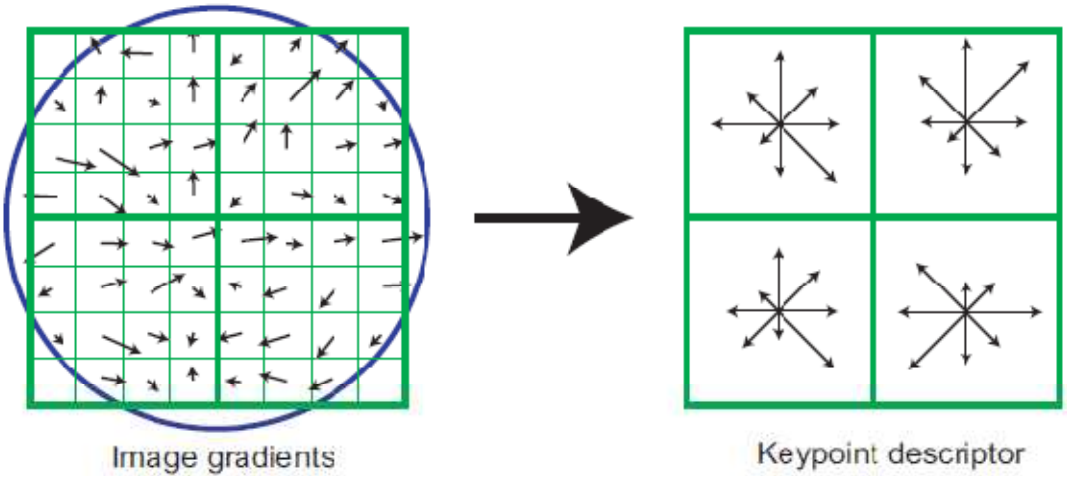


ura 3.2

- **Selecció i indexació dels keypoints:** Davant una nova imatge s'han de localitzar quins trets característics ofereixen més bon **matching**, llavors existeix el problema de com accedir als **keypoints** emmagatzemats i quins d'ells seleccionar. Lowe va emprar l'algorisme anomenat **Best-bin-first,** el qual associa a cada **keypoint** el valor produït per la distancia Euclidiana entre els nous **keypoints** produïts de la nova imatge i els anteriors. Llavors, cada **keypoint** de la imatge nova té associada una cua (estructura de dades) d'ordre ascendent ordenada a partir de la distancia euclidiana, aquesta metodologia és coneguda per **Nearest-Neighbour**, amb els seus respectius **keypoints** de la imatge anterior, o de training. Per reduir l'espai aquestes estructures de dades s'utilitzen valors llindars per a les distàncies euclidianes, amb la intenció d'acotar la longitud de la cua.

- **Assignació de l'orientació:** Per a cada **keypoint** s'assignen un o varies orientacions basant-se en la direcció del gradient de la imatge local. Les futures operacions són executades sobre dades de la imatge que ha sigut transformada d'acord a la orientació, escalat i localització assignat per a cada **keypoint**, així es proporciona invariància a transformacions de rotació.

- **<u>Descriptor del keypoint:</u>** els gradients de la imatge local són calculats a l'escala seleccionada dins la regió al voltant de cada **keypoint**. Aquests són transformats a una representació que permet, per a nivells significatius, distorsió de forma local i canvis d'il·luminació.



Una vegada obtingut un candidat a **keypoint** sobre una nova imatge,entrem en post-procés de selecció i filtratge de punts erronis, ja sigui per un mal contrast o erròniament posicionats. De cada **keypoint** candidat, examinem el voltant d'aquest amb el propòsit d'extreure informació de la posició, escalat i proporció de corbes principals.

Aquesta informació permet rebutjar punts amb baix contrast, i per tant sensibles al soroll, o erròniament localitzats al llarg d'una vora o cantonada. També cal eliminar les vores, ja que els resultats obtinguts de les diferencies de gaussianes (**LoG**) comportarà una resposta ferma al llarg de les vores, també en situacions on molt a prop d'aquestes vores existeix soroll. Per evitar aquest acoblament del soroll a les vores eliminarem aquestes.

<u>Exemple d'extracció de característiques o **keypoints**.</u>

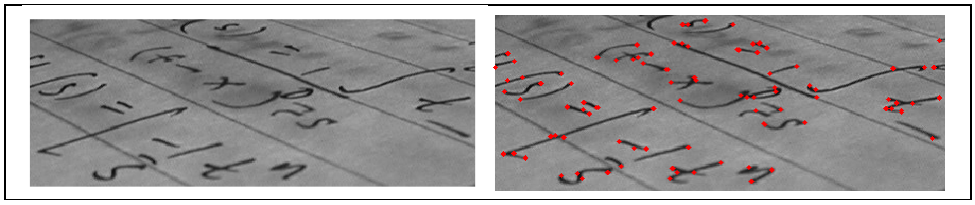


Figura 3.4

Exemple de correspondència de característiques:

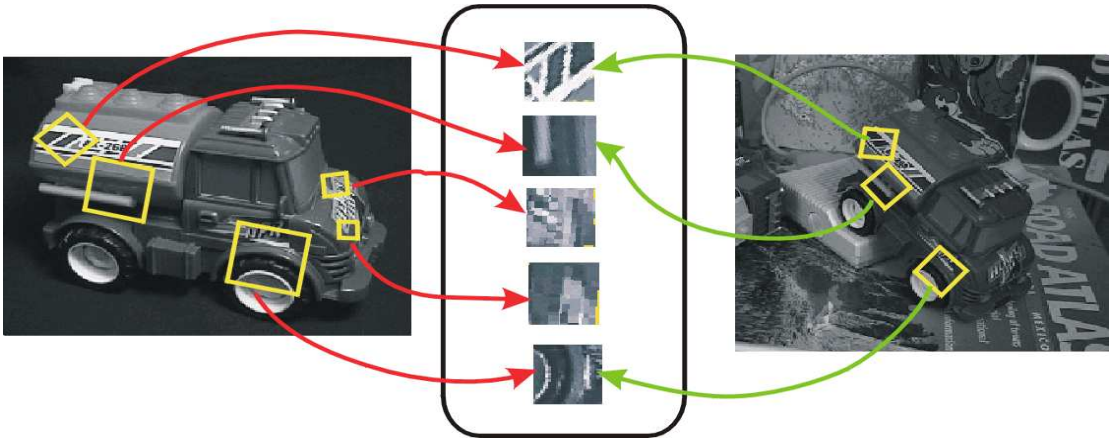


Figura 3.5

3.3 SURF (Speed Up Robust Features)

SURF[10] és una tècnica derivada del SIFT, on la seva principal aportació respecte la tècnica anterior és la seva velocitat de processament. Aquesta rapidesa de càlcul prové de la suma d'aproximacions de les sortides dels **Haar wavelet** i fent un ús eficient del càlcul integral de la imatge.

Els **Haar wavelets** tenen una estreta similitud amb els **Haar-like features** ja comentats. De la mateixa manera, es divideix la imatge en regions rectangulars determinats per la transformació Haar Wavelet, i pel valor integral de la suma dels píxels de la imatge, com es feia a **Haaar-like features**. Com en el cas dels **Haar-like features,** aquest procés comporta agilitat a l'accés i indexació de les regions de la imatge, contribuint a un cost computacional menor.

Els **Haar wavelets** transformen el contingut de les regions a una senyal discreta, obtenint desavantatge tècnic de no poder diferenciar la senyal de sortida. Amb aquest motiu el senyal de sortida d'una regió de la imatge vindrà donat per una sèrie d'aproximacions de les transformacions **Haar-wavelet** que conté. Aquestes aproximacions venen donades a partir de projectar la cada wavelet amb la matriu **Haar Transform.** El resultat final dels **Haar-wavelet** serviran per donar-nos la orientació del **keypoint**.

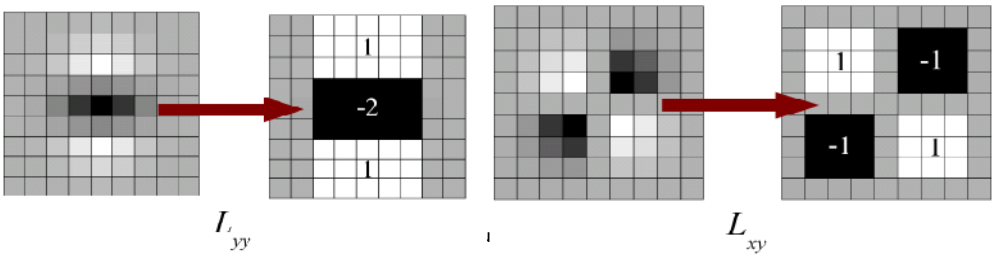
Pel que fa a la cerca dels **keypoints**, també seran obtinguts de les regions rectangulars en la qual s'ha subdividit la imatge. En el cas del **SURF**, els **keypoints** candidats seran els punts invariants obtinguts d'aplicar una piràmide gaussiana amb un filtre Laplacià, en comptes de

la diferencia de gaussianes (LoG) utilitzat al SIFT. Els diferents punts d'interès conformaran una matriu Hessiana de la següent forma:

$$H = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix}$$

Expressió 3.1

On L_{ii} és la sortida del filtre Laplacià



Aquestes transformacions localitzen els candidats a **keypoints**

Exemples de localització i matching de característiques amb SURF:

Objecte a seguir: Nas Transformació: Canvi de pose	
	

Figura 3.7

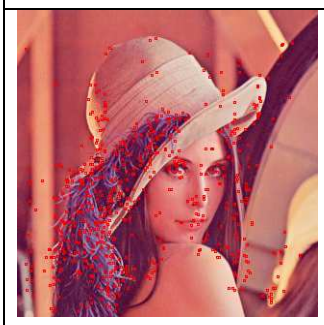
Transformació: Canvi de rotació 45º, 90º i 180º			
			

Figura 3.8

3.4 COMPARATIVA DE MÈTODES: SIFT I SURF

Anàlisi en quant al cost computacional dels dos mètodes:

	SURF	SIFT
Cost espacial	Mètode baix cost: 64 floats → 256 bytes Mètode estàndard: 128 floats → 512 bytes	Mètode estàndard 32 floats → 128 bytes
Cost temporal	2.4 segons	6 segons
Mitjana de keypoints detectats. Imatge 1024x768	1000 **keypoints** aproximadament	Més de 3000 **keypoints**

A l'hora de fer un anàlisi qualitatiu dels mètodes observem les següents característiques:

- Surf és un bon mètode per al reconeixement i seguiment de característiques, si hi ha hagut alguna alteració sobre la nova imatge tal com entelada, suavitzat o difuminació, a la literatura especialitzat trobem aquest fenomen com a **Gaussian blur**.
- També es pot afirmar, que el seguiment de **keypoints** dona bons resultats per a transformacions de rotació, tal i com es mostra a l'experiment anterior.
- Surf és unes tres vegades més ràpid computacionalment que Sift.
- Surf no és un bon mètode per al seguiment de característiques que experimenten un canvi de posició o un canvi de punt de vista. Tal com veiem a l'experiment del seguiment del nas d'un humà, la correspondència de **keypoints** entre les imatges no és prou precisa.
- Surf tampoc és una tècnica eficaç per a quan les imatges experimenten canvis d'il·luminació locals.

Finalment concloem que Surf és una variant de SIFT basada en l'aproximació, amb una coherent planificació d'afrontar la totalitat de la imatge, realitzant una subdivisió de l'espai i extraient-ne el màxim d'informació local, a través del càlcul simbòlic. Qualitativament, en comparació amb el mètode SIFT, es pot dir que dona pitjors resultats pel que fa a la detecció i correspondència de **keypoints**, però en canvi, el seu ràpid processament computacional li permet ser implementat en diverses aplicacions en temps real.

3.5 OPTICAL FLOW

3.5.1 METODE LUKAS-KANADE

L'algorisme de Lukas-Kanade actualment és un dels més importants i utilitzats a l'hora de realitzar el seguiment d'un flux òptic de punts. Una de les claus de la seva popularitat, recau en la seva senzillesa a l'hora de ser implementat sobre un subconjunt de punts, donades unes imatges d' entrada.

El mètode **Lukas-Kanade**[11] treballa exclusivament en el seguiment de característiques, és a dir, no s'encarrega del pre-processament d'aquestes. Això també aporta un cert grau de independència a l'hora d'aconseguir els **keypoints. Lukas-Kanade** es basa únicament en realitzar una examinació de l'espai al voltant dels **keypoints,** per a un post-procesament d'aquesta informació local. Com en els mètodes anteriorment comentats, es fa una subdivisió de l'espai, la dimensió d'aquesta sol ser anomenada **mida de finestra,** per a examinar els voltants d'un **keypoint**.

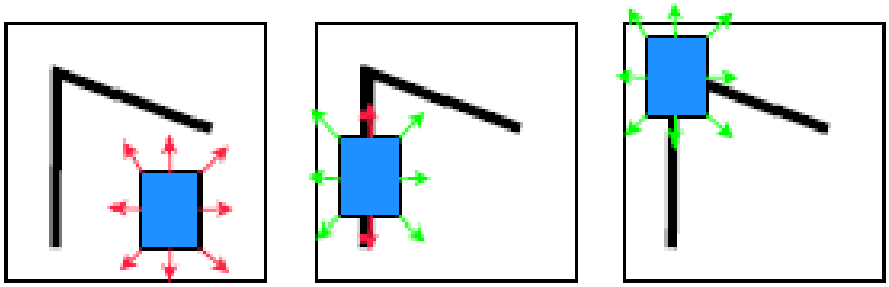
Una mida de finestra petit pot comportar, en algunes situacions, la deslocalització dels **keypoints**, degut a ràpids i llargs moviments d'aquests. Amb la intenció d'evitar aquest fenomen, l'algorisme de **Lukas-Kanade** desenvolupa un metodologia piramidal, controlant el moviment del flux dels **keypoints** començant amb imatges amb poc detall, i va pujant el detall d'aquestes uns tres o quatre nivells, normalment. Amb aquesta composició piramidal de detall, les finestres locals de cada imatge, sobre diferents nivells de detall, seran capaç de trobar moviments llargs de **keypoints.**

Abans d'entrar en el funcionament intern de l'algorisme, explicarem breument les dues tècniques emprades per a la selecció de **keypoints** des de la imatge d'entrenament**:**

3.5.2 APROXIMACIÓ DE HARRIS

Harris al 1988 va definir un punt invariant com aquell punt que tolera transformacions tals com translació, rotació, escalat i il·luminació. Partint d'aquesta premissa, va definir tres tipus

de regions que es podem trobar en una imatge: regió plana, regió de vora i regió de cantonada.



Amb el suport de la figura, veiem el primer cas, una regió plana, no presentaria cap canvi en qualsevol sentit on li apliquéssim una transformació. En el segon cas, una regió de vora, no presentaria cap canvi al realitzar transformació al llarg de l'eix vertical, és a dir, al llarg de la vora. Mentre que al tercer cas, en qualsevol dels sentit on féssim una transformació la regió seria variant. L'objectiu de l'aproximació de Harris es trobar **keypoints** que segueixen el patró de les regions planes. Per a la cerca d'aquesta propietat, Harris utilitza derivades de segon ordre sobre les regions, resultant una matriu Hessiana. La principal aportació de Harris, és el desenvolupament d'una matriu, anomenada **d'autocorrelació**, formada per a cada una de les regions que conformen la imatge.

$$H(f) = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix}$$

Expressió 3.2

$$M(x,y) = \begin{bmatrix} \sum_{-K \le i,j \ge K} w_{i,j} I_x^2 & \sum_{-K \le i,j \ge K} w_{i,j} I_x I_y \\ \sum_{-K \le i,j \ge K} w_{i,j} I_x I_y & \sum_{-K \le i,j \ge K} w_{i,j} I_y^2 \end{bmatrix}$$

Expressió 3.3

Segons la hipòtesi de Harris, a la matriu d'autocorrelació, podríem dir que es tracta d'una regió de cantonada si es troben dos eigenvalues suficientment alts. Empíricament es pot demostrar si dins de la regió hi ha una textura on es defineixen dues direccions. Amb el mateix raonament si trobem un sol valor d'eigenvalue alt, direm que es tracta d'una vora. Finalment si els valors tendeixen a zero, direm que es tracta d'una regió plana.

3.5.3 APROXIMACIÓ SHI TOMASSI

L'any 1994 Jianbo Shi i Carlo Tomassi, fan una modificació a l'algorisme de Harris. La modificació és molt senzilla, es pren un valor llindar dels eigenvalues anteriors, si es supera aquest valor llindar seran considerats com a regions invàlides. Aquesta aportació dona una forma més acurada i senzilla al mètode per a ser implementat.

Una vegada tenim el conjunt de **keypoints**, tornem a l'algorisme de **Lukas-Kanade.** La seva idea principal de funcionament es base en les tres següents assumpcions:

- **Lluentor constat:** el píxel pertanyent a un objecte dins d'una seqüència de vídeo, no exercirà un canvi brusc d'aparença entre dues imatges, o **frames,** consecutives. Aquest canvi, es produirà de forma suavitzada. Per tant, tampoc es produirà una transformació excessiva sobre els valors de lluentor dels píxels de la imatge. Si transcorre un canvi de lluentor, aquest es farà de forma escalada en el temps. Expressat mitjançant formalització matemàtica queda:

$$\boldsymbol{f(x,t) = I(x(t),t) = I(x(t+dt),t+dt)}$$

$$\frac{\partial f(x)}{\partial t} = 0$$

I(x;t) es la intensitat, o il·luminància, del píxel sobre el temps t. Desenvolupant la seva derivada parcial sobre el temps, zero indica nul·litat de canvi de lluentor.

- **Persistència temporal**: Amb aquesta premissa es pressuposa que els **keypoints** exerceixen moviments curts, és a dir, exerceixen transformacions de translació de amb valors baixos de distància neta. Per a descriure els moviments dels **keypoints,** sobre la dimensió horitzontal i vertical, utilitzarem la següent expressió:

$$\frac{dz}{dt} = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt}$$ Expressió 3.3

Reduint amb la antiderivada en ambdós costats, trobem la funció que descriu la posició dels keypoints a la superfície:

$$\underbrace{\left.\frac{\partial I}{\partial x}\right|_t}_{I_x}\underbrace{\left(\frac{\partial x}{\partial t}\right)}_{v}+\underbrace{\left.\frac{\partial I}{\partial t}\right|_{x(t)}}_{I_t}=0$$

Expressió 3.4

En primer terme tenim I_x la derivada espacial, que mesura la quantitat de canvi sobre l'eix horitzontal. Mentre que I_t es la derivada temporal que ens indica de canvi sobre el temps transcorregut. Finalment v pertany al valor de la velocitat de canvi. La velocitat de canvi sobre l'eix horitzontal s'expressarà amb:

$$v = -\frac{I_t}{I_x}$$

- **Coherència espaial:** S'assumeix que els **keypoints** d'una mateixa superfície, o veïns, desenvolupen un moviment semblant, que finalment, s'assumeix com igual.

Les bases principals de l'optical flow són les mateixes de la lògica espaciotemporal, citades i emprades durant el transcurs del projecte. En quant al cost computacional, si el número de **keypoints** és menor o igual a 500 característiques, es pot treballar perfectament en temps real.

Exemples d'aplicacions amb optical flow, seguiment d'ulls:

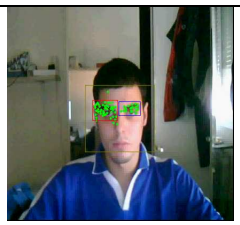
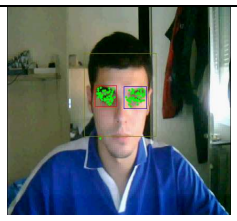
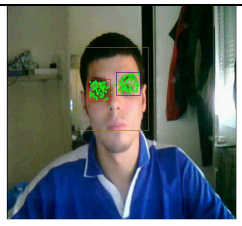
			
t=1s Inicialització	t=2s Seguiment	t=6s Canvi Escala	t=10s Rotació
			
t=14s Canvi lluentor i oclusions parcials	t=16s aprenentatge en temps real (amb ulleres)	t=20s Seguiment	t=31s Translació i escalat

Figura 3.10

El seguiment és prou robust sempre i quan es mantingui les condicions de la lògica espaciotemporal. El seguiment ha sigut de 250 **keypoints** per ull, amb el suport de Viola-Jones per a la cerca del rostre, amb la intenció de limitar la regió de treball.

3.6 MEAN SHIFT / CAM SHIFT

L'algorisme mean shift[11] es una variant prou robusta del mètode **matching template.** Com el mètode anterior, la seva finalitat és trobar màxim i mínims sobre la informació distribuïda d'una imatge. En compte de buscar un cert patró, com en el cas del **matching template** ¸aquest mètode es capaç de seguir distribucions continues d'informació visual. Aquestes distribucions comparteixen uns valors propers d'intensitat d'**il·luminància,** forma i mida. Com a eina per a treballar sobre les imatges s'utilitza l' **histograma**.

Aquest robust mètode conté un cert sentit estadístic en el procés de seguiment de característiques. Aquest caire es veu quan ignora els valors atípics de la mostra, que a la practica sol ser soroll. Això es tradueix en que no es tenen compte aquells valors que produeixen pics, o llunys de la norma de l' histograma. El control i examinació d'una regió de la imatge, s'implementa a través d'una finestra local. Aquesta finestra anirà basculant sobre la totalitat de la imatge, oferint-nos els resultats i posicionant-se sobre la regió de **tracking.**

L'algorisme de Cam Shift es basa en:

1. Escollir la mida de finestra, basat en :
   - Posició inicial de l'objecte a seguir
   - Tipus de moviment de l'objecte a seguir (uniforme, polinòmia, exponencial o Gaussià)
   - Forma de l'objecte (simètric o asimètric, possiblement rotat, arrodonit o rectangular).
   - Mida mitjana de l'objecte.

2. Obtenir el centre de gravetat del contingut de la finestra, basat en les densitats de l' histograma.
3. Posicionar la finestra al centre de gravetat.
4. Tornar al pas 2 fins que no existeixi moviment a la finestra.

Donant una definició formal al procediment esmentat, utilitza la disciplina de trobar el una estimació de densitat del nucli de l'objecte a seguir (**kernel density estimation).**
La variant Cam Shift es que la es sol utilitzar a les implementacions de visió per computador. Aquesta variant ajusta la mida de finestra sobre cada frame de seguiment de característiques.

És un mètode que funciona robustament sobre el seguiment d'objectes amb moviments continus i lents. No es tolerant a canvis bruscos d'il·luminació i mida de l'objecte. Pot funcionar bé sobre el tracking de cares on l'escena no es molt variant. Es fortament depenent de l'escala de color la qual es projectada la imatge. Funciona bé per al tracking de cares, ja que el color de la pell es descriptor prou característic per a realitzar seguiments.

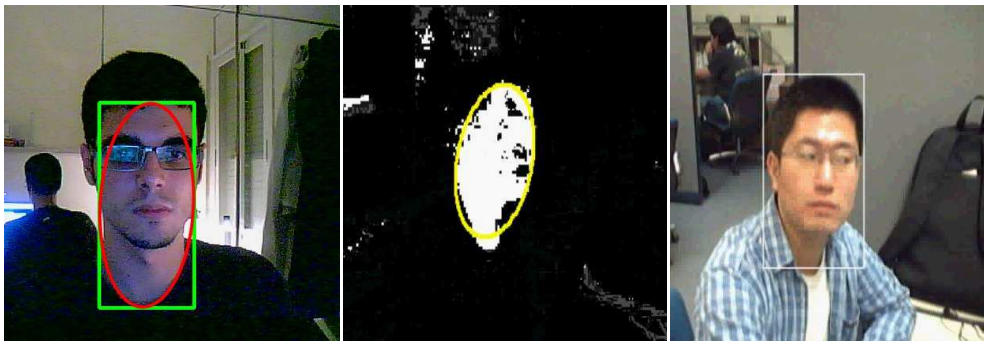


Figura 3.11

3.7 BLOB DETECTION

Aquesta tècnica cerca objectes a l'escena amb un nivell de lluentor, o obscurs, diferenciats de la totalitat de l'escena. Per a la cerca d'aquests objectes s'utilitza la diferenciació de les gaussianes, a través de la projecció d'un espai escala a la imatge. Per a filtrar el nombre d'imatges es donen certs valors de mides vàlides, tals com area, longitud vertical i horitzontal.

Una vegada detectats aquests objectes, es guarda l' histograma per al seguiment i cerca en el següent frame. El procés de seguiment exerceix un template matching, actualitzat frame a frame, de l'objecte, o **blob**[12], anteriorment detectat o seguit.
Pot funcionar amb robustesa detectant objectes molt diferenciats de la resta de la imatge, tals com vehicles sobre una carretera. No es apte per aplicacions on el **background** canviï. Es capaç de fer un seguiment correcte d'un objecte, sempre i quan aquest moviment sigui continu i la mida es mantingui uniforme frame a frame.

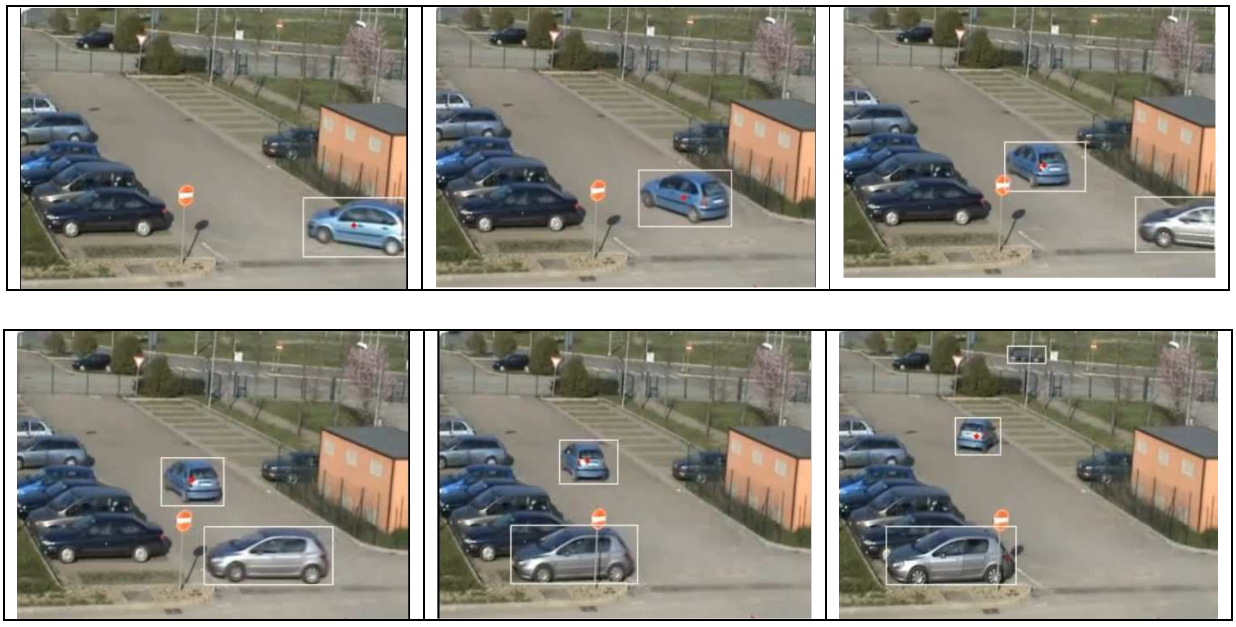


Figura 3.12

4. MODELS ESTADISTICS DE FORMA I APARENÇA

4.1 INTRODUCCIÓ ALS MODELS ESTADISTICS

Durant el transcurs de l'anàlisi dels diferents mètodes, podem fer una primera observació: les tècniques de detecció i seguiment d'objectes emprades queden limitades en situacions de soroll, distorsió o deformació. La majoria de situacions errònies de les tècniques anteriors, venen donades degut a la manca de coneixement, total o parcial, de la seva complexitat estructural, forma o aparença de l'objecte a detectar o a seguir,en un instant determinat **t**.

Entenem que aquests mètodes comentats, se'ls ha d'afegir coneixement, o dotar d'una certa intel·ligència per treballar l'entorn. Dita capacitat cognitiva, no ve donada de forma explicita en les tècniques anteriorment comentades, sinó es un agent extern, tal com el mateix dissenyador de l'aplicació, que ha de formular una lògica robusta capaç de predir i corregir situacions errònies provocades pel context de l'aplicació. Aquest control sobre el procediment comporta certes limitacions de llibertat de moviment i transformació dels objectes a detectar o seguir, que finalment es tradueix en aplicacions que només funcionen en entorns controlats.

Els mètodes basats en models o patrons, en principi, tenen la capacitat de conèixer parcialment la forma i aparença de l'objecte que intentem detectar o seguir. Amb la els mètodes de cerca i seguiment de models o patrons sobre la imatge, tals com Viola-Jones, amb una bon conjunt d'aprenentatge segurament obtindrem una taxa d'encerts prou bona, produint una interpretació plausible de la realitat, sempre i quan l'objecte a detectar entri dins dels valors del conjunt d'aprenentatge. Imaginem que durant una seqüència detectem un objecte frame a frame amb **Viola-Jones**, però en un cert instant de temps, es produeix una variació de il·luminació sobre una regió de l'objecte i no es detecta l'objecte. **Viola-Jones** ha rebutjat la totalitat l'objecte degut a que una regió no entrava dins del conjunt d'aprenentatge, tot i que la resta de l'objecte si que pertanyia. Si haguéssim conegut la complexitat estructural de l'objecte que estàvem detectant, podríem haver deduït que l'objecte que havia experimentat un canvi d'il·luminació continuava tenint una forma o aparença semblant als del model d'aprenentatge.

Precisem d'un mètode que analitzi les diferents parts d'un objecte, proporcionant-nos un factor de **matching**, a través de l'examinació de la complexitat estructural, de forma o aparença en la seva totalitat. Aquest factor ha de tenir una alta variabilitat. L'alta variabilitat proporcionarà un alt nivell de deformació a l'objecte, sempre mantenint les característiques essencials que li fan pertànyer a una certa classe o model. Obrint d'aquesta manera el rang de possibles positius sobre el conjunt d'aprenentatge. El tècnica a utilitzar serà el Active Shape Models/Active Appereance Models.

Aquests mètodes consisteixen en crear models d'aprenentatge sobre la forma i aparença, amb un alt grau de variabilitat, per a després trobar correspondència o matching en una nova imatge. Per al seguiment de l'objecte **x** a l'instant **t+1** es fa la següent assumpció:

$$\mathbf{x} \approx \bar{\mathbf{x}} + \Phi \mathbf{b}$$

On **x** és l'objecte l'instant **t+**1, fruit d'una aproximació de la imatge $\bar{x}$ a l'instant **t**, modificada sobre el conjunt de paràmetres de deformació **b**, que més endavant es detalla.

4.2 ACTIVE SHAPE MODELS

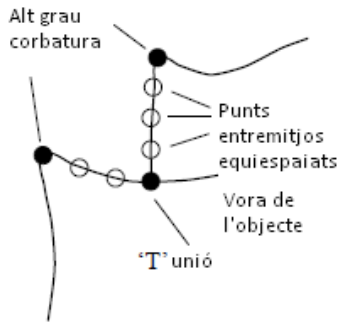
4.2.1 CONFECCIÓ D'UN MODEL

Començarem a descriure els models estadístics de forma, Active Shape Models, que utilitzarem per a representar objectes sobre les imatges. La forma d'un objecte serà representada a través d'un conjunt de **n** punts, cada un d'aquests l'anomenarem **landmark**. La magnitud de **n** variarà en conseqüència a les dimensions de l'objecte de que volem

descriure. Els punts son representacions de posició (x,y) sobre imatges 2D i (x,y,z) en 3D.

1. **Elecció dels lankdmarks:**

L'elecció dels òptims landmarks vindrà donada, per aquells punts que siguin consistentment localitzats i diferenciats entre l'objecte i el background. Normalment sobre les primeres imatges del conjunt d'aprenentatge, es fa manualment a través del coneixement de l'expert. Una vegada tenim una aproximació de la forma de l'objecte, es poden utilitzar mètodes automàtics o semiautomàtics d'elecció de **landmarks.**
A imatges en dues dimensions els punts poden ser localitzats sobre cantonades pronunciades i unions 'T'. El problema es que els objectes no solen tenir una gran quantitat d'aquests tipus de punts, llavors també es localitzaran els punts entremitjos entre cantonades i unions 'T', de manera equiespaciada.



Si una forma es descrita per **n** punts sobre una dimensió **d**, representarem la forma per un vector **nd**, format per la concatenació de la posició individual dels landmarks. Per exemple, sobre una imatge 2-D podem representar **n** landmarks sobre el vector **x** com**:**

$$\mathbf{x} = (x_1, \ldots, x_n, y_1, \ldots, y_n)^T$$

Donat una conjunt d'entrenament de **s** imatges, generarem **s** vectors $\mathbf{x_j}$. Abans d'entrar a l'anàlisi estadístic d'aquests vectors, es molt important que aquestes formes,representades ara per un vector, estiguin totes posicionades respecte una mateix origen de coordenades. Aquest procés s'anomena alineació.

2. **Alineació de les formes**

La tècnica per a posicionar totes les formes sobre el mateix origen de coordenades serà la de Procrustes Analysis [14 app_models.pdf] . Aquesta tècnica alinea cada forma amb l'objectiu de minimitzar la suma de distancia de cada forma respecte la forma promig $\bar{x}$:

$$(D = \sum |\mathbf{x}_i - \bar{\mathbf{x}}|^2)$$ Expressió 4.1

El problema de minimització es resol mitjançant el següent algorisme iteratiu:

1. Traslladar cada forma per tal de el seu centre de gravetat sigui el seu origen.
2. Elegir una forma com a estimació inicial de la forma promig, i escalar aquesta per a que es compleixi que $|\bar{x}| = 1$.
3. Guardar aquesta primera estimació com a $\overline{x}_o$,sent la referència de la forma promig.
4. Alinear totes les demes formes prenent com a referència la forma promig.
5. Tornar a calcular la forma promig en funció de les distancies de les demes formes a aquesta
6. Guardar aquesta nova estimació com a $\overline{x}$, sent la referència de la forma promig i escalant la forma per tal que es compleixi que $|\bar{x}| = 1$.
7. Si no es produeix convergència, tornem al pas 4. Prenen com a criteri de convergència que no hi hagis canvis considerables en el nou càlcul de la forma promig.

Les operacions que es duran a terme durant el procés d'alineació de la forma repercutiran en la seva distribució final. Diferents aproximacions d'alineació poden produir diferents distribucions de les formes alineades. El nostre objectiu es mantenir una distribució compacta de les formes, amb la mínima no-linearització.

3. Proporcionar variabilitat al model

Fins ara el model que disposem conté un conjunt vectors de landmarks alineats. Aquest vectors formen una distribució sobre el **nd** espai dimensional on es projecten. Si projectem noves formes sobre aquest espai, i es posicionen d'una manera similar al conjunt d'aprenentatge , podem decidir si els nous exemples son plausibles o no.
Per examinar la constitució de la forma es busca una model parametritzat de la forma
X = M(b)
On **b** es un vector amb els paràmetres del model. Diferents models generaran diferents vectors, x. Creant una distribució dels paràmetres de forma del model, **p**, seriem capaços de limitar la projecció dels nous objectes. Amb aquestes projeccions es podria exercir una classificació examinant la similariat del conjunt d'entrenament amb els nous objectes.

Per a simplificar el problema i millorar el tractament de els dades, reduïm la dimensionalitat **nd** que prové dels vectors que contenen els landmarks. Una solució molt efectiva es processar l'algorisme Principal Component Anàlisis (PCA) a la informació de forma del conjunt d'aprenentatge. Aplicant el PCA sobre les dades es formarà un núvol de punts al projectar la informació de cada de forma, sobre un espai de dimensió reduïda. El procés esmentat es detalla en els següents passos:

1. Trobar el centre de masses de la informació de forma del conjunt d'aprenentatge de mida **s.**

$$\bar{\mathbf{x}} = \frac{1}{s}\sum_{i=1}^{s}\mathbf{x}_i$$ Expressió 4.2

2. Càlcul de la covariància de la informació de forma del conjunt d'aprenentatge.

$$\mathbf{S} = \frac{1}{s-1}\sum_{i=1}^{s}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$$ Expressió 4.3

3. Obtenció dels **eigenvectors,** φ_i amb els seus corresponents **eigenvalues** λ_i. Reordenació dels eigenvalues en ordre descendent, provocant la reordenació dels eigenvectors.

4. Elegir el nombre d'eigenvectors a utilitzar en funció del percentatge de representació dels eigenvalues. Es sol utilitza el nombre d'eigenvectors que representen un 98% de la informació a través dels eigenvalues.

Si φ conté **t** eigenvectors corresponents als majors eigenvalues, llavors podem aproximar la projecció d'una nova forma **x** amb el conjunt d'entrenament $\overline{x}$ amb la següent expressió:

$$\mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{\Phi b}$$ Expressió 4.4

On b es un vector de **t** dimensió donat per:

$$\mathbf{b} = \mathbf{\Phi}^T(\mathbf{x} - \bar{\mathbf{x}})$$ Expressió 4.5

El vector **b** defineix un conjunt de paràmetres que defineixen una deformació del model pertanyent al conjunt d'aprenentatge.
Per a determinar si un model es plausible, quan projectem la forma de la nova imatge sobre les formes del conjunt d'aprenentatge, obtindrem un vector **b** amb els paràmetres de la forma. Crearem p(**b**) com a un estimador de la distribució de la forma sobre el conjunt d'aprenentatge. Decidirem si aquesta nova forma es plausible si $p(\mathbf{b}) \geq p_t$, essent p_t un valor mitjà de distribucions del conjunt d'aprenentatge elegit arbitràriament.

4. **<u>Procés de convergència de punts</u>**

Una vegada hem projectat la nova imatge sobre el model après, i per tant, hem obtingut hem obtingut el vector **b** que conforma el conjunt de paràmetres que corresponen a la forma.

Abans, una vegada projectada la nova forma, ens sorgeix el problema de considerar plausible aquesta forma. La primera projecció de la nova imatge, ens pot instanciar a una forma amb soroll:

Figura 4.2

Canviarem el nostre vector **b** per **b'**, el qual seran els paràmetres de forma d'una imatge,**x',** plausible, del conjunt d'aprenentatge molt propera a **x:**

$$\mathbf{b}' = \mathbf{\Phi}^T(\mathbf{x}' - \bar{\mathbf{x}}).$$

Obtenint una primera forma sense soroll:

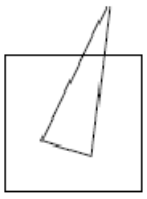


Figura 4.3

Una vegada tenim una primera aproximació de la forma, de manera plausible, iniciem el procés de convergència i adaptació d'aquesta forma als nous punts de la imatge.
Un exemple del model entrenat es descriu per els paràmetres de forma,**b**, en combinació amb transformacions tals com: translació (X_t ,Y_t), rotació (θ) i escalat (s).

La posició dels punts del model sobre la imatge, **x**, vindrà donats per:

$$\mathbf{x} = T_{X_t,Y_t,s,\theta}(\bar{\mathbf{x}} + \mathbf{\Phi}\mathbf{b})$$

Si volguéssim convergir un exemple del model al punt (x,y) ens quedaria:

$$T_{X_t,Y_t,s,\theta}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} X_t \\ Y_t \end{pmatrix} + \begin{pmatrix} s\cos\theta & s\sin\theta \\ -s\sin\theta & s\cos\theta \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}$$

Expressió 4.6

En el cas de vulguem trobar una correspondència entre un exemple del model après, sobre el conjunt de punts de la nova forma,**Y**, ens resulta una expressió que cal resoldre minimitzant:

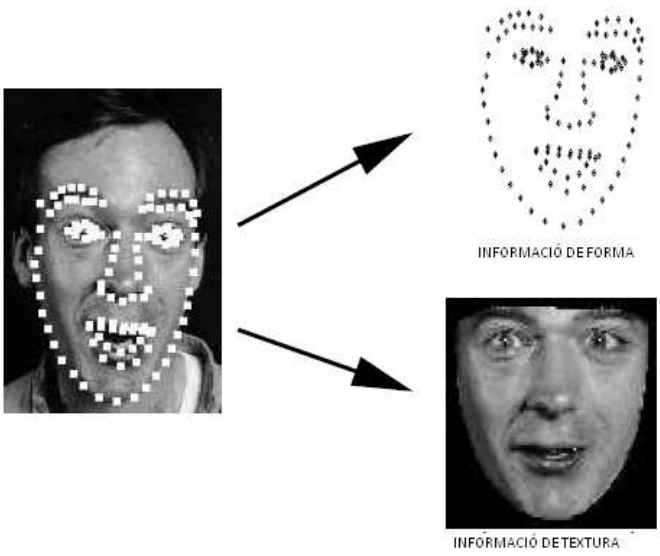
$$|\mathbf{Y} - T_{X_t,Y_t,s,\theta}(\bar{\mathbf{x}} + \mathbf{\Phi}\mathbf{b})|^2$$

Expressió 4.7

4.3 ACTIVE APPEREANCE MODELS

Tal com hem vist, la informació de forma, i la seva variabilitat, proporciona una gran quantitat d'informació per a descriure objectes o estructures. Per acabar de sintetitzar un objecte sobre una imatge, podem afegir una altra propietat apart de la forma, aquesta pot ser la textura i la seva variabilitat. Tal com descrivim els objectes seran representats a través d'una variació d'un model de forma, una variació d'un model de textura i la correlació entre aquestes dues propietats. La introducció de la textura aporta un alt volum d'informació descriptiva d'un model.

Per textura entenem graus d'intensitat i color sobre una regió de la imatge. Aquesta regió es produeix l'acotació de l'objecte a través de la informació de forma.



Per a minimitzar l'efecte de la variació de la lluminositat, normalitzem cada textura aplicant un factor d'escalat **α** i un desplaçament **β**.

$$\mathbf{g} = (\mathbf{g}_{im} - \beta\mathbf{1})/\alpha$$

Els valors **α** i **β** vindran donats per una valor promig del total de textures a normalitzar.
Un cop normalitzada la informació, tal com vam fer al tractament de la informació de la forma, reduirem la dimensionalitat de la informació de textura, aplicant l'algorisme PCA.

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_g\mathbf{b}_g$$

Una textura exemple del model,**g,** vindrà donat pel valor promig de les textures, $\overline{g}$, més la informació reduïda, amb PCA, donada per un conjunt de valors d'escala de grisos, $\mathbf{b_g}$, i un conjunt de factors de variació ortogonal d'aquests, $\mathbf{P_g}$.

La representació del model ara es farà per mitjà d'una combinació de forma i aparença.
Aquest informació estarà a $\mathbf{b_s}$ i $\mathbf{b_g}$. Per a mantenir una correlació entre les variacions de forma i textura, aplicarem PCA a tot el conjunt combinat d'informació expressat de la següent manera:

$$\mathbf{b} = \begin{pmatrix} \mathbf{W}_s\mathbf{b}_s \\ \mathbf{b}_g \end{pmatrix} = \begin{pmatrix} \mathbf{W}_s\mathbf{P}_s^T(\mathbf{x} - \bar{\mathbf{x}}) \\ \mathbf{P}_g^T(\mathbf{g} - \bar{\mathbf{g}}) \end{pmatrix}$$

On **x** es refereix a la informació de forma, mentre que **g** a la informació de textura. En quant a $\mathbf{W_s}$ is la diagonal de la matriu de coeficients, o pesos, de cada paràmetre referent a la informació de forma. Aplicant PCA a tot el conjunt **b**, ens redueix la informació:

$$\mathbf{b} = \mathbf{P}_c\mathbf{c}$$

On $\mathbf{P_c}$ són els **eigenvectors** i **c** és el vector que conté els paràmetres de descripció combinada, de forma i aparença. Analitzant la informació de forma i textura de manera separada:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}_s\mathbf{W}_s^{-1}\mathbf{P}_{cs}\mathbf{c} \quad , \quad \mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_g\mathbf{P}_{cg}\mathbf{c}$$

$$\mathbf{P}_c = \begin{pmatrix} \mathbf{P}_{cs} \\ \mathbf{P}_{cg} \end{pmatrix}$$

El procés d'elegir exemples del model que siguin plausibles. segueix el mateix patró que a la tècnica emprant només forma, però ara treballarem amb els valors del conjunt $\mathbf{P_cC}$.

4.3.1 INTERPRETACIO D'UNA REGIO DE LA IMATGE

Per a la interpretació de la imatge utilitzant un model, necessitem trobar un conjunt de paràmetres de tinguin un alt grau de correspondència, amb un exemple del conjunt d'aprenentatge del model. Com ja sabem, aquest paràmetres defineixen la forma, posició i aparença de l'objecte o estructura sobre la imatge.
Necessitarem una funció *F(c)* la qual ens indiqui si ens estem apropant o allunyant a un exemple del model, donats uns paràmetres **c**. Aquesta funció ens ha de representar la probabilitat de que un model estigui descrivint l'objecte o estructura que es troba a la imatge, $P(\mathrm{c}|\mathbf{I})$, on **I** és la imatge. Llavors el problema es redueix a trobar els paràmetres que minimitzin aquesta funció. Una aproximació, es mesurar la diferencia entre els paràmetres trobats a la imatge i la dels paràmetres d'un exemple del model. Una mínima diferencia ens conduiria a una bona aproximació.

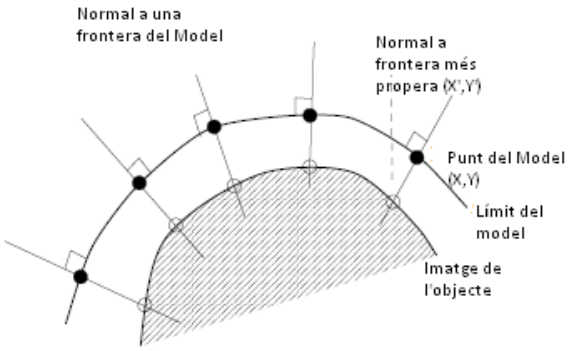


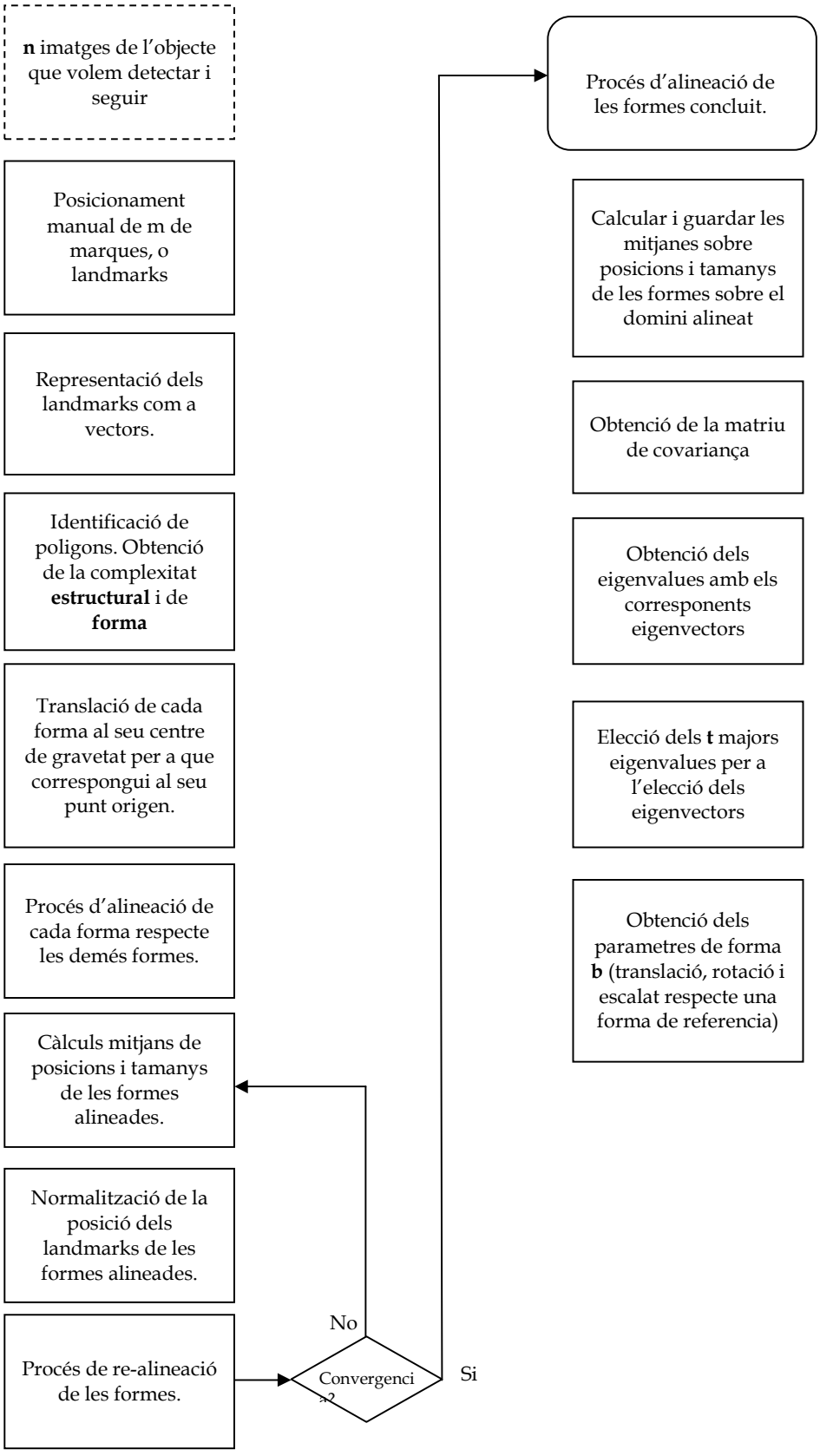
Figura 4.5

Donats els punts del conjunt d'aprenentatge del model continguts en **X**, i els més propers de la imatge que estem tractant a **X'**, la mesura de l'error vindrà donada per la minimització de l'expressió següent:

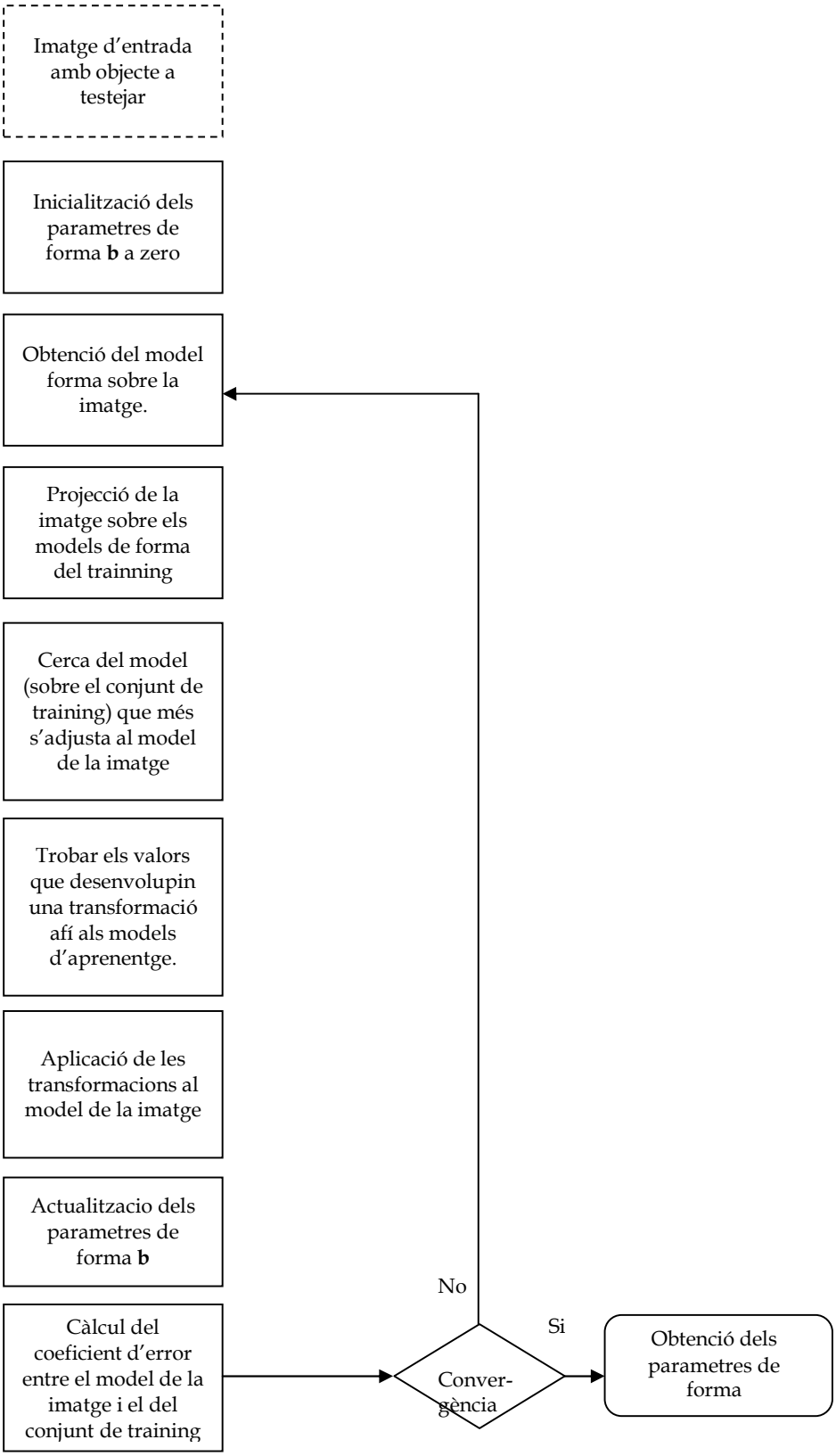
$$F(\mathbf{b}, X_t, Y_t, s, \theta) = |\mathbf{X}' - \mathbf{X}|^2$$ Expressió 4.8

El procediment de convergència per als paràmetres de textura, es calcula directament sobre l'**eigenspace** al projectar directament les característiques d'aparença de la regió continguda per les característiques de forma, sobre els paràmetres de textura del conjunt d'entrenament del model. Una valor de distancia entre el nucli, o centre de masses, de la projecció de les característiques de textura del conjunt d'aprenentatge i la textura que estem tractant, ens indicarà si es considera valida. Per tant, l'objectiu tornarà a ser reduir la dita distancia.
Aquest procediment tornarà a resultar iteratiu, actualitzant els paràmetres de forma i textura en cada iteració, fins a trobar convergència. Per tal de ser robust i eficient, durant procés s'utilitzaran piràmides gaussianes per tal de treballar amb diferents resolucions de la imatge, que aportin informació de forma i textura.

4.4 DIAGRAMA PROCÉS D'APRENENTATGE ASM

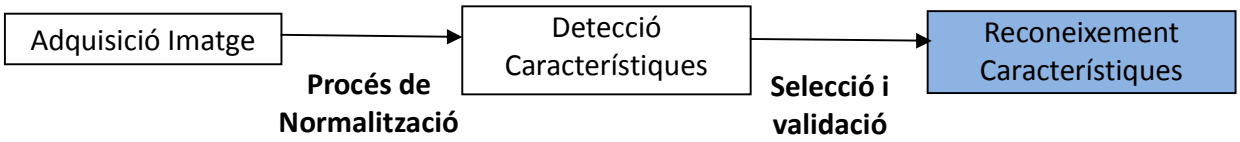


4.5 DIAGRAMA PROCÉS DE TEST ASM



5. TECNIQUES DE MACHINE LEARNING

Una vegada analitzades les diferents tècniques de detecció i seguiment de característiques, avancem a l'últim bloc del nostre sistema.



Ens trobem en una situació on el sistema detector ha trobat un recull de paràmetres característiques d'un objecte o estructura, que segons la intel·ligència dotada a aquests sistemes, es considera com a cert positiu. En molts problemes, a vegades, no tan sols ens interessa la informació binaria de si existeix l'objecte o no, sinó exercim un anàlisi del comportament de l'objecte al context. Per tant, ens trobem davant una situació de classificació d'estats.

La classificació d'estats d'un objecte esdevé de les tècniques anomenades **Machine learning**. El **Machine learning** es una disciplina científica basada en el disseny y desenvolupament d'algorismes que permeten als computadors, projectar i classificar els comportaments de dades empíriques. Una de les principals línees d'investigació d'aquestes tècniques es la capacitat d'aprendre models automàticament, per a després, exercir un robust reconeixement de noves dades, realitzant decisions intel·ligents basades en l'aprenentatge.

La principal dificultat radica en el fet de que el conjunt de tots els possibles comportaments, o estats, de totes les possibles entrades al sistema de reconeixement, pot arribar a ser massa complex per a ser descrit en termes generals sobre llenguatges de programació. Un dels problemes típics també sol ser, quan agrupem molts estats diferents per a una mateixa entrada de dades. Aquest fet comporta certa compactació dels **clústers** que identifiquen, o classifiquen, un comportament, estat o inclús un objecte. Per **clústers** entenem com agrupacions, o d'una forma gràfica com a núvols de punts, on es concentren objectes, estats o comportaments de la mateixa classe davant un cert context.

Seguidament comentarem les diferents tècniques analitzades que esdevenen del Machine learning:

- Boosting
- Support Vector Machine
- Nearest Neigbor
- Xarxes Neuronals
- Combinació de classificadors binaris.

5.1 BOOSTING

El **Boosting**, va ser creat per Kearns [13,14], que va proposar la idea de crear varis classificadors lleus en compte d'utilitzar només un molt discriminant. Per classificador lleu, o **weak** a la literatura, entenem per un classificador que considera com a cert positiu, quan el conjunt de les característiques d'un objecte, es consideren valides al voltant del 60%-70%, sempre una taxa d'encert que superi l'atzar. Pel contrari un classificador molt discriminant es aquell que considera com a cert positiu, quan el global de les característiques d'un objecte es consideren com a vàlides en un 90% o més.

La variant més utilitzada es **AdaBoost,** la qual utilitza un seguit de classificadors lleus en cascada, assignant un coeficient, o pes, a cada un. El pes indica la magnitud de classificació, es a dir, si és més o lleu o menys. Aquest pes serveix per a crear la cascada de classificadors en ordre ascendent, és a dir, des de el classificador menys discriminant al que més. Aquesta tècnica es va emprar a l'hora de detectar objectes, o patrons, amb **Viola-Jones** .

Adaboost és sensible al efectes de soroll i als valors atípics. Un dels avantatges es que evita amb robustesa els problemes **d'overfitting**, aprenent gran variabilitat de patrons.
El Boosting es un classificador supervisat, i com a la majoria d'aquests tipus, la seva taxa d'encert dependrà molt de la robustesa del conjunt d'entrenament sobre el qual es basa a fer la classificació.

5.2 SUPPORT VECTOR MACHINE

Es tracta d'un classificador supervisat, utilitzat per tant per a problemes de regressió com de classificació. Donat un conjunt d'entrenament es capaç d'etiquetar els exemples en una o dues categories, per això diem que es tracta d'un classificador binari.

Donant una visió gràfica de l'espai de característiques, Support Vector Machine[15] construeix un hiperplà o un conjunt d'hiperplans, sobre un espai d'altes o infinites dimensions, on pot ser utilitzat per la classificació o regressió. Per tant, aquest algorisme realitza una tasca iterativa de buscar l' hiperplà on existeix la màxima distancia, o separació, entre dos clústers de dades. Per a la cerca d'aquest hiperplà, si s'escau, s'augmenta la dimensió de l'espai de característiques. Per tant el Support Vector Machine tracta un problema de maximització de distancies.

El procés de reconeixement basa la classificació segons on es posicionin les dades característiques respecte l'eix de d'hiperplà.

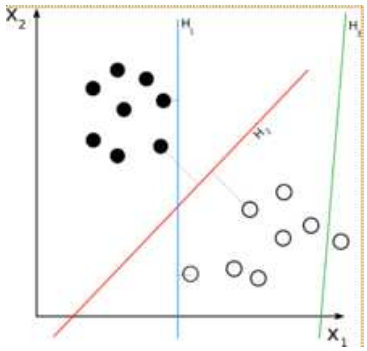


Figura 5.1

L' hiperplà H3, no separa els dos clústers. L' hiperplà H1 no aconsegueix la maximització de les distancies entre cada clúster. H2 es l' hiperplà òptim.

5.3 NEAREST NEIGBOR

L'algorisme **Nearest Neigbor**, o **k-nearest neigbours** *(k-NN)*, és un mètode de classificació basat estrictament, en la projecció de l'espai de característiques del conjunt d'aprenentatge. Es potser l'algorisme de **Machine learning** més simple d'utilitzar. La classe d'un objecte ve donada per la seva proximitat local als objectes de l'espai de característiques del conjunt d'aprenentatge. Per a definir la classe a la qual pertany un objecte, o estat, li associem un valor sencer positiu,***k***, en forma de comptador d'associació de classe. Per exemple si elegim com ***k*=1,** associarem l'objecte a la classe que pertany l'objecte més proper del conjunt d'aprenentatge.

En problemes de regressió la metodologia és molt similar. El valor d'assignació que estem buscant serà produït per la mitjana de valors dels **k** veïns més pròxims del conjunt d'aprenentatge.

K sol tenir un valor imparell, així evitem situacions d'empat.

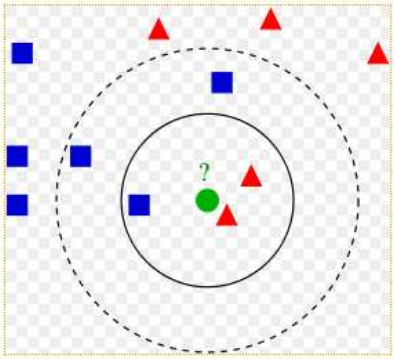


Figura 5.2

*La mostra de test es el punt verd. Si **k=1** verd pertany a la classe dels triangles vermells. El mateix passaria per a **k=3**, però en el cas de **k=5** el test pertany a la classe dels quadrats blaus, ja que hi ha 3 quadrats i només 2 triangles.*

5.4 XARXES NEURONALS ARTIFICIALS

Es un mètode molt complex sobre el qual ens podríem estendre molt. Només explicarem a grans trets per a que serveixen i com treballen, sense entrar en detall el seu funcionament intern.

Les xarxes neuronals artificials, **xarxes neuronals**, són un model matemàtic, que treballant amb els llenguatges de computació es tradueix en un model computacional, que intenten simular la estructura i el funcionament d'una xarxa neuronal biològica. Es tracta d'un sistema adaptatiu que canvia la seva estructura segons la entrada d'informació externa, és a dir, aquelles dades que ens interessen tractar o classificar, o internes, modificacions de les dades externes i càlculs produïts per la pròpia xarxa. Aquesta adaptació es produeix a la fase d'aprenentatge.

Les xarxes neuronals solen utilitzar eines de modelatge no-lineals per al tractament de les dades. Això comporta un alta complexitat al sistema, alhora de crear una la relació d'informació d'entrada i sortida, que serveix per a trobar correspondències, o patrons semblants per a la classificació.

La utilitat dels models de xarxes neuronals artificials es troben en el fet de que poden ser utilitzats per a inferir una funció a través de les seva observació. És particularment útil en aplicacions on la complexitat de les dades és molt elevada, o realitzar un disseny precís de comportament es molt poc pràctic.

En moltes aplicacions d'enginyeria trobem implementades xarxes neuronals, alguns exemples són:

- Aproximació de funcions, anàlisi de regressió, predicció de series, aproximació de **fitness** i modelatge
- Classificació
- Processament de dades, filtratge, **clustering** o agrupació i compressió
- Robòtica

5.5 COMBINACIO DE CLASSIFICADORS BINARIS

Sovint **SVM** és converteix en una bona tècnica de classificació automàtica. Degut a la seva naturalesa dicotòmica, sorgeix la necessita d'implementar nous mètodes on es puguin resoldre problemes multi classe. Amb aquesta idea es proposen dues idees per ampliar aquest mètode a problemes multi classe, mitjançant la combinació de classificadors binaris.

5.5.1 ONE-AGAINST-ALL

Es construeixen ***k*** classificadors que defineixen ***n*** hiperplans que separen la classe ***i*** de les ***k-1*** restants. Per exemple, per un problema de 4 classes, es crear classificadors 1 vs 2-3-4, 2 vs 1-3-4, 3 vs 1-2-4 i 4 vs 1-2-3. Al rebre noves dades a classificar, es sotmès als ***k*** classificadors, escollint com a resultat aquell que maximitza el marge, o distancia entre el clúster i l' hiperplà. Figura 5.3

$$\hat{C}_i = \arg \max_{i=1,\ldots,k} (w_i x + b_i)$$

W correspon a d'hiperplà, mentre que **b** correspon al desplaçament del clúster respecte l' hiperplà.

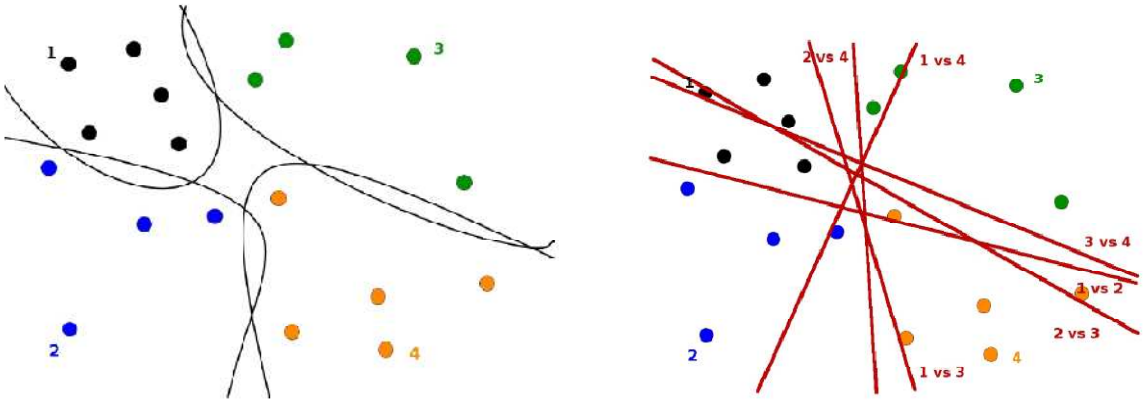


Figura 4.5

5.5.2 ONE-AGAINST-ONE

Es construeixen $\frac{k(k-1)}{2}$ classificadors, un per a cada parell de classes possibles, enfrontant totes les classes una a una. Per exemple, sobre un problema amb 4 categories generaríem els següents classificadors: *1 vs 2, 1 vs 3, 1 vs 4, 2 vs 3, 2 vs 4* i *3 vs 4*. Una vegada realitzat això, es sotmet cada entrada dades de test a totes els classificadors, on s'afegeix un vot a la classe guanyadora en cada cas. Finalment, aquella que més vots obtingui serà la classe proposada pel sistema. Figura 5.4

6. IMPLEMENTACIO PRACTICA

6.1 DISSENY

Per aconseguir tots els objectius del projecte, elaborem una part pràctica on és plantegen les possibles solucions d'un problema molt explotat per la visió per computador, el tractament de la **posició del cap.**

Durant la última dècada aquesta qüestió ha obert múltiples línies d'investigació aprofitant les diferents tècniques de processament d'imatges i **Machine learning**. El nostre objectiu, una vegada elaborat un estudi sobre la literatura del tractament d'aquest problema, consistirà en la elaboració d'un prototip basat en les tècniques i mètodes anteriorment comentats. En alguna ocasió aquests mètodes seran ajustats i modificats parcialment per aconseguir els resultats òptims donat el context que estem treballant.

Es realitzen dues solucions al problema basades en el seguiment de característiques facials, utilitzen metodologies diferents. Per a definir la dimensionalitat del problema i ajustar-nos a un context clar, els prototips han de adaptar-se a les següents propietats:

- Les característiques facials segueixen una lògica espaciotemporal.
- Els prototips han de tenir una alta variabilitat de subjectes a tractar.
- Ha de funcionar en temps real.
- Ha de treballar amb robustesa davant de canvis d'il·luminació i efectes de soroll.
- Ha de funcionar sobre computadors de prestacions normals, tals com ordinadors domèstics.

La primera solució plantejada es basa en una multi classificació de la posició del cap, creant un conjunt, ben definit, d'aprenentatge per a després avaluar amb diferents mètodes de classificació de patrons. En aquest prototip s'utilitzen mètodes d'extracció de característiques per a després fer un us exhaustiu del **Machine learning,** per a dur a terme la tasca de reconeixement de patrons, en aquest cas, de posicions del cap. D'aquest mètode existeixen nombroses publicacions tractant el mateix problema. S'avaluaran els resultats obtinguts, per a després comentar i discutir el mètode emprat.

La segona solució s'utilitzen els Active Appereance Model per a la detecció i seguiment de característiques. S'analitzarà la robustesa de detecció i seguiment d'aquest mètode, davant de situacions de soroll, canvis d'il·luminació i variabilitat del subjecte a tractar. Es proposa una classificació de l'espai de característiques per al tractament de la posició del cap, amb la fi d'analitzar i discutir l'eficàcia i rendiment del mètode davant el problema plantejat.

6.2 SOFTWARE

A l'hora de triar el software per al desenvolupament d'una aplicació,es va tenir en compte que el llenguatge elegit havia de ser capaç de treballar amb densos volums de dades, com es el cas de les imatges. També s'ha de ser conscient que una aplicació treballant l'àrea de la visió per computador havia requereix de càlculs i algorismes complexos. Amb aquesta idea, com a primera opció, es va decidir utilitzar C++. L'elecció d'aquest software ve donada per la seva robustesa i a l'alta velocitat d'execució que un aplicació en temps real ha de complir. D'altra banda, ens estalvia la implementació de les diferents metodologies més comuns de l'àrea de la visió per computador, a través de la llibreria OpenCv. **Open Computer Vision Library** va ser inicialment creada per Intel, però ara és una llibreria de codi lliure. Posseeix la implementació de diferents algorismes de detecció d'objectes, detecció de moviment , seguiment de característiques, primitives de filtres transformadors i eines per al **Machine learning** entre d'altres. A més, OpenCv conté una col·lecció de primitives d'alt nivell i estructures de dades orientades a treballar amb imatges, optimitzades per a treballar en els actuals, i últimament comuns, processadors multi nucli.

Actualment, l'última versió d'OpenCv és multi plataforma , fet que ens dona independència d'implementació existint versions per a Linux, Mac OS i Windows.

El desenvolupament de l'aplicació incloïa certs apartats d'anàlisi i recerca, comportant la utilització d'un software àgil capaç de ser versàtil en quant a la confecció de diferents jocs de proves i tests , a cada un dels subsistemes del prototip que estem dissenyant. Matlab va ser la opció elegida, sent un software molt apropiat per a realitzar observacions del comportament de les dades basat un context, gracies a eines de grafisme i mineria de dades . Matlab és una bona eina per a la sintonització, o **tunning**,de certs paràmetres dels algorismes principals de visió per computador. Una altra avantatge es que pot entendre codi de C++ gracies al seu compilador MEX, permetent l'anàlisi de les implementacions de la llibreria OpenCv. El seu inconvenient es la seva baixa velocitat computacional, fet que associa aquest software a una etapa d'observació i anàlisi de les diferents situacions.
L'entorn de treball que utilitzarem per al desenvolupament dels nostres prototips és Visual Studio 2005. Aquest programa conté una col·lecció de menús de configuració molt intuïtius per al programador. També s'ha elegit a que agrupa compilador, i un editor de debugació molt complet.

6.3 HARDWARE

Una de les premisses importants de treball dels prototips, és la de poder executar-ne l'aplicació sobre computadors amb prestacions normals. Per a prestacions normals s'entén ordinadors domèstics des de fa uns tres anys fins ara (Juny del 2010). Les especificacions hardware amb les que s'ha creat els prototips són:

- Computador Intel Dual Cora 2Ghz
- 1 GB de memòria RAM
- Web Cam USB de 1,3 megapíxels

6.4 PROTIPUS I: Classificació discreta de pose basada en l'aprenentatge amb correcció a través de les característiques facials.

En bases generals, el mètode utilitzat correspon a un problema clàssic de detecció i classificació basat en tècniques de **Machine learning**. La diferencia, recau en la utilització de les característiques facials i la lògica espaciotemporal, per a corregir situacions, o estats, inacceptables al sistema, fet que comporta robustesa al procediment.

6.4.1 DESCRIPCIÓ DEL SISTEMA

Informació d'entrada: seqüència d'imatges ordenades en el temps corresponent a un subjecte.

Informació de sortida: discretització en temps real de la posició del cap en els següents estats:

- Frontal
- Dreta
- Esquerra
- Inferior
- Superior

Post-tractament de la informació de sortida: la posició del cap pot ser mostrada de forma continua, donant els valors dels angles *pitch, yaw* i *roll*, a través dels angles d'Euler (Annex II).

Explicació de la tasca que realitzen cada un dels elements que el composen el sistema detector de posició del cap:

- **Conjunt d'aprenentatge**: Col·lecció d'imatges classificades de forma manual, segons la posició del cap. El nombre de subagrupacions correspon al nombre d'estats a classificar. Cada subgrup conté 1200 imatges.

- **Imatge test:** Correspon a la informació purament d'entrada, sobre la qual analitzarem i donarem una discretització de la posició del cap.

- **Normalització de la imatge**: és un procés que es dona tant en les imatges del conjunt d'aprenentatge com en la imatge de test. Tota imatge del sistema ha de mantenir proporcions de mida, contrast i alineació. En quant a la mida les imatges, abans de tractar-les es posicionen sobre un marc de 125x125 píxels. Per al contrast, a totes les imatges se'ls aplica un filtre atenuador de brillantors i un altre gaussià difuminant-ho. El filtre atenuador provoca una normalització lumínica, amb l'objectiu d'evitar diferencies a l'espai de mostreig de la mateixa pose, per a diferents il·luminacions. El filtre gaussià ens eliminarà petites regions on hi pot existir soroll.

- **Mòdul detector de característiques facials i lògica del seu seguiment**: aquest apartat treballa sobre la imatge ja normalitzada en quant a mida i filtratge. Les característiques facials que volem trobar són: cara,parell d'ulls i nas. Per a la detecció utilitzarem Viola-Jones amb patrons d'ulls i nas extrets directament de la llibreria OpenCv. L'ordre d'extracció ha de ser el citat, degut a que la regió d'interès es va reduint, fet que comporta menys errors de detecció. En quant a la inicialització del sistema de detecció, establim una lògica valors fiables de posició. Donarem per bona una posició del cap si durant tres frames es troba la una cara centrada al marc de la imatge, en posicions relativament semblants. Una vegada trobada la cara, la posició dels reduirem la nostra regió d'interès al 60% superior on s'ha trobat el rostre. Per a la detecció dels ulls s'estableixen premisses tals com, l'angle entre el pla horitzontal de l'ull esquerra i l'ull dret no pot superar els 40 graus i aquests, han d'estar a separació mínima de 8 píxels. Aquests valors s'han donat per bons a base de realització de proves de situacions reals.
  Una vegada fet la inicialització, sobre les característiques facials de les successives imatges de la seqüència s'estabiliran les premisses de la lògica espaciotemporal. Si pel cas no en alguna iteració no es compleixen, el seguiment de la característica que ha incomplert la norma, a partir d'aquest moment el seguiment d'aquesta característica es farà a través d'Optical Flow. La regió sobre la qual treballarà l'**Optical Flow**, serà la regió de la iteració anterior, on abans es posicionava la característica i si es complia la lògica espaciotemporal.

  En la següent seqüència, el seguiment de la característica es farà a través d'**Optical Flow**, encara que paral·lelament s'utilitzarà també Viola-Jones. Si tant la posició de la característica a través d'Optical Flow, com la posició amb Viola-Jones tenen valor relativament coincidents, la característica tornarà a ser seguida a mitjançant Viola-Jones.

Una vegada tenim cercades les característiques facials de forma consistent procedim a l'alineació dels ulls. (figura 6.1)

Tot aquest sistema serveix per a corregir una posició incorrecta de les característiques facials. El seu principal avantatge és que es capaç d'auto corregir-se en temps d'execució.

- **Reducció de la dimensionalitat amb PCA i creació del conjunt d'aprenentatge:** Les imatges processades fins ara tenen una mida de 125x125 píxels, codificades en escala de grisos, tenim que cada imatge conté una informació de 125x125 valors. Classificar la informació de cada imatge sobre un espai de característiques amb aquesta **dimensionalitat**, seria una tasca feixuga i poc productiva a l'hora de trobar un espai de característiques excloent, capaç de poder dividir-ho en clústers amb l'objectiu de trobar-ne models plausibles. Per solucionar aquest problema es procedeix a la reducció de la dimensionalitat amb l'algorisme **PCA**. Analitzant la sortida dels **eigenvalues**, s'observa que recollint **els primers 120 valors dels eigenvectors**, obtenim un 95% d'informació representativa de la imatge. Amb aquest procediment hem reduït la dimensionalitat de 125x125 fins a 120. Un cop hem aplicat la reducció, projectem els valors del conjunt d'aprenentatge sobre els primers 120 **eigenvectors**, més la posició de les característiques facials de ulls i nas. Finalment guardem aquests vectors resultants, **etiquetats manualment** amb la posició del cap a la qual corresponen. També cal guardar la matriu de projecció, és a dir, els eigenvectors generats per en un futur projectar imatges de test. En aquesta aplicació, també s'ha guardat una matriu que conté els valors promig de les imatges del conjunt d'aprenentatge, amb la intenció d'excloure aquesta informació redundant de la projecció de les imatges. Fins aquí, ja tenim un conjunt d'aprenentatge etiquetat, normalitzat i reduït en quant a dimensionalitat.

- **Projecció a l'eigenspace de les imatges de test:** En aquest punt eliminem informació redundant de la imatge de test, a partir de la diferencia dels valors promigs produïts a la fase anterior. Amb les dades resultants projectem la imatge de test amb els **eigenvectors** produïts a la fase d'entrenament. Aquesta imatge de test, resultarà en dimensió 120 com les altres imatges del conjunt d'aprenentatge i la posició *(x,y)* de cada una de les característiques facials.

- **Mòdul classificador discret de Pose:** utilitzarem la combinació de classificadors binaris del tipus Suport Vector Machine. En concret la variant de combinació és **one against one**. L'elecció de l'estat final serà seleccionat mitjançant un sistema de vot.

	Frontal	Dreta	Esquerra	Inferior	Superior
Frontal					
Dreta					
Esquerra					
Inferior					
Superior					

Figura 6.1

- **Lògica de classificació de Pose:** ens trobem una altra vegada davant d'un bloc corrector, és a dir, corregeix estats impossibles. Per exemple si ens trobem davant d'una situació que el classificador ens produeix la sortida de dreta i l'acceptem com a valida, la següent imatge de la seqüència mai hauria ser esquerra. La transició d'estats possibles davant cert estat actual són els següents:

    - **Frontal:** tots els estats possibles
    - **Dreta:** Dreta, Frontal, Superior i Inferior
    - **Esquerra:** Esquerra, Frontal, Superior i Inferior
    - **Inferior:** Inferior, Frontal, Dreta i Esquerra
    - **Superior:** Superior, Frontal, Dreta i Esquerra

  Per dotar de certa robustesa al bloc, un estat no canviarà si no es produeix una transició a un estat possible durant com a mínim tres vegades consecutives.

  Altre aportació que proporciona rigidesa al sistema, consisteix en que com es tenen controlades la posició de les característiques facials de cada imatge de la seqüència, si es produeix una petició de canvi de transició però s'observa, que sobre el conjunt de característiques no s'ha produït un moviment o desplaçament considerable sobre l'eix vertical o horitzontal, ometrem la petició de canvi d'estat.

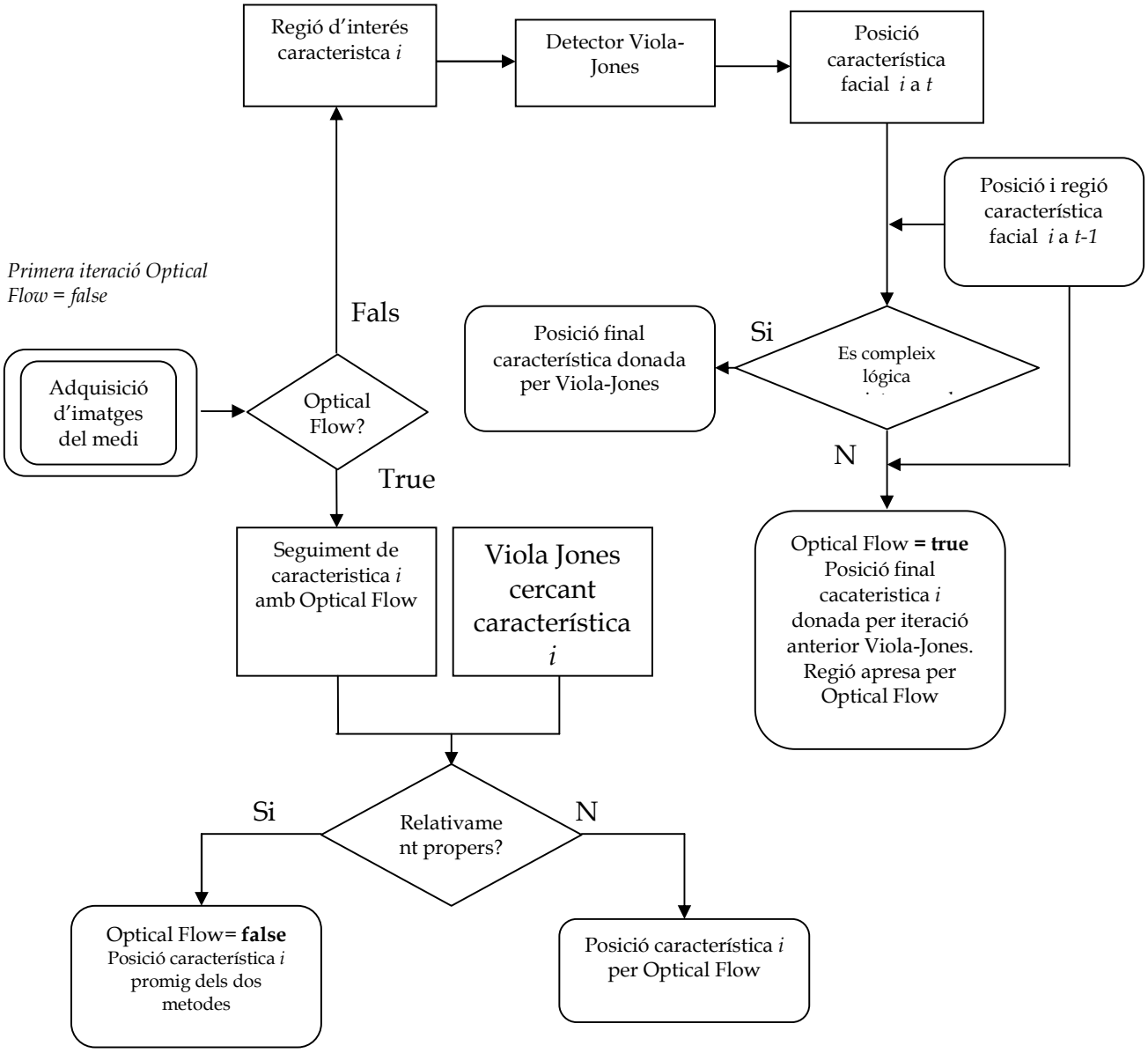
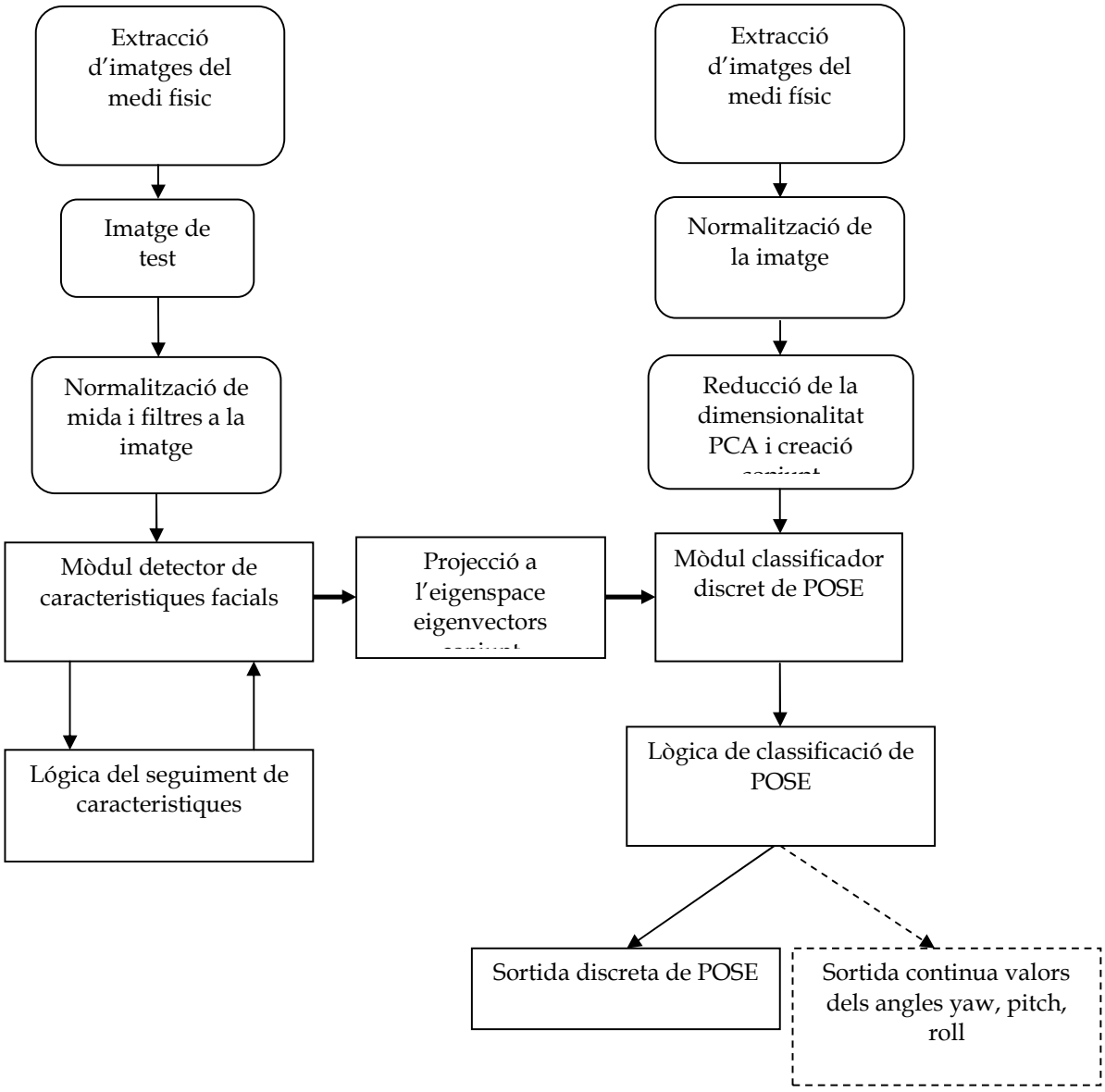

Regió d'interés caracteristca *i*
Detector Viola-Jones
Posició característica facial *i* a *t*
Posició i regió característica facial *i* a *t-1*
Primera iteració Optical Flow = false
Fals
Posició final característica donada per Viola-Jones
Si
Es compleix lógica
Adquisició d'imatges del medi
Optical Flow?
N
True
Seguiment de caracteristica *i* amb Optical Flow
Viola Jones cercant característica *i*
Optical Flow **= true** Posició final cacateristica *i* donada per iteració anterior Viola-Jones. Regió apresa per Optical Flow
Si
Relativament propers?
N
Optical Flow= **false** Posició característica *i* promig dels dos metodes
Posició característica *i* per Optical Flow

Figura 6.1

6.4.2 DIAGRAMA DE FUNCIONAMENT



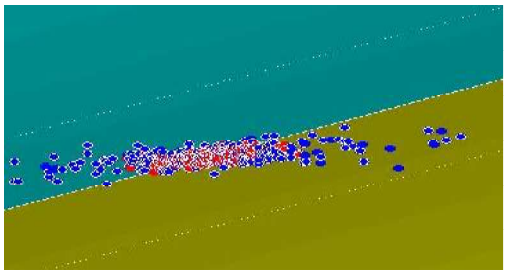
Analitzant el sistema s'ha observat que la robustesa d'aquest és molt depenent de la bona localització de les característiques facials. Degut a que l'entorn de treball on s'aplica el prototipus, és un entorn molt poc controlat existeixen sovint errors de detecció de característiques facials. A vegades aquest error encara es pronuncia, i és propaga, quan es produeix l'alineació dels ulls. S'ha decidit eliminar aquest procés de normalització, fet que

provocarà una discriminació menor en quant a la classificació de la pose, perjudicant la robustesa del classificador. El conjunt d'aprenentatge també es fortament decisiu a l'hora de realitzar la classificació, i per tant, obtenir un model plausible.

6.4.3 Visualització dels classificadors binaris Suport Vector Machine

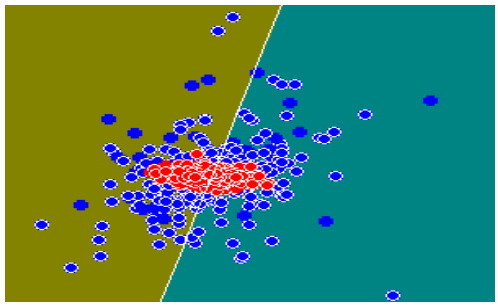
- **Frontal VS Dreta**

Màxima separació entre conjunts	0.290858
Marge de l' hiperplà	3.708427
Percentatge de classificació sobre 1200 imatges de cada conjunt	81.5%
Punts Blaus	Frontals
Punts Vermells	Dreta



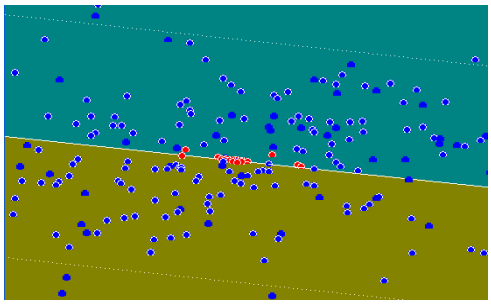
- **Frontal VS Esquerra**

Màxima separació entre conjunts	1.221193
Marge d'hiperplà	1.809830
Percentatge de classificació sobre 1200 imatges de cada conjunt	79.2%
Punts Blaus	Esquerra
Punts Vermells	Frontal



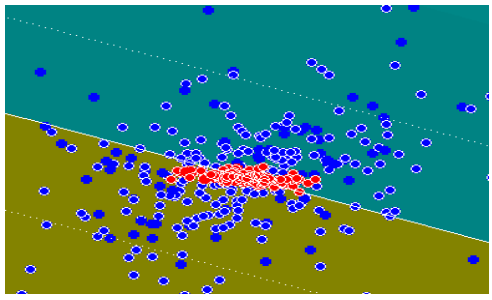
- **Frontal VS Superior**

Màxima separació entre conjunts	0.376029
Marge d'hiperplà	3.261513
Percentatge de classificació correcta sobre 1200 imatges de cada conjunt	74.8%
Punts Blaus	Frontals
Punts Vermells	Superiors



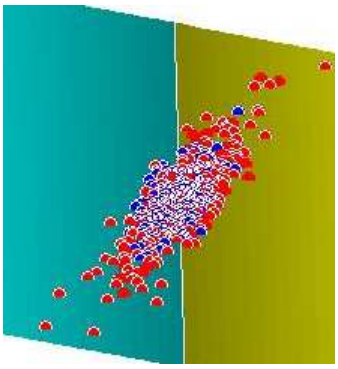
- **Frontal VS Inferior**

Màxima separació entre conjunts	0.615739
Marge d'hiperplà	2.548776
Percentatge de classificació correcta sobre 1200 imatges de cada conjunt	74.2%
Punts Blaus	Frontals
Punts Vermells	Inferiors



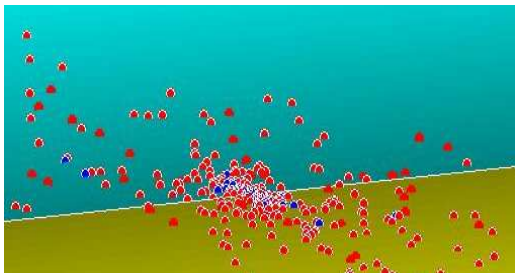
- **Dreta VS Esquerra**

Màxima separació entre conjunts	0.345281
Marge de l'	3.403640
Percentatge de classificació correcta sobre 1200 imatges de cada conjunt	89.2%
Punts Blaus	Esquerra
Punts Vermells	Dretes



- **Inferior VS Superior**

Màxima separació entre conjunts	3.465342
Marge de l'	1.074378
Percentatge de classificació correcta sobre 1200 imatges de cada conjunt	91.5%
Punts Blaus	Superiors
Punts Vermells	Inferiors



6.5 PROTOTIPUS II: Classificació discreta de la pose utilitzant models adaptatius i deformables basats en Active Appereance Model

Aquest prototip es basa en una metodologia molt fidel a la proposada per Tim Cootes amb el seu algorisme **Active Appereance Models**. L'objectiu es crear tres models seguint les propietats d'adaptativa i deformables de l'algorisme **AAM**. Aquests models pertanyen a una composició combinada de paràmetres de forma i aparença de cares frontals, girades a la dreta i girades a l'esquerra.

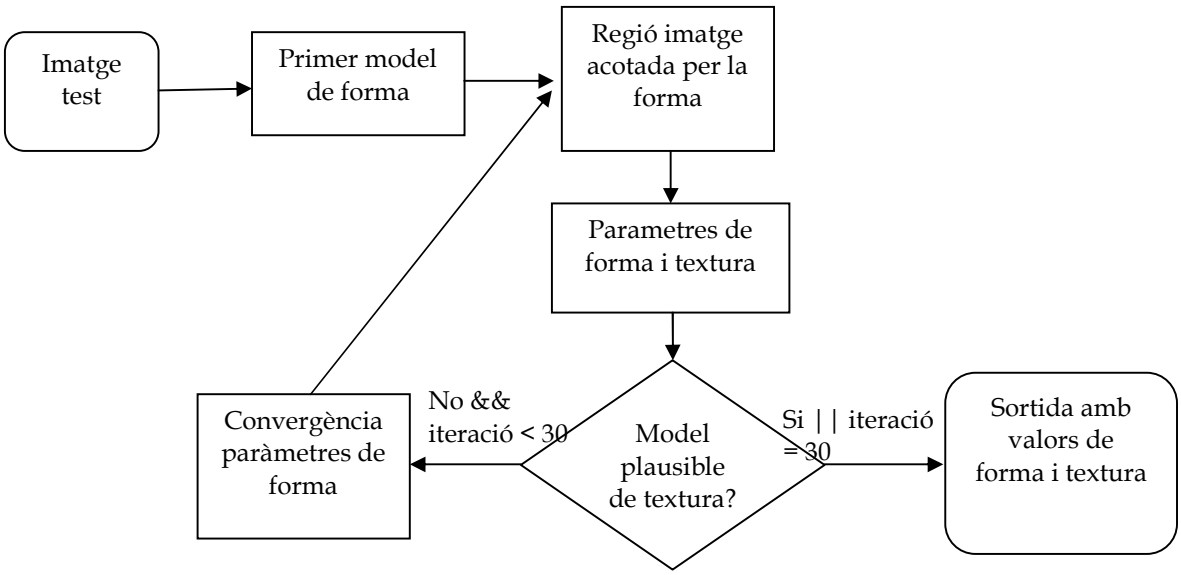
Primerament, s'ha formulat la hipòtesi que els diferents models donats per la combinació de paràmetres de forma i aparença, es situen espacialment de forma diferenciable i fàcils d'agrupar en clústers. Si aconseguim aquest fet, tornarem a utilitzar tècniques de **Machine learning** per al procés de classificació discreta de la pose.

En quant a l'elaboració del conjunt d'entrament existeix una primera fase manual de posicionament dels **landmarks** sobre les imatges. En aquesta fase també s'etiquetarà cada imatge amb el model de posició del cap que pertany.

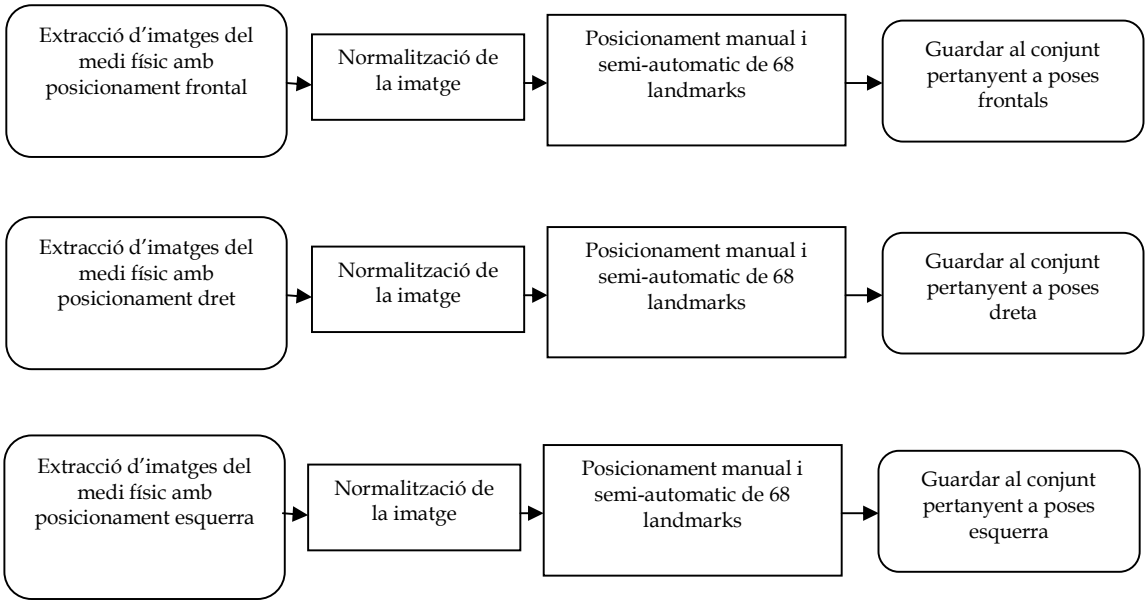
Existeix una segona fase en quant al tractament de les dades d'aprenentatge. Consisteix en la ja comentada, i coneguda, fase de reducció de la dimensionalitat de les dades amb PCA. Amb la finalitat de reunir un 95% d'informació representativa de forma i aparença, ens quedem amb els 8 **eigenvectors** que produeixen un alt valor **d'eigenvalue** en quant a la forma, i 30 per a representar la textura.

Una de les principals diferencies amb el prototip anterior, es que la imatge test no es projectarà en un sol espai de característiques, sinó es projectarà en cada un dels models creats. El coeficient de fiabilitat de pertànyer a un clúster determinat,vindrà donat pel valor absolut de la distancia euclidiana de la projecció final del test al centre de gravetat del clúster.

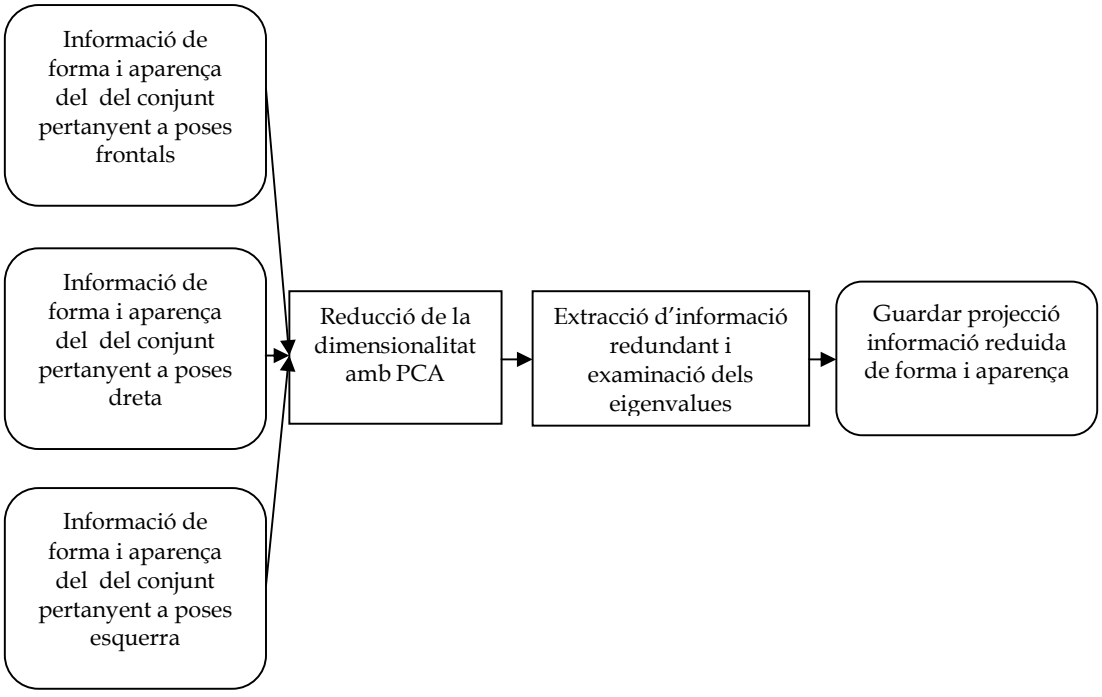
Durant el procés de projecció de la imatge de test sobre un espai de característiques determinat, s'inicia un procediment iteratiu de convergència. La raó de convergència es un coeficient d'error calculat en funció de les diferencies d'aparença, o textura, de la imatge de test sobre el conjunt d'aprenentatge. En aquest procés també intervenen les característiques de forma del mateix model on estem avaluant la convergència. Dit procés finalitzarà quan s'estableixi un valor llindar acceptable sobre les característiques de textura, o al realitzar un màxim de 30 iteracions. Els paràmetres de textura vindran donats per la regió acotada pels paràmetres de forma. Alhora que intentem obtenir un model plausible a través de les característiques de forma, també es produeix un procés de convergència dels paràmetres de forma, tendint cap a valors plausibles del conjunt d'aprenentatge.



6.5.1 Fase I: Pre-procesament d'imatges del conjunt d'entrenament



6.5.2 Fase II: Anàlisi i reducció de la dimensionalitat en característiques de forma i aparença



6.5.3 PROJECCIÓ VISUAL DE LES CARACTERISTIQUES DE FORMA

Nombre d' imatges de cada conjunt: 155
Nombre de **landmarks** sobre cada imatge: 68
Percentatge de representació 98.45%
Descripció dels 8 primers Eigenvalues:

	1	2	3	4	5	6	7	8
Valor Eigenvalue	376.6830	282.6133	10.1954	4.6719	2.3683	1.3670	0.4032	0.3147
Representació del valor %	55.3917%	41.5587%	1.4992%	0.6870%	0.3483%	0.2010%	0.0593%	0.0463%
Representació acumulada %	55.3917%	96.9504%	98.4496%	99.1366%	99.4849%	99.6859%	99.7452%	99.7915%

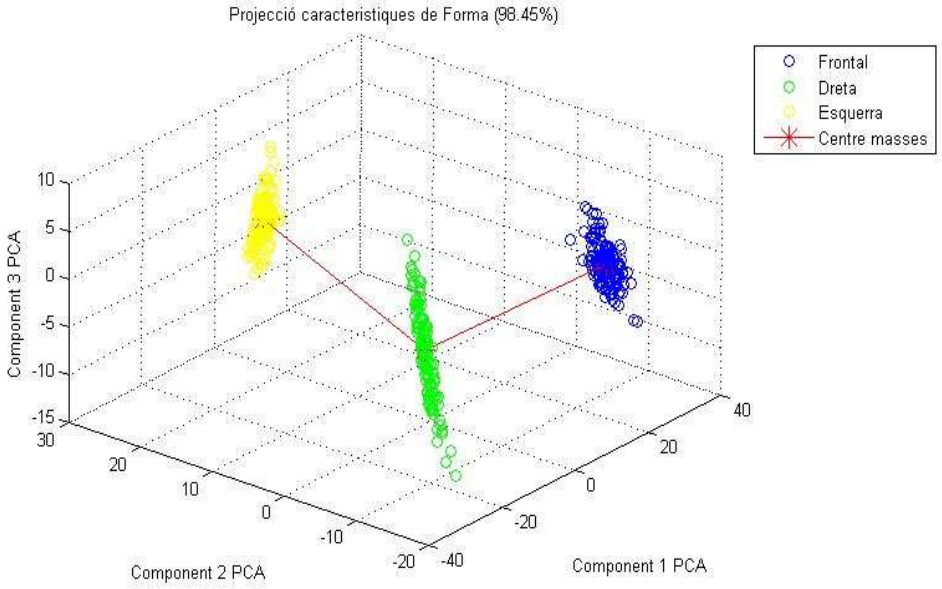
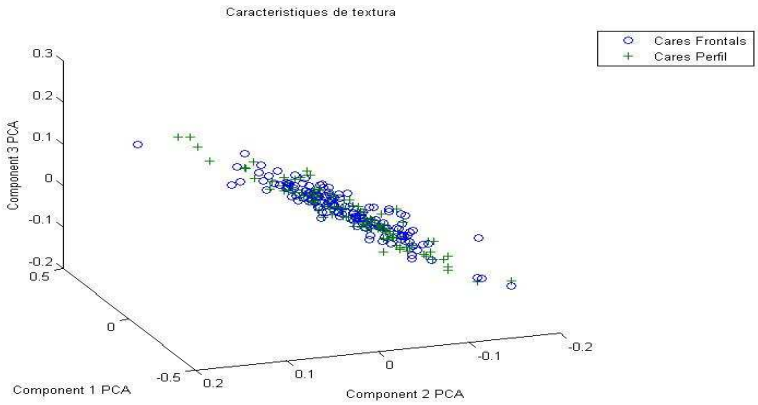


Figura 6.2

Analitzant el gràfic anterior, s'observen tres clústers, corresponent a cada posició del, ben diferenciats. Aquesta separació sobre el mateix espai de projecció, ens indica que els paràmetres contenen valors molt discriminants, els quals donaran robustesa al sistema de classificació.

6.5.4 PROJECCIÓ VISUAL DE LES CARACTERISTIQUES DE TEXTURA

Nombre de imatges de cada conjunt: 155
Nombre de píxels de cada regió: 21681
Percentatge de representació 86%



En canvi, en quant als paràmetres de textura, tant els valors pertanyents cares frontals, com els que pertanyen a cares de perfil, tenen un comportament similar al projectar-les en un mateix espai de característiques. Això és degut a que la textura d'una cara de perfil es semblant a la d'una cara frontal, ja que en ambdós casos la textura predominant és la pell. També s'observa una oscil·lació en senti decreixent sobre l'eix vertical, aquesta rang de valors pertany a diverses intensitats d'una textura similar, proporcionant variabilitat a les textures que considerarem plausibles.

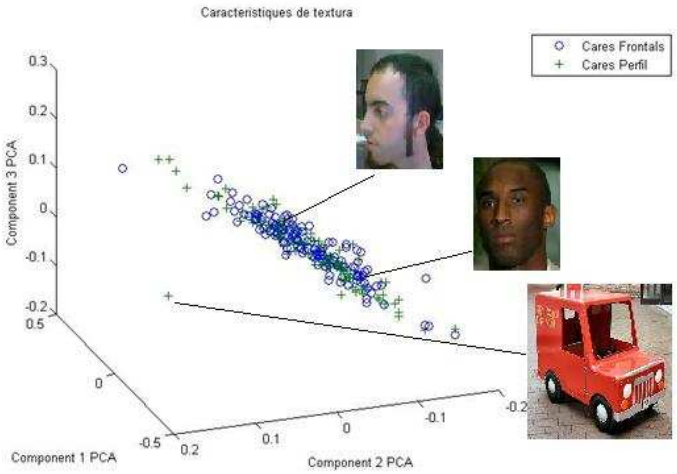
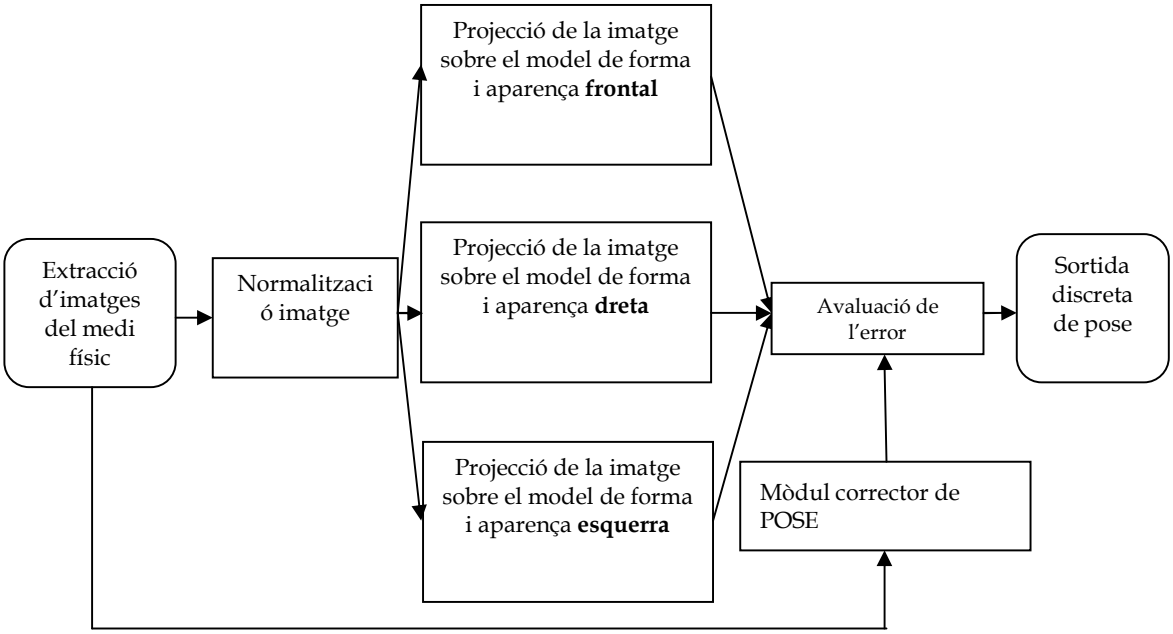


Figura 6.4

6.5.5 Fase III: Test

Quan el prototipus s'executa amb una seqüència d'imatges com a entrada del sistema, durant la primera iteració es realitza la detecció de cares mitjançant Viola-Jones. Per a filtrar els possibles falsos positius s'estableix la lògica espacial, on només existeix un subjecte i que sol estar centrat al marc de la imatge.

La normalització de la imatge afecta característiques de mida, contrast i soroll. Les transformacions associades consisteixen en un escalat de la imatge per obtenir imatges de mida 125x125 píxels, l'aplicació d'un filtre gaussià per eliminar petits sorolls i una equalització del contrast de la imatge.



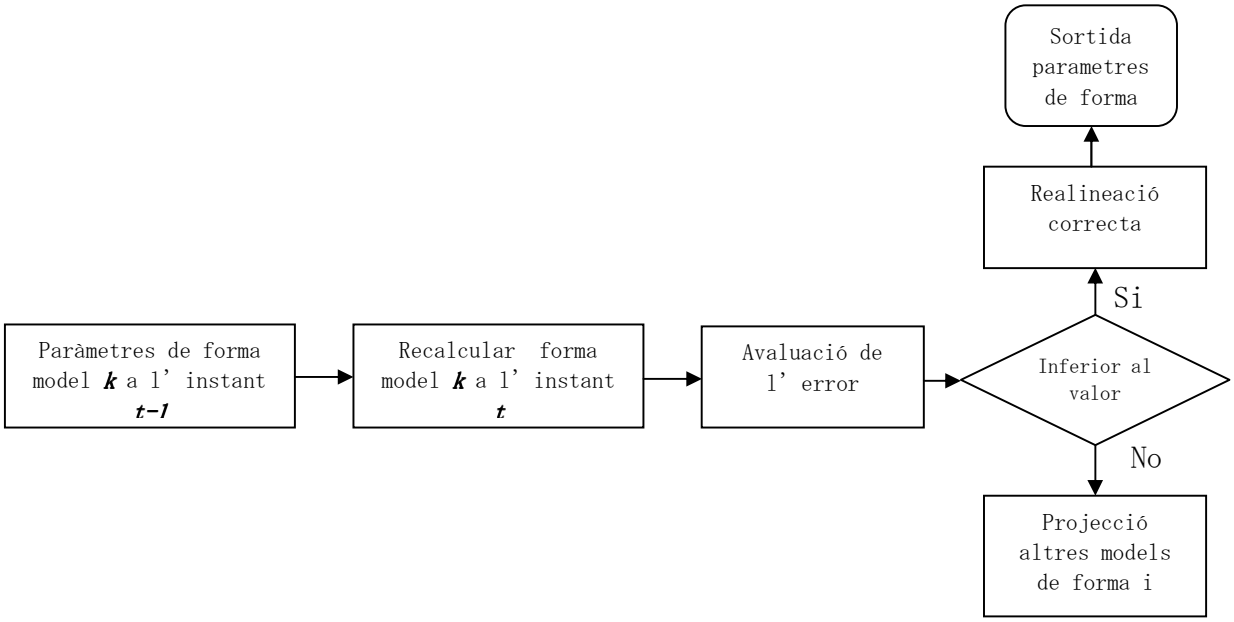
La sortida del sistema obtindrà els valors discrets de Frontal, Dreta o Esquerra. També podem donar una sortida continua en quant a valors dels angles de pitch, roll i yaw del cap, amb la utilització dels angles d'Euler tal i com es mostra a l'annex II.

6.5.6 MÒDUL CORRECTOR DE POSE

Per dotar de certa robustesa al sistema i estalviar cost computacional, ja que es tracta d'una aplicació en temps real, utilitzem un mòdul corrector d'estats impossibles. Aquest mòdul ometrà classificacions impossibles donada una transició f a l'instant t. Tornant a la lògica espaciotemporal , si en un instant determinat t, a l'instant següent $t+1$ només ha de permetre transicions suavitzadores i espacialment veïnes, ometent la resta. Observem la taula de transicions:

Estat a ***t***	Sortida del classificador ***t+1***	Estat final corregit ***t+1***
Frontal	Frontal	Frontal
Frontal	Dreta	Dreta
Frontal	Esquerra	Esquerra
Dreta	Frontal	Frontal
Dreta	Dreta	Dreta
Dreta	Esquerra	Dreta
Esquerra	Frontal	Frontal
Esquerra	Esquerra	Esquerra
Esquerra	Dreta	Esquerra

Una altra aportació d'aquest bloc és l'estalvi de càlcul, produint un decrement del cost computacional. Aquest subprocés ens permet no haver de projectar la imatge de test sobre els tres models a cada instant ***t***, si no s'han produït grans canvis a l'escena. Per exemple, un subjecte es sol mantenir en un dels estats durant més d'una unitat de temps. Per això entre dues imatges d'una seqüència, no cal projectar cada imatge sobre els tres models, sinó amb una variabilitat de forma respecte els paràmetres de forma de la imatge anterior a l'instant ***t-1***, produirem un projecció plausible dels nous paràmetres de forma, corresponents a l'estat actual a l'instant ***t***.



7. RESULTATS

En aquest apartat mostrarem els resultants obtinguts aplicant les diferents aplicacions sobre diferents conjunts de test. Les principals línies d'avaluació dels prototipus seran les següents:

- Eficàcia
- Rendiment
- Abast

Per eficàcia entenem al percentatge d'encert del prototipus relatiu a la nostra formulació teòrica. Durant aquest primer anàlisi els conjunts de test seran ideals, dirigits al context per al qual ha sigut desenvolupada l'aplicació, a través de les diferents hipòtesis de correcte funcionament. Dins d'aquest apartat es reflexa la fidelitat de la part pràctica amb la teòrica, alhora també s'estudien tant les mancances de la formulació teòrica com les limitacions de la implementació pràctica.

Tal i com es va plantejar de bon principi, una de les premisses majors d'aquesta implementació pràctica es la seva funcionalitat en temps real. A l'apartat de rendiment s'examinarà tant la complexitat espacial com temporal per a detallar el seu comportament computacional. El computador amb el qual s'han avaluat dites característiques sempre es manté constant, ja que es considera que compleix els requisits mínims de computació. Les especificacions hardware es van detallar a l'apartat anterior.

Pel que fa l'anàlisi de l'abast, es realitzaran diferents proves sobre els prototipus amb conjunts de test molt diversos, amb la intenció d'acotar el seu espai de funcionament. Amb aquesta acotació anomenarem el seu context de treball amb una òptima eficàcia, nombrant també contextos on no és una bona pràctica utilitzar aquests prototips.

L'anàlisi d'aquestes propietats ens permetrà donar una visió global del funcionament dels prototipus, contribuint a obrir noves línies de d'investigació i modificació dels mètodes, per tal de millorar els resultats obtinguts. Els resultats ens induiran a conclusions,les quals ens ajudaran a reflexionar sobre la possibilitat d'utilització de dites tècniques i mètodes a diferents aplicacions d'enginyeria, com per exemple aplicacions comercials, industrials o científiques.

7.1 PROTOTIP I

Abans d'avaluar directament la sortida de tot el sistema, dividirem en parts l'anàlisi de resultats.

- Procés de detecció de característiques facials:

La detecció de característiques facials és un element fonamental del sistema, avaluem primer la seva consistència.

Resultats de la detecció de característiques cap, ulls i nas:

Test utilitzat: seqüència de vídeo de 4302 frames.

Contingut del test: Un sol subjecte centre al centre del marc de la imatge. Moviments de rotació, esquerra, dreta, superior i frontal. Distanciament i apropament. Respectant la lògica espacio temporal, sense moviments bruscos. Background heterogeni.

Figura 7.1

Nombre de deteccions fiables de rostre per Viola-Jones: 3985/4302 frames
Nombre de correccions amb lògica espaciotemporal del rostre: 317/4302 frames
Nombre de deteccions fiables d'ulls per Viola-Jones: 2796/4302 frames
Nombre de correccions amb optical flow d'ulls: 1506/4302 frames
Nombre de deteccions fiables de nas amb Viola-Jones: 2065/4302 frames
Nombre de correccions amb lògica espaciotemporal del nas: 2237/4302 frames.

Nombre de deteccions correctes omitint detecció del nas (supervisat manualment): 3571/4302
Nombre de deteccions correctes, amb detecció de nas (supervisat manualment): 2925/4302

Percentatge de fiabilitat sense detecció del nas: 83%
Percentatge de fiabilitat amb detecció del nas: 69%
Test utilitzat: seqüència de vídeo de 2641 frames.

Contingut del test: Un sol subjecte centre al centre del marc de la imatge. Moviments de rotació, esquerra, dreta, superior i frontal. Distanciament i apropament. Respectant la lògica

espacio temporal, sense moviments bruscos. Background homogeni. No s'utilitza la detecció de nas.

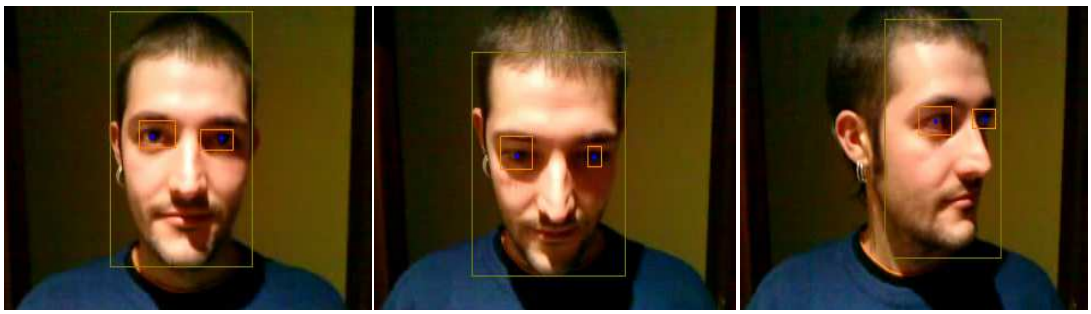


Figura 7.2

Nombre de deteccions fiables de rostre per Viola-Jones: 2192/2641 frames
Nombre de correccions amb lògica espaciotemporal del rostre: 449/2641 frames
Nombre de deteccions fiables d'ulls per Viola-Jones: 2059/2641 frames
Nombre de correccions amb optical flow d'ulls: 1338/2641 frames

Nombre de deteccions correctes omitint detecció del nas (supervisat manualment): 2033/2641
Percentatge de fiabilitat sense detecció del nas: 77%

Test utilitzat: seqüència de vídeo de 3354 frames.

Contingut del test: Un sol subjecte centre al centre del marc de la imatge. Moviments de rotació, esquerra, dreta, superior i frontal. Distanciament i apropament. Respectant la lògica espacio temporal, sense moviments bruscos. Background homogènia. Amb detecció de nas.

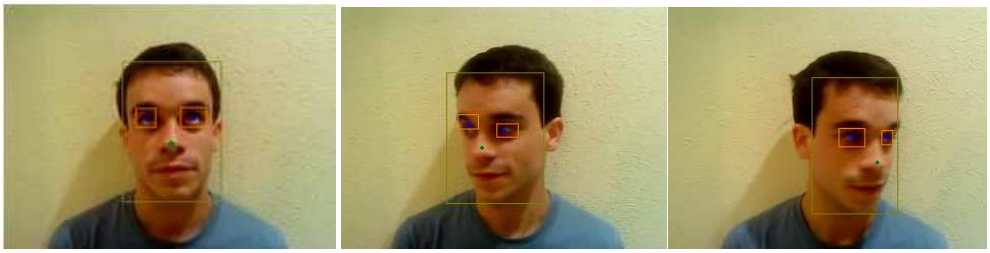


Figura 7.3

Nombre de deteccions fiables de rostre per Viola-Jones: 2817/3354 frames
Nombre de correccions amb lògica espaciotemporal del rostre: 537/3354 frames
Nombre de deteccions fiables d'ulls per Viola-Jones: 2347/3354 frames
Nombre de correccions amb optical flow d'ulls: 1007/3354 frames
Nombre de deteccions fiables de nas amb Viola-Jones: 2080/3354 frames

Nombre de correccions amb lògica espaciotemporal del nas: 1274/3354 frames.

Nombre de deteccions correctes omitint detecció del nas (supervisat manualment): 2583/3354
Nombre de deteccions correctes, amb detecció de nas (supervisat manualment): 2180/3354

Percentatge de fiabilitat sense detecció del nas: 77%
Percentatge de fiabilitat amb detecció del nas: 65%

Degut a la caiguda del percentatge de fiabilitat observada quan s'utilitza la detecció del nas, les imatges de test seran reduïdes als 120 valors a través del PCA i la posició dels ulls, ometent així la posició del nas. Altres característiques facial com el rostre o els ulls, són més fàcils de detectar davant les deformacions de rotació que ofereix el subjecte.

- Procés de classificació

Test utilitzat: seqüència de vídeo de 2485 frames

Contingut del test: Un sol subjecte centre al centre del marc de la imatge. Moviments de rotació, esquerra, dreta, superior i frontal. Distanciament i apropament. Respectant la lògica espacio temporal, sense moviments bruscos. Background heterogeni. S'omet la detecció del nas.

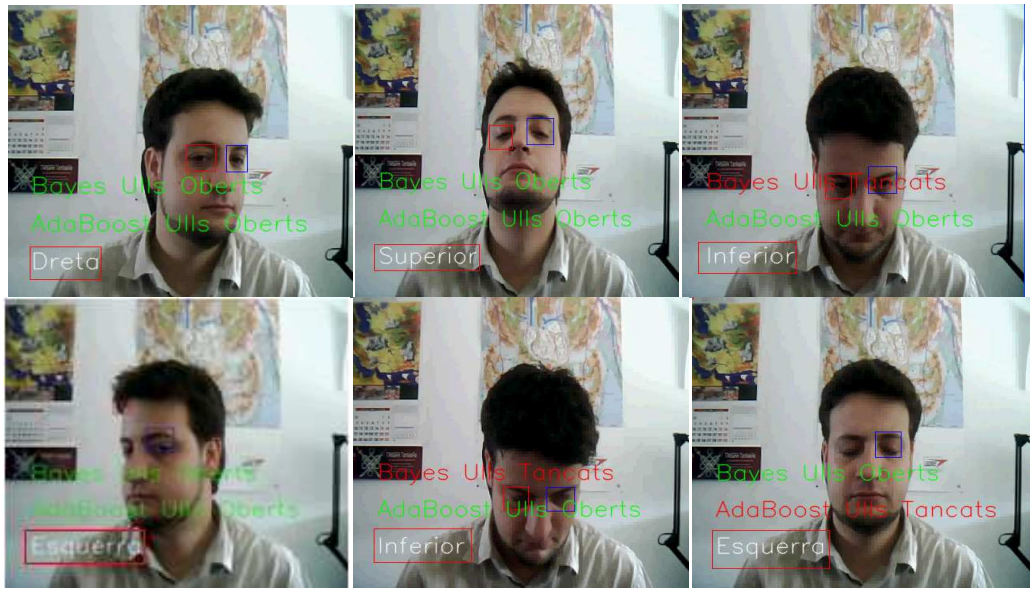


Figura 7.4

Nombre d'imatges ben classificades, supervisió manual: 1689/2485 frames
Percentatge d'encert: 68%

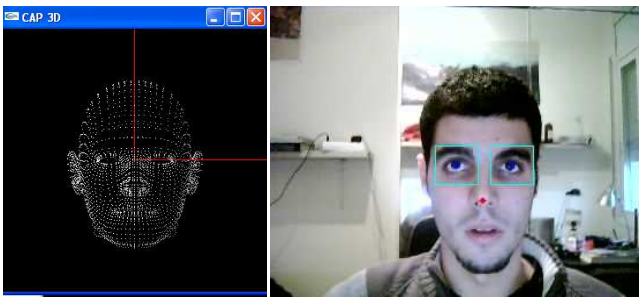
Test utilitzat: seqüència de vídeo de 5826 frames

Contingut del test: Un sol subjecte centre al centre del marc de la imatge. Moviments de rotació, esquerra, dreta, superior i frontal. Distanciament i apropament. Respectant la lògica espacio temporal, sense moviments bruscos. Background heterogeni. S'omet la detecció del nas.

Figura 7.5

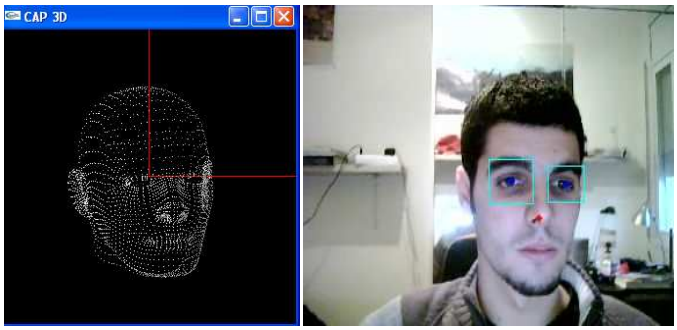
Nombre d'imatges ben classificades, supervisió manual: 5011/5826 frames
Percentatge d'encert: 86%

Sortida continua amb valors *pitch yaw* i *roll*



Pitch = 0
Yaw = 0
Roll = 0

Figura 7.6



Pitch = 18.44º
Roll = -6.0385
Yaw = 8.5560

Figura 7.7

Cost Computacional

La complexitat espacial de l'aplicació en temps d' execució consta de:

- Imatge real extreta de la càmera, 125x125x3 píxels
- Imatge actual de 125x125 píxels escala de grisos
- Imatge anterior de 125x125 píxels escala de grisos
- 200 floats per a l'ús d'optical flow en l'ull dret
- 200 floats per a l'ús d'optical flow en l'ull esquerra
- 5 imatges de tamany 125x125 per aplicar filtres gaussians.
- 12 estructures que contenen dos enters per a guardar la posició de les característiques facials

Aquest es el tamany màxim que es pot carregar en memòria durant el temps d'execució de l'aplicació.

La complexitat temporal de l'aplicació d'execució es basa en:

- Operació d'extracció imatge de la web-cam i normalització de mida → 37 ms
- Aplicació de filtre gaussià i detecció del rostre sobre una espai 125x125 → 57.6 ms
- Detecció d'ulls sobre la regió acotada i lògica correctiva → 41.3 ms
- Inicialització d'Optical Flow amb 200 floats per ull → 82 ms
- Seguiment amb Optical Flow i Viola-Jones a la vegada → 94 ms aprox.
- Projecció de la imatge a l'espai de característiques → 71 ms
- Classificació amb one-agains-ones de Suport Vector Machine → 98 ms aprox.

Temps per una imatge de la seqüència sense la utilització d'optical flow: 305 ms (temps mínim)
Temps per una imatge on s'inicialitza l'optical flow: 386 ms (temps mitjà)
Temps per una imatge on s'empra Viola-Jones i optical flow en paral·lel: 398.9 ms (temps

màxim)

El temps màxim correspon a quan es produeix el seguiment de 200 característiques per ull mitjançant l'optical flow.

Figura 7.8

7.2 PROTOTIPUS II

Sobre aquesta aplicació aplicarem un seguit de conjunts de test per a mostrar models plausibles a partir de la visualització de malles frontals i de perfils.

Test I: Aplicació d'un conjunt d'imatges normalitzades amb textures ben diferenciades al model frontal:

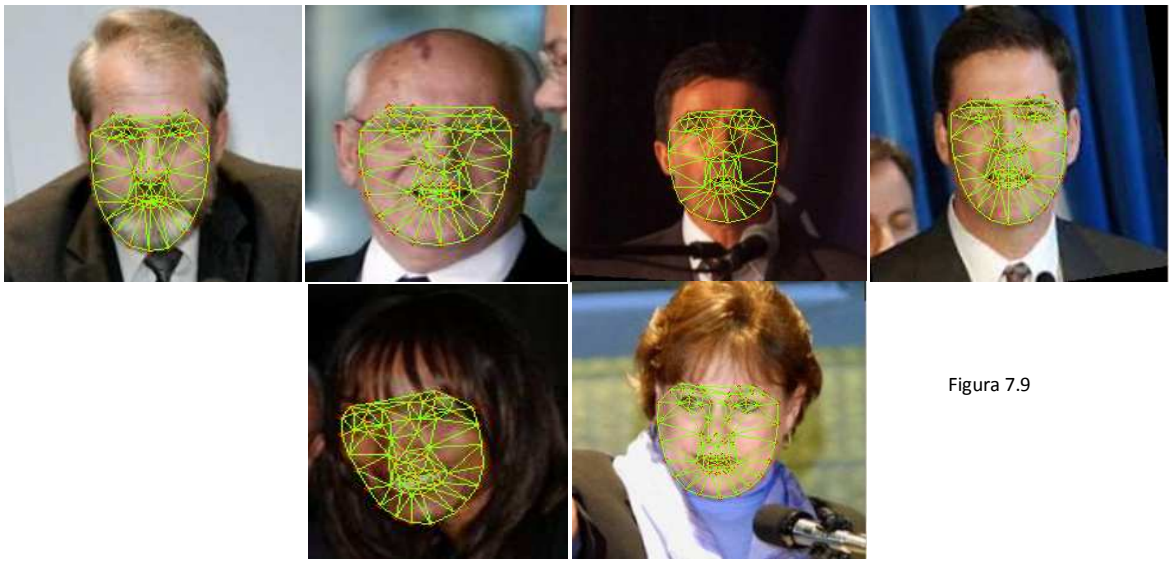


Figura 7.9

Aquest test conté 5000 imatges. Si existeix la totalitat del rostre el reconeixement és molt eficient. Només s'aprecien errors en quant al matching de punts molts concrets de característiques facials com poden ser els llavis o les celles, o bé quan existeixen oclusions, encara que en alguns casos d'una una aproximació prou bona. Però donant una visió global

al problema al conjunt de característiques facials, podem dir que el matching és correcte en un 93% de les imatges frontals.

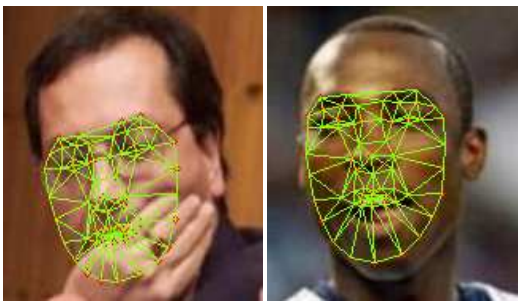


Figura 7.10

Correcció errors d'oclusió i error a l'ajust dels punts dels llavis

Test II: Aplicació d'un conjunt d'imatges normalitzades amb textures ben diferenciades al model esquerre:

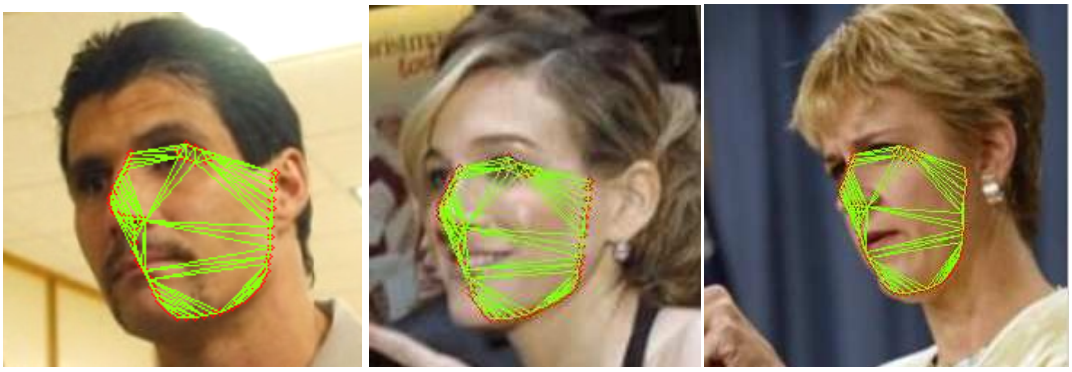


Figura 7.11

Aquest conjunt és de 2000 imatges. Tal i com va ocórrer a l'anterior test, si la imatge facial és clara hi ha una bona associació que comporta un bon encaixament de la malla. En alguns casos no es localitza bé l'ull dret. Sobre les 2000 imatges hem trobat una taxa d'encert del 89%.

Test III: Aplicació d'un conjunt d'imatges normalitzades amb textures ben diferenciades al model dreta:

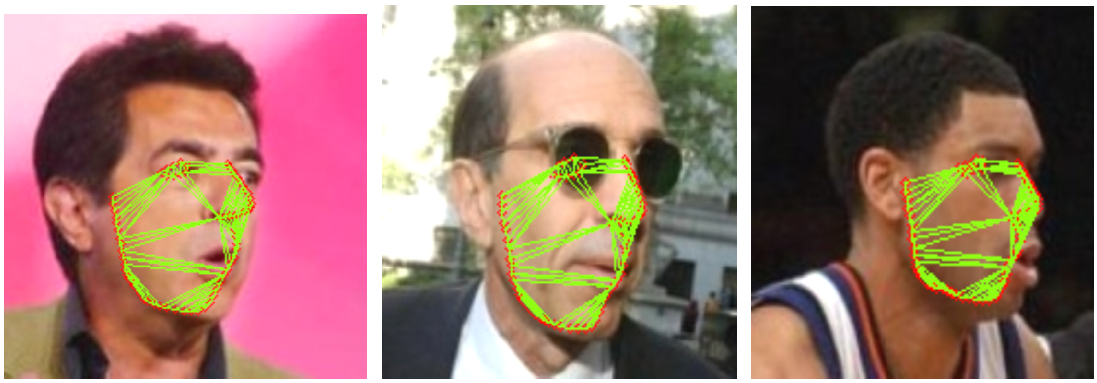


Figura 7.12

Es compleix la hipòtesi de que la taxa d'encert ha de ser molt similar, ja que el conjunt d'entrenament ha sigut el mateix que el model anterior, exercint una transformació de 180 graus a l'eix horitzontal, és a dir adquirint el simètric. Aquest test conté les mateixes 2000 imatges anteriors, obtenint la seva imatge simètrica.

Tests finals: Cercar el model més a través d'una seqüència ordenada d'imatges proporcionades per la web-cam

Figura 7.13

Seqüència de 2655 frames.
Correspondència exacta amb el model: 2497 frames
Percentatge d'encert 95%

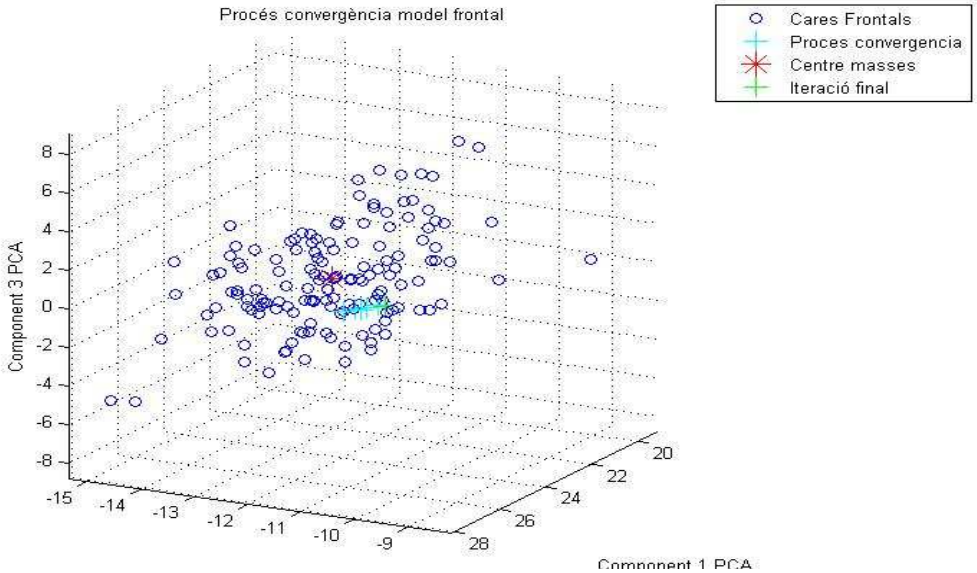
Figura 7.14

Seqüència de 3478 frames amb molta brillantor.
Correspondència exacta amb el model: 3269 frames
Percentatge d'encert 94%

Visualització del procés de convergència:


Figura 7.15

Visualització procés de convergència amb èxit ja que la imatge convergeix cap al conjunt d'imatges frontals.



Nombre d'iteracions fins la convergència de textura: 11
Distància del test al centre de masses: 1.451

Figura 7.17

Projecció de la mateixa imatge al model dret:

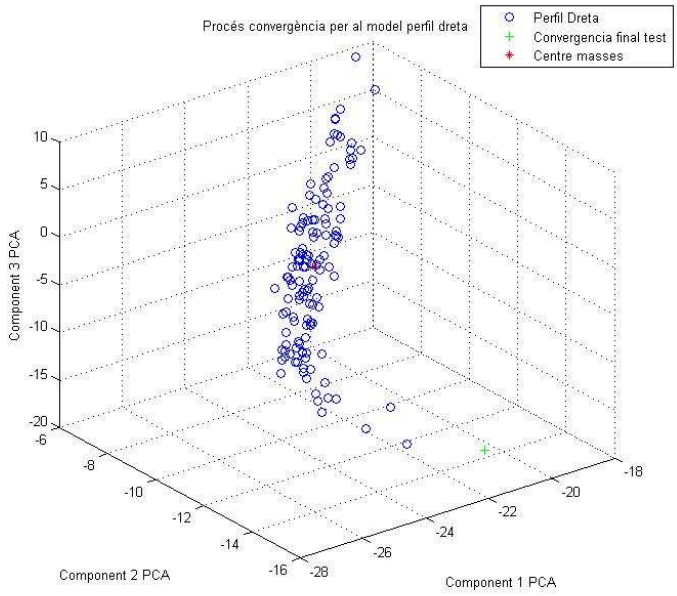


Figura 7.18

Nombre d'iteracions fins convergència amb la textura: 22
Distancia del test al centre de masses: 18.5861

Observem amb claredat que la imatge (punt verd) s'allunya del conjunt.

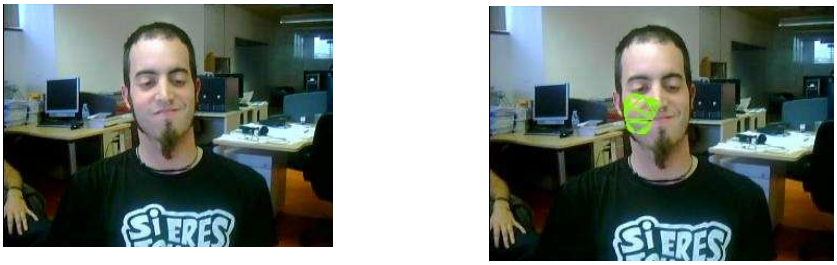


Figura 7.19

Temps d'execució:

Operació d'extracció imatge de la web-cam i normalització de mida → 37 ms
Càlcul d'un model →221 ms
Càlcul dels tres models i elecció → 648 ms
Posicionament i visualització malla → 74 ms
Temps d'execució amb re alineació → 332 temps mínim
Temps d'execució amb cerca de correspondència → 759 ms

8. CONCLUSIONS I TREBALL FUTUR:

La capacitat d'estimar la posició del cap d'una persona respecte un punt de vista, no és exclusivament un problema de detecció, com pot ser una aplicació de detecció o seguiment d' objectes marcats per un cert patró.

Es considera que es un problema tan complex, que no només cal fer ús de les tècniques i mètodes propis de l'àrea de visió per computador. Encara que aquestes siguin estrictament necessàries per al procés d'extracció i seguiment de la informació, cal dotar al sistema de certa intel·ligència per tractar i observar el comportament de la informació extreta.

Una part fonamental del sistema d'estimació és la classificació de característiques. El sistema classificador ens condueix directament als valors de sortida, i per tant l'estimació de la posició del cap. La classificació utilitzada sobre els prototips es basen en tècniques de **Machine learning**. Degut a la utilització de classificadors supervisats, el grau d'eficàcia dels classificadors està fortament relacionat amb la consistència del conjunt d'aprenentatge. Es per això que s'ha d'elaborar un estudi exhaustiu de l'abast i entorn de treball del problema a tractar, per a fer una modelització fidel del conjunt d'aprenentatge, amb l'objectiu que existeixi semblança amb els models que ens trobarem en un futur. Es per aquest motiu que s'ha fet especial atenció en les tècniques de normalització i eliminació d'informació redundant i soroll. L'obtenció d'un espai de característiques fortament diferenciat, i distanciat entre els diferents clústers, és una garantia d'èxit en quant al procés de classificació per a mètodes supervisats.

Altrament, l'elaboració d'un estudi del context on treballarà l'aplicació es fonamental per a l'acotació del problema. L'acotació no s'ha de veure com una limitació del problema que estem tractant, sinó com la definició del context i abast on el nostre sistema treballarà de forma robusta. Un sistema amb un alt grau de llibertat de detecció i classificació comporta un espai d'anàlisi més ampli, que a efectes pràctics, la seva implementació pot ser intractable, produint un sistema poc robust.

Coneixent l'abast i context de l'aplicació es pot dotar aquest de certa intel·ligència, amb l'objectiu de corregir possibles situacions impossibles. Aquesta intel·ligència es proporcionada a la aplicació amb la implementació d'un lògica externa, formulada a partir de l'observació del context de funcionament. Sobre els prototips, la lògica formulada ha sigut la referent a la espaciotemporal. Es veritat que redueix l'abast de l'aplicació, per exemple no es poden localitzar dos subjectes a la vegada, però ens permet corregir estats que sabem per endavant que són impossibles que es donin. Aquesta reducció de l'abast té per contra contribuir a la robustesa de l'aplicació.

Arribem a un primera conclusió, de que el percentatge d'èxit o fracàs del sistema d'estimació, no ve donat tan sols per la bona implementació dels mètodes de detecció i seguiment propis de visió per computador, sinó un bon desenvolupament en conjunt tant de les tècniques de detecció com de les de classificació, i un estudi exhaustiu de l'abast i context de treball de l'aplicació, contribuiran als bons resultats del sistema de forma global.

Prototipus I

Observant els resultats en quant a detecció de característiques facials amb Viola-Jones, es reflexa que hi ha característiques que es detecten de forma més robusta que d'altres. La cara té un percentatge d'encert entorn al 85% , ulls d'un 65% i nas d'un 60% . Es considera que la eficàcia de detecció del rostre es suficient. Però observem amb claredat que la eficàcia de detecció d'ulls i nas es totalment insuficient. Es per això que es va utilitzar el mètode híbrid de Viola-Jones i Optical Flow per a la detecció ulls, que puja la eficàcia de detecció entorn el 80%, quedant pal·liat el problema de detecció d'ulls. En quant a la detecció del nas, ara mateix es un problema intractable amb Viola-Jones, ja que el seu percentatge d'encert és molt baix. Tampoc es pot utilitzar la solució hibrida emprada als ulls, ja que el conjunt de característiques que composen el nas no es gaire discriminant, i per tant, el nas és perd ràpidament. Finalment, descartem la detecció del nas degut a la ineficàcia dels mètodes utilitzats.

En quant als sistemes de classificació, observem que existeix dispersió dels clústers, produint formacions no solides d'agrupacions i sense una diferencia prou clara. Aquesta dispersió afecta directament al classificador Support Vector Machine, el qual no superen en cap situació un percentatge de classificació del 75%.

Pel que fa al temps d'execució, pot realitza tots els càlculs en qualsevol de forma ràpida, proporcionant una velocitat de processament entre 2 i 3 imatges per segon. Podem concloure que el prototipus I compleix les condicions per a ser utilitzat en temps real.

Els resultats obtinguts del sistema final son molt variants depenent del subjecte que s'estigui tractant. Sobre la primera seqüència s'obté un percentatge d'encert del 68%. Es un percentatge força baix. Mentre que a la segona seqüència el percentatge d'encert és del 86%. Aquesta variabilitat es degut a que la cara del segona seqüència ofereix un bon matching amb les del conjunt d'aprenentatge. Mentre que les cares extretes de la primera seqüència es troben molt lluny dels centres de masses dels clústers.

Els fets comentats ens permeten dir que el sistema detector de rostre i ulls es considera vàlid. Però el procés de classificació manca de fiabilitat, ja que el conjunt d'aprenentatge no es prou variable per a considerar com plausibles models que continguin certa variabilitat respecte el conjunt. Una de les possibles causes d'aquesta ineficàcia és degut a un nefast tractament del

procés de normalització, ja que es descarta l'ampliació del conjunt d'aprenentatge, tenint molta variabilitat de subjectes per posició del cap (1200 imatges per estat).

Com a observació final, diem que el prototipus I només funcionarà eficientment per a rostres que siguin semblants al conjunt d'aprenentatge, donant per fracassat el prototipus ja que una de les premisses bàsiques de l'aplicació es la variabilitat de subjectes.

Prototipus II

Tal i com s'han observat els percentatges d'eficàcia per a conjunts d'imatges, la fiabilitat es prou bona. El problema que s'observa en algunes imatges, es que degut a la variabilitat de malla frontal, en algunes ocasions aquest es capaç d'adaptar-se en situacions amb el subjecte de perfil. En aquets casos el sistema ens proporciona la sortida frontal, encara que això no és del tot cert. Altrament, la variabilitat de malla de perfil també es pot adaptar amb facilitat a cares frontals, produint-nos el mateix error. Una possible solució al problema consistiria en etiquetar cada una de les imatges del conjunt d'entrenament en els següents estats:

- Frontal
    - Frontal
    - Mig-dreta
    - Mig-esquerra
- Dreta
    - Dreta total
    - Mig dreta
- Esquerra
    - Esquerra total
    - Mig Esquerra

Gracies a la diferenciable distribució dels diferents clústers corresponents a la forma, podria ser possible aquesta subdivisió de l'espai de cada model. En quant a l'elecció del submodel final podria venir donat per un classificador *k-nearest-neighbor*.

El temps d'execució del programa es variant en funció de l'estat on es trobi. Si es troba en una fase de re alineament respecte una forma anterior, el temps d'execució serà de 332 ms, donant lloc a un processament de 3 imatges per segon aproximadament. Mentre que en situacions de cerca de la correspondència sobre els tres models, el cost és de 759 ms. Aquest últim temps pot comportar la sensació de salts si el subjecte exerceix molts moviments ràpids. Es considera que el temps d'execució no es pot rebaixar a costa de reduir el nombre d'iteracions de l'algorisme Active Appereance Model, ja que sinó es produiria ineficiència en quant al procés de detecció.

En molts casos la detecció de punts interiors del rostre no troba una correspondència prou eficient. Aquest fet podria ser un problema si desitgéssim analitzar les característiques facials del subjecte. En la nostra aplicació no comporta cap problema, ja que l'objectiu es trobar les característiques de forma global, només per associar al tipus frontal, dreta o esquerra. Una possible solució per a donar robustesa a aquest procés podria consistir en augmentar el nombre d'iteracions del procés d'Active Apperence Model, o bé tractar cada característica fàcil de forma independent al conjunt global.

Concloem que el mètode utilitzat compleix les premisses de variabilitat del subjecte i la possibilitat de funcionar en temps real de forma suficient. Encara que aquest sistema queda una mica curt en quant a abast, ja que només classifica tres estats possibles, per tant, s'hauria d'estudiar la possibilitat d'ampliar el nombre d'estats possibles, per augmentar l'aplicabilitat del mètode.

9.1 ANNEX I: co-autor amb Antonio Hernàndez, Sergio Escalera i Petia Radeva a la publicació CVPR 2010 San Francisco. Workshop Analysis and Modeling Faces and Gestures. Seleccionat com a presentació oral. L'aportació fa referència a la part de reconeixement de la pose a través d'Active Appereance Models

Spatio-Temporal GrabCut Human Segmentation for Face and Pose Recovery

Antonio Hernández[1] ahernandez@cvc.uab.es

Miguel Reyes[1] mreyese@gmail.com

Sergio Escalera[2] sergio@maia.ub.es

Petia Radeva[1] petia@cvc.uab.es

[1] Computer Vision Center, Campus UAB, 08193 Bellaterra, Barcelona, Spain.

[2] Dept. Matem` atica Aplicada i An` alisi, University of Barcelona, Gran Via de les Corts Catalanes 585, 08007 Barcelona, Spain.

Abstract

In this paper, we present a full-automatic Spatio-Temporal GrabCut human segmentation methodology. GrabCut initialization is performed by a HOG-based subject detection, face detection, and skin color model for seed initialization. Spatial information is included by means Shift clustering whereas temporal coherence is considered by the historical of Gaussian Mixture Models. Moreover, human segmentation is combined with Shape and Appearance Models to perform full face and pose re- Results over public data sets as well as proper huaction base show a robust segmentation and recovery face and pose using the presented methodology.

1. Introduction

Human segmentation in uncontrolled environments is a hard task because of the constant changes produced in natscenes: illumination changes, moving objects, changes point of view, or occlusions, just to mention a few. of the nature of the problem, a common way to to discard most part of the image so that the analcan perform over a reduced set of small candidate re- In [5], the authors propose a full-body detector based cascade of classifiers [13] using HOG features. This methodology is currently being used in several works rethe pedestrian detection problem [8]. GrabCut [11] shown high robustness in Computer Vision segmentation problems, defining the pixels of the image as of a graph and extracting foreground pixels via itGraph Cut optimization. This methodology has been to the problem of human body segmentation with success [7]. In the case of working with sequences images, this optimization problem can also be considered to have temporal coherence. In the work of [4], the authors extended the Gaussian Mixture Model (GMM) of algorithm so that the color space is complemented the derivative in time of pixel intensities in order to include temporal information in the segmentation optimization process. However, the main problem of that method is that moving pixels corresponds to the boundaries between foreground and background regions, and thus, there is no clear discrimination.

Once determined a region of interest, pose is often recovered by the determination of the body limbs together *of Mean* with their spatial coherence (also with temporal coherence in case of image sequences). Most of these approaches are probabilistic, and features are usually based on edges or 'ap- *Active* pearance'. In [10], the author propose a probabilistic ap- *covery.* proach for limb detection based on edge learning comple- *man* mented with color information. The image of probabilities *of both* is then formulated in a Conditional Random Field scheme and optimized using believe propagation. This work has obtained robust results and has been extended by other authors including local GrabCut segmentation and temporal refinement of the CRF model [7].

In this paper, we propose a full-automatic Spatio-Temporal GrabCut human segmentation methodology. ural First, subjects are detected by means of a HOG-based cas- in the cade of classifiers. Face detection and skin color model Because are used to define a set of seeds used to initialize Grab- proceed is Cut algorithm. Spatial information is taken into account by ysis means of Mean Shift clustering, whereas temporal informa- gions. tion is considered taking into account the pixel probability on a membership to an historical of Gaussian Mixture Models. Moreover, the methodology is combined with Shape and lated to Active Appearance Models to define three different meshes has also of the face, one near frontal view, and the other ones near lateral views. Temporal coherence and fitting cost are con- nodes sidered in conjunction with GrabCut segmentation to allow erated a smooth and robust face fitting in video sequences. Fi- applied nally, The limb detection and CRF model of [10] is applied high over the obtained segmentation, showing high robustness of capturing body limbs because of the accurate human segmentation.

The rest of the paper is organized as follows: Section 2 GrabCut describes the proposed methodology, presenting the spatio- with temporal GrabCut segmentation, the Active Appearance

models for face fitting, and the pose recovery methodology. Experimental results on two different data sets are performed in Section 3. Finally, Section 4 concludes the paper.

2. Full-body pose recovery

In this section, we present the Spatio-Temporal GrabCut methodology to deal with the problem of automatic human segmentation in video sequences. Then, we describe the Shape and Active Appearance Models used to recover the face, and the body pose recovery methodology based on the approach of [10]. All methods presented in this section are combined to improve final segmentation and pose recovery. Figure 1 illustrates the different modules of the project.

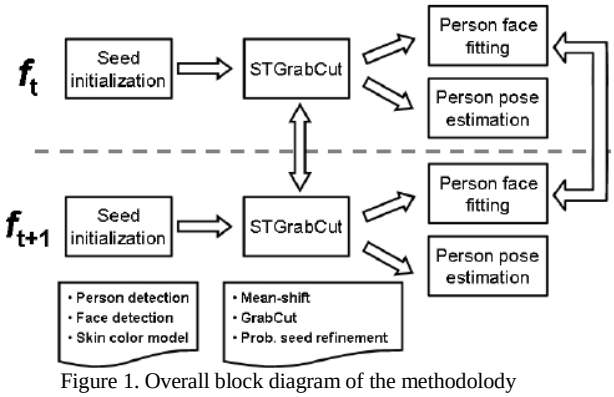


Figure 1. Overall block diagram of the methodolody

2.1. Spatio-Temporal GrabCut segmentation

In [11], the authors proposed an approach to find a binary segmentation -Background, Foreground- of an image by formulating an energy minimization scheme as the one presented in [1], extended using color instead of just gray-scale information. Given a color image, let us consider the array $\mathbf{z} = (z_1, \ldots, z_n, \ldots, z_N)$ of N pixels where $z_i = (R_i, G_i, B_i)$, $i \in [1, \ldots, N]$ in RGB space. The segmentation is defined as array $\boldsymbol{\alpha} = (\alpha_1, \ldots \alpha_N)$, $\alpha_i \in \{0, 1\}$, assigning a label to each pixel of the image indicating if it belongs to background or foreground. A trimap T is de- fined by the user -in a semi-automatic way-, consisting on three regions: T_B, T_F and T_U, each one containing initial background, foreground, and uncertain pixels, respectively.

Pixels belonging to T_B and T_F are clamped as background and foreground respectively -that means GrabCut will not be able to modify these labels-, whereas those belonging to T_U are actually the ones the algorithm will be able to label. Color information is introduced by GMMs. A full covariance GMM of K components is defined for background pixels ($\alpha_i = 0$), and another one for foreground pixels ($\alpha_i = 1$), parametrized as follows

$$\boldsymbol{\theta} = \{\pi(\alpha, k), \mu(\alpha, k), \Sigma(\alpha, k), \alpha \in \{0, 1\}, k = 1..K\}, \tag{1}$$

being π the weights, μ the means and Σ the covariance matrices of the model. We also consider the array $\mathbf{k} = \{k_1, \ldots, k_n, \ldots k_N\}$, $k_i \in \{1, \ldots K\}, i \in [1, \ldots, N]$ indicating the component of the background or foreground GMM (according to α_i) the pixel z_i belongs to. The energy function for segmentation is then

$$\mathbf{E}(\boldsymbol{\alpha}, \mathbf{k}, \boldsymbol{\theta}, \mathbf{z}) = \mathbf{U}(\boldsymbol{\alpha}, \mathbf{k}, \boldsymbol{\theta}, \mathbf{z}) + \mathbf{V}(\boldsymbol{\alpha}, \mathbf{z}), \tag{2}$$

where $\mathbf{U}$ is the likelihood potential, based on the probability distributions $p(\bullet)$ of the GMM:

$$\mathbf{U}(\boldsymbol{\alpha}, \mathbf{k}, \boldsymbol{\theta}, \mathbf{z}) = \sum_i -\log p(z_i | \alpha_i, k_i, \boldsymbol{\theta}) - \log \pi(\alpha_i, k_i) \tag{3}$$

and V is a regularizing prior assuming that segmented regions should be coherent in terms of color, taking into account a neighborhood C around each pixel

$$\mathbf{V}(\boldsymbol{\alpha}, \mathbf{z}) = \gamma \sum_{\{m,n\} \in C} [\alpha_n = \alpha_m] \exp(-\beta z_m - z_n^2) \tag{4}$$

With this energy minimization scheme and given the initial trimap T, the final segmentation is performed using a minimum cut algorithm [1]. The classical semi-automatic GrabCut algorithm is summarized in Algorithm 2.1.

Algorithm 2.1: Original GrabCut algorithm.

1: Trimap T initialization with manual annotation.
2: Initialize $a_i = 0$ for $n \in T_B$ and $a_i = 1$ for $n \in T_U \cup T_F$.
3: Initialize Background and Foreground GMMs from sets $a_n = 0$ and $a_n = 1$ respectively, with k-means.
4: Assign GMM components to pixels.
5: Learn GMM parameters from data z.
6: Estimate segmentation: Graph-cuts.
7: Repeat from step 4, until convergence

Our proposal bases on the previous GrabCut framework, focusing on human body segmentation and extending it by taking into account temporal coherence. We refer to each frame of the video as f_t, $t \in \{1, \ldots, M\}$ being M the length of the sequence. Given a frame f_t, we first apply a person detector based on a cascade of classifiers using HOG features [5]. Then, we initialize the trimap T from the bounding box B retuned by the detector: $T_U = \{z_i \in B\}$, $T_B = \{z_i \in B\}$. Furthermore, in order to increase the /

(a) (b) (c) (d) (e) (f)

Figure 2. STGrabcut pipeline example: (a) Original frame, (b) Seed initialization, (c) GrabCut, (d) Probabilistic re-assignment, (e) Refinement and (f) Initialization mask for f_{t+1}

accuracy of the segmentation algorithm, we include Foreground seeds exploiting spatial and appearance prior information. On one hand, we define a small central region R inside B and set these pixels as Foreground. On the other, we apply a face detector based on a cascade of classifiers using Haar-like features [13] over B, and learn a skin color model h_{skin}. All pixels inside B fitting in h_{skin} are also set to foreground. Therefore, we initialize $T_F = \{z_i \in R\} \cup \{z_i \in \delta(z_i, h_{skin})\}$, where δ returns the set of pixels belonging to the color model defined by h_{skin}. An example of seed initialization is shown in Figure 2(b).

Once we have initialized the trimap, we apply the iterative minimization algorithm shown in steps 4 to 7 of origi- nal GrabCut (algorithm 2.1). However, instead of applying k-means for the initialization of the GMMs we use Mean- Shift clustering, which also takes into account spatial coherence. After convergence, we obtain a segmentation $\boldsymbol{\alpha}$ and the updated foreground and background GMMs $\boldsymbol{\theta}$ at frame f_t, which are used for further initialization at frame f_{t+1}. The result of this step is shown in Figure 2(c). Finally, we refine the segmentation of frame f_t eliminating false positive foreground pixels. By definition of the energy minimization scheme, GrabCut tends to find convex segmentation masks having a lower perimeter, given that each pixel on the boundary of the segmentation mask contributes on the global cost. Therefore, in order to eliminate these background pixels (commonly in concave regions) from the foreground segmentation, we re-initialize the trimap T as follows

$$\begin{aligned} T_B &= \{z_j / \alpha_j^t = 0\} \cup \left\{ z_j / k = t-j \;\middle|\; \frac{p(z_j/\alpha_j^t=0, k_j, \boldsymbol{\theta})}{p(z_j/\alpha_j^t=1, k_j, \boldsymbol{\theta})} > \right\} \\ T_U &= \{z_j / \alpha_j^t = 1\} \therefore T_B \\ T_F &= \{z_i \in \delta(z_i, h_{skin})\} \end{aligned} \quad (5)$$

where the pixel background probability membership is computed using the GMM models of previous segmentations. The result of this step is shown in Figure 2(d). Once the trimap has been redefined, false positive foreground pix- els still remain, so the new set of seeds is used to iterate again GrabCut algorithm, resulting in a more accurate segmentation, as we can see in Fig. 2(e). Finally, considering A as the binary image representing $\boldsymbol{\alpha}$ at f_t -the one obtained before the refinement-, we initialize the trimap for f_{t+1} as follows

$$\begin{aligned} T_F &= \{z_i \in A \quad ST_d\} \\ T_U &= \{z_i \in A \oplus ST_e\} \therefore T_F \\ T_B &= \{z_i, i = 1..N\} \therefore (T_F \cup T_U) \end{aligned} \quad (6)$$

where and $\oplus$ are erosion and dilation operations with their respective structuring elements ST_d and ST_e. The structuring elements are simple squares of a given size depending on the size of the person and the degree of movement we allow from f_t to f_{t+1}, assuming smoothness in the movement of the person. An example of a morphological mask is shown in Figure 2(f). The whole segmentation methodology is detailed in the ST-GrabCut algorithm 2.2.

Algorithm 2.2: Proposed ST-GrabCut algorithm.

1: Person detection on f_1.
2: Face detection and skin color model learning.
3: Trimap T initialization with detected bounding box and learnt skin color model.
4: Initialize $a_i = 0$ for $n \in T_B$ and $a_i = 1$ for $n \in T_U \cup T_F$.
5: Initialize Background and Foreground GMMs from sets $a_n = 0$ and $a_n = 1$ respectively, with Mean-shift.
6: **for** $t = 1 \ldots M$
7: Person detection on f_t.
8: Assign GMM components to pixels of f_t.
9: Learn GMM parameters from data z.
10: Estimate segmentation: Graph-cuts.
11: Repeat from step 8, until convergence Re-
12: initialize trimap T (equation 5). Assign
13: GMM components to pixels.
14: Learn GMM parameters from data z.
15: Estimate segmentation: Graph-cuts.
16: Repeat from step 12, until convergence
17: Initialize trimap T using segmentation obtained in step 11 after convergence (equation 6) for f_{t+1}.

2.2. Face fitting

Once we have properly segmented the body region, next step consists of fitting the face and the body limbs. For the case of face recovery, we base our procedure on mesh fitting using Active Appearance Models (AAM), that benefits from Active Shape Models and color and texture information [2].

Active Appearance Model is generated by combining a model of shape and texture variation. First, a set of points are marked on the face of the training images that are aligned, and a statistical shape model is build [3]. Each training image is warped so the points match those of the mean shape. This is raster scanned into a texture vector, $\mathbf{g}$, which is normalized by applying a linear transformation, $\mathbf{g} \rightarrow (\mathbf{g} - \mu_g \mathbf{1})/\sigma_g$, where $\mathbf{1}$ is a vector of ones, and μ_g and σ_g^2 are the mean and variance of elements of $\mathbf{g}$. After normalization, $\mathbf{g}^T \mathbf{1} = 0$ and $|\mathbf{g}| = 1$. Then, eigenanalysis is applied to build a texture model. Finally, the correlations between shape and texture are learnt to generate a combined appearance model. The appearance model has parameter $\mathbf{c}$ controlling the shape and texture according to

$$x = \bar{x} + \mathbf{Q}_s \mathbf{c} \quad (7)$$

$$g = \bar{g} + \mathbf{Q}_g \mathbf{c} \quad (8)$$

where $\bar{x}$ is the mean shape, $\bar{g}$ the mean texture in a mean shaped patch, and $\mathbf{Q}_s$, $\mathbf{Q}_g$ are matrices designing the modes of variation derived from the training set. A shape $\mathbf{X}$ in the image frame can be generated by applying a suitable transformation to the points, $\mathbf{x}$: $\mathbf{X} = S_t(\mathbf{x})$. Typically, S_t will be a similarity transformation described by a scaling s, an in-plane rotation, θ, and a translation (t_x, t_y).

Once constructed the AAM, it is deformed on the image to detect and segment the face appearance as follows. During matching, we sample the pixels in the region of interest $\mathbf{g}_{im} = T_u(\mathbf{g}) = (u_1 + 1)\mathbf{g}_{im} + u_2\mathbf{1}$, where $\mathbf{u}$ is the vector of transformation parameters, and project into the texture model frame, $\mathbf{g}_s = T_u^{-1}(\mathbf{g}_{im})$. The current model texture is given by $\mathbf{g}_m = \bar{g} + \mathbf{Q}_g \mathbf{c}$. The current difference between model and image (measured in the normalized texture frame) is as follows

$$\mathbf{r}(\mathbf{p}) = \mathbf{g}_s - \mathbf{g}_m \quad (9)$$

Given the error $E = |\mathbf{r}|^2$, we compute the predicted displacements $\delta\mathbf{p} = -\mathbf{R}\mathbf{r}(\mathbf{p})$, where $\mathbf{R} = \left(\frac{\partial \mathbf{r}^T}{\partial \mathbf{p}} \cdot \frac{\partial \mathbf{r}}{\partial \mathbf{p}}\right)^{-1} \frac{\partial \mathbf{r}^T}{\partial \mathbf{p}}$.

The model parameters are updated $\mathbf{p} \rightarrow \mathbf{p} + k\delta\mathbf{p}$, where initially $k = 1$. The new points $\mathbf{X}$ and model frame texture $\mathbf{g}_m$ are estimated, and the image is sampled at the new points to obtain $\mathbf{g}_{mi}$, obtaining the new error vectoras $\mathbf{r} = T_u^{-1}(g_{im}) - g_m$. A final condition guides the end of each iteration: if $|\mathbf{r}|^2 < E$, then we accept the new estimate, otherwise, we set to $k = 0.5$, $k = 0.25$, and so on.

Taking into account the discontinuity that appears when a face moves from frontal to profile view we use three different AAM corresponding to three meshes of 21 points: frontal view $_F$, right lateral view $_R$, and left lateral view $_L$. In order to include temporal and spatial coherence, meshes at frame f_{t+1} are initialized by the fitted mesh points at frame f_t. Additionally. we include a temporal change-mesh control procedure, as follows

$$^{t+1} = \min_{^{t+1}} \{E_F, E_R, E_L\}, \quad ^{t+1} \in \nu( ) \quad (10)$$

where $\nu(\cdot)$ corresponds to the meshes contiguous to the mesh ^{t} fitted at time t (including the same mesh). This constraint avoids false jumps and imposes smoothness in the temporal face behavior (e.g. a jump from right to left profile view is not allowed).

In order to obtain a more accurate pose estimation, after fitting the mesh, we take advantage of its variability to differentiate among a set of head poses. We have defined five different head poses: right, middle-right, frontal, middle- left, and left. In order to define this set, the fitted frontal meshes in the training set are classified in three different poses: middle-right, frontal, and middle left, whereas the training samples of the left and right meshes are directly classified in full-left and full-right poses, respectively. In order to learn the five different head poses, training images are aligned, and PCA is applied to save the 20 most representative eigenvectors. Then, a new mesh is projected to that new space and classified to one of the five different head poses according to a 3-Nearest Neighbor rule.

Figure 3 shows examples of the AAM model fitting in natural images (obtained from [9]) for the three different meshes.

Figure 3. From left to right: left, frontal, and right mesh fitting using AAM.

2.3. Pose recovery

Considering the refined segmented body region, we construct a pictorial structure model [6]. We use the method of The procedure is repeated until no improvement is made to the error.

The posterior of a configuration of parts $L = l_i$ given a frame f_t is

$$P(L|f_t) \propto \exp\left(\sum_{(i,j)\in E} \Psi(l_i, l_j) + \sum_i \Phi(l_i|f_t) \right) \quad (11)$$

The pairwise potential $\Psi(l_i, l_j)$ corresponds to a spatial prior on the relative position of parts and embeds the kinematic constraints. The unary potential $\Phi(l_i|f_t)$ corresponds to the local image evidence for a part in a particular position. Inference is performed over tree-structured conditional random field by sum-product Belief Propagation.

Since the appearance of the parts is initially unknown, a first inference uses only edge features in Φ. This delivers soft estimates of body part positions, which are used to build appearance models of the parts and background (color histograms). Inference in then repeated with Φ using both edges and appearance. This parsing technique simultaneously estimates pose and appearance of parts. For each body part, parsing delivers a posterior marginal distribution over location and orientation (x, y, ϕ) [10, 7].

3. Results

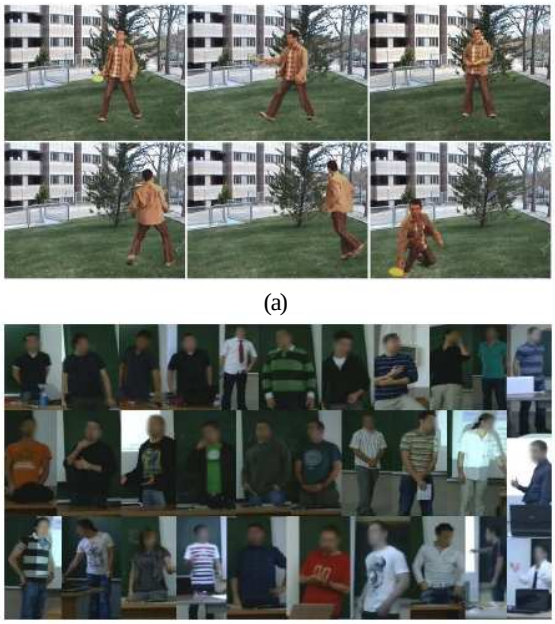
Before the presentation of the results, we discuss the data, methods and parameters of the comparative, and validation measurements.

◦ *Data*: We use the public image sequences of the Chroma Video Segmentation Ground Truth (cVSG) [12], a corpus of video sequences and segmentation masks of people. Chroma based techniques have been used to record Foregrounds and Backgrounds separately, being later combined to achieve final video sequences and accurate segmentation masks almost automatically. Some samples of the sequence we have used for testing are shown in Fig- ure 4(a). The sequence has a total of 307 frames. This image sequence includes several critical factors that make segmentation difficult: object textural complexity, object struc- ture, uncovered extent, object size, Foreground and Back- ground velocity, shadows, background textural complexity, Background multimodality, and small camera motion. Al- ternatively as a second database we have also used a set of 30 videos corresponding to the defense of undergraduate thesis at the University of Barcelona to test the methodol- ogy in a different environment (UBDataset). Some samples of this data set are shown in Figure 4(b).

◦ *Methods*:

We test the classical semi-automatic GrabCut algorithm for human segmentation comparing with the proposed ST-GrabCut algorithm. We also test the mesh fitting and body pose recovery methodologies over the obtained segmentation.

◦ *Validation measurements*: In order to evaluate the robustness of the methodology for human body segmentation,



(a)

(b)

Figure 4. (a) Samples of the cVSG corpus and (b) UBDataset image sequences.

face and pose fitting, we use the ground truth masks of the video sequences to compute the overlapping factor O as follows

$$O = \frac{M_{GC} \cap M_{GT}}{M_{GC} \cup M_{GT}} \quad (12)$$

where M_{GC} and M_{GC} are the binary masks obtained for spatio-temporal GrabCut segmentation and the ground truth mask, respectively.

3.1. Spatio-tempral GrabCut Segmentation

First, we test the proposed ST-GrabCut segmentation on the sequence from the public cVSG corpus. The results for the different experiments are shown in Table 1. In order to avoid the manual initialization of classical GrabCut algorithm, for all the experiments, seed initialization is per- formed applying the commented person HOG detection, face detection, and skin color model. First row of Table 1 shows the overlapping performance of 12 applying GrabCut segmentation with k-means clustering to design the GMM models. Second row shows the overlapping performance considering Mean Shift clustering to design the GMM models. One can see a slight improvement when using the second strategy. This is mainly due to the fact that Mean Shift clustering takes into account spatial information of pixels in clustering time, which better defines contiguous pixels of image to belong to GMM models of foreground and back- ground. Third performance in Table 1 shows the overlap- ping results considering the morphology refinement based

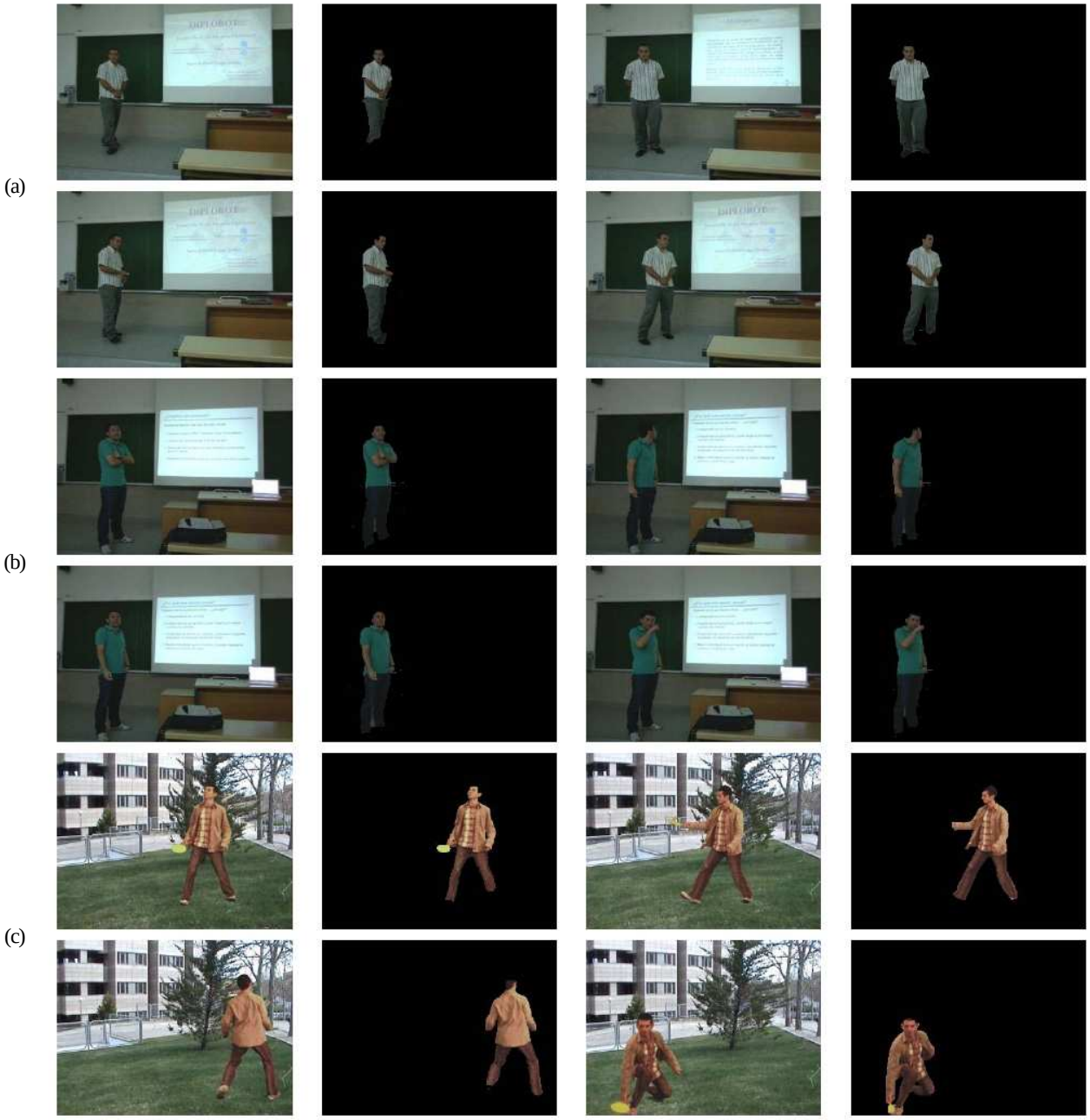


Figure 5. Segmentation examples of (a) UBDataset sequence 1, (b) UBDataset sequence 2 and (c) cVSG sequence.

on previous segmentation. In this case, we obtain near 10% of performance improvement respect the previous result. Finally, last result of Table 1 shows the full-automatic ST-GrabCut segmentation overlapping performance. One can see that it achieves about 25% of performance improvement in relation with the previous best performance. Some segmentation results obtained by the GrabCut algorithm for the cVSG corpus are shown in Figure 5. Note that the ST-GrabCut segmentation is able to robustly segment convex regions. We have also applied the ST-GrabCut segmentation methodology on the image sequences of UBDataset. Some segmentations are shown in Figure 5.

3.2. Face fitting

In order to measure the robustness of the spatio-temporal AAM mesh fitting methodology, we performed the overlapping analysis of meshes in both un-segmented and segmented image sequence of the public cVSG corpus. Over-

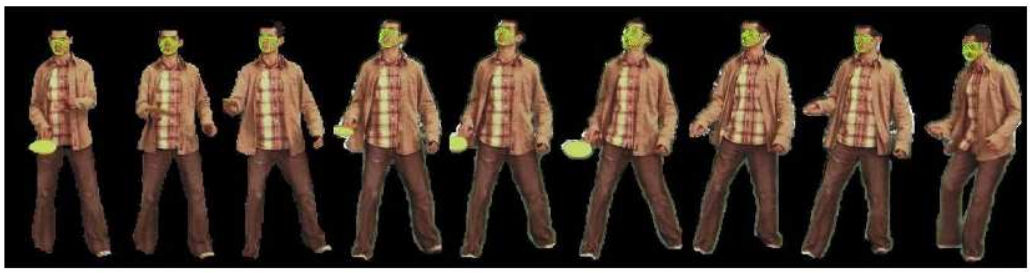


Figure 6. Samples of the segmented cVSG corpus image sequences fitting the different AAM meshes.

Approach	Mean overlapping
K-means	0.5356
Mean-shift	0.5424
Morphology	0.6229
ST-GrabCut	0.8747

Table 1. GrabCut and ST-GrabCut Segmentation results on cVSG corpus.

Approach	Mean overlapping
Mesh fitting without segmentation	0.8960
ST-Grabcut & Temporal mesh fitting	0.9636

Table 2. AAM mesh fitting on original images and segmented images of the cVSG corpus.

Face view	Percentage of frames
Left view	0.1300
Near Left view	0.1470
Frontal view	0.2940
Near Right view	0.1650
Right view	0.2340

Table 3. Face pose percentages on the cVSG corpus.

lapping results are shown in Table 3. One can see that the mesh fitting works fine in unsegmented images, obtaining a final mean overlapping of 89.60%. However, note that combining the temporal information of previous fitting and the ST-GrabCut segmentation, the face mesh fitting considerably improves, obtaining a final of 96.36% of overlap- ping performance. Some example of face fitting using the AAM meshes for different face poses of the cVSG corpus are shown in Figure 6.

Finally, we have tested the classification of the five face poses on the cVSG corpus, obtaining the percentage of frames of the subject at each pose. The obtained percent- ages are shown in Table 3.

3.3. Body limbs recovery

Finally, we combine the previous segmentation and face fitting with a full body pose recovery [10]. In order to show the benefit of applying previous ST-GrabCut segmentation, we perform the overlapping performance of full pose recovery with and without human segmentation, always within the bounding box obtained from HOG person detection. Results are shown in Table 4. One can see that pose recovery considerably increases its performance when reducing the region of search based on ST-GrabCut segmentation. Some examples of pose recovery within the human segmentation regions for cVSG corpus and UBdataset are shown in Figure 7. One can see that in most of the cases body limbs are correctly detected. Only in some situations, occlusions or changes in body appearance can produce a wrong limb fitting.

Approach	Mean overlapping
Limb recovery without segmentation	0.7919
ST-Grabcut & Limb recovery	0.8760

Table 4. Overlapping of body limbs based on ground truth masks.

Finally, in Figure 8 we show the application of the whole framework to perform temporal tracking, segmentation and full face and pose recovery. The colors correspond to the body limbs. The colors increase in intensity based on the instant of time of its detection. One can see the robust detection and temporal coherence base on the smooth displacement of face and limb detections.

4. Conclusion

In this paper, we presented an evolution of the semi-automatic GrabCut algorithm for dealing with the problem of human segmentation in image sequences. The new full-automatic ST-GrabCut algorithm uses a HOG-based per- son detector, face detection, and skin color model to initialize GrabCut seeds. Spatial coherence is introduced via Mean Shift clustering, and temporal coherence is considered based on the historical of Gaussian Mixture Models. The segmentation procedure is combined with Shape and Active Appearance models to perform full face and pose recovery.

This general and full-automatic human segmentation, pose recovery, and tracking methodology showed higher performance than classical approaches in public image sequences from uncontrolled environments, which makes it useful for general human face and gesture analysis applications

9.2 ANNEX II: ANGLES D' EULER

Tractant el cap com un projecte en un espai tridimensional, distingim tres angles de rotació sobre tres eixos perpendicular. Aquests angles els anomenem angles d'Euler[16] :

- Yaw, angle sobre de rotació l'eix x
- Pitch , angle de rotació sobre l'eix y
- Roll, angle de rotació sobre l'eix z

Les matrius de transformació corresponent a cada angle són:

$$R_{z,\varphi} = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix};$$

$$R_{y,\Theta} = \begin{bmatrix} \cos\Theta & 0 & \sin\Theta \\ 0 & 1 & 0 \\ -\sin\Theta & 0 & \cos\Theta \end{bmatrix};$$

$$R_{x,\psi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix}.$$

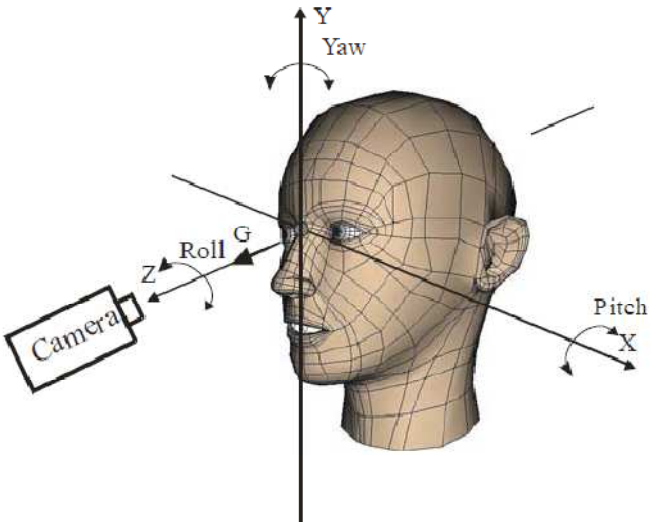
La rotació completa en 3D ve donada pel producte de les matrius en ordre *Rz,Ry i Rx*, donant el següent resultat:

$$\mathbf{R} = \begin{bmatrix} c\varphi \cdot c\Theta & c\varphi \cdot s\Theta \cdot s\psi - s\varphi \cdot c\psi & c\varphi \cdot s\Theta \cdot c\psi + s\varphi \cdot s\psi \\ s\varphi \cdot c\Theta & s\varphi \cdot s\Theta \cdot s\psi + c\varphi \cdot c\psi & s\varphi \cdot s\Theta \cdot c\psi - c\varphi \cdot s\psi \\ -s\Theta & c\Theta \cdot s\psi & c\Theta \cdot c\psi \end{bmatrix}$$

On *c* correspon al cosinus i *s* al sinus

Abans d'aplicar les transformacions sobre l'objecte cal establir un origen, en el nostre cas l'origen correspon al punt entre els dos ulls, obviant l'eix Z, ja que segons la nostra aplicació no treballem sobre la dimensió Z.

$$X_0 = X_l + \frac{X_r - X_l}{2};\ Y_0 = Y_l + \frac{Y_r - Y_l}{2}$$



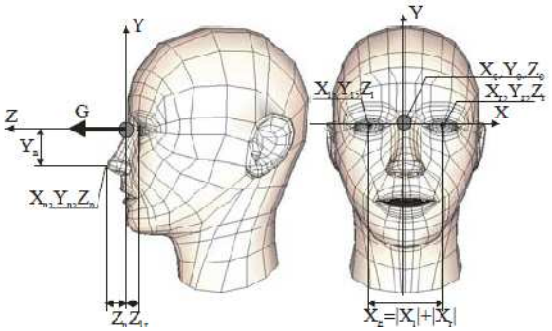
Degut a que tenim tant les coordenades de la posició inicial i les coordenades de la posició final, haurem de resoldre el següent sistema per obtenir els diferents angles de gir:

$$\mathbf{R}\cdot\begin{bmatrix} X_{px}\cdot d \\ Y_{px}\cdot d \\ Z_{px}\cdot d \end{bmatrix} = \begin{bmatrix} X_{fx}\cdot d \\ Y_{fx}\cdot d \\ Z_{fx}\cdot d \end{bmatrix}.$$

On X_{px} , Y_{px} , Z_{px} corresponen als punts inicials, abans de la rotació 3D, de la característica facial x.

On X_{fx} , Y_{fx} , Z_{fx} corresponen als punts finals, després de la rotació 3D, de la característica facial x.

Resultant un sistema de nou equacions. Com que es tracta d'equacions no lineals utilitzarem el mètode d'optimització dels mínims quadrats per a resoldre-ho.



Cal tenir en compte que després d'una rotació les posicions dels paràmetres sofreixen certa variacions degut a la vista en perspectiva. Ho corregim mitjançant:

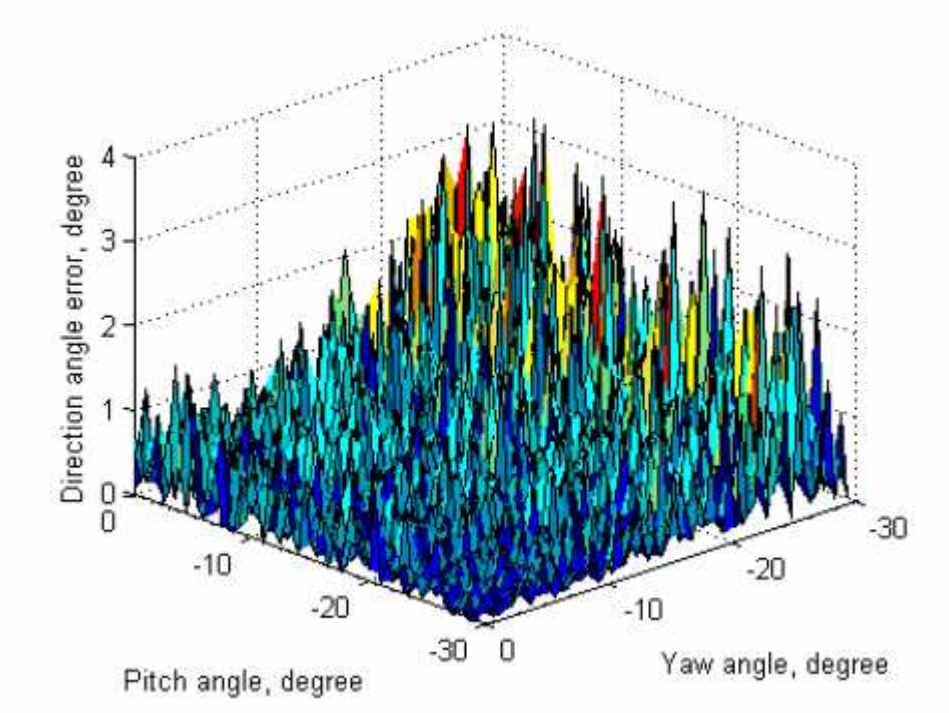
$$d - \frac{\sqrt{(x_{pr} - x_{pl})^2 + (y_{pr} - y_{pl})^2}}{X_E}.$$

Encara ens faltaran certs paràmetres respecte a distancies sobre l'eix z, prendrem alguns valors estàndards i mitjans:

Z_n = 21 mm, Y_n = 41 mm, Z_{lr} = 19.5 mm, X_E = 70 mm
L'eficiència del sistema recau principalment en una bona estimació dels punts sobre el rostre, aquest punts són:

- Ull dret
- Ull esquerre
- Nas

Amb una bona localització dels punts i suposant que l'angle roll es mantingui a zero, per a imatges de mida 320x240 produïm els següents errors:



S'observen que els grans errors es produeixen amb un angle de yaw entre -20 i -30 juntament amb angles pitch entre 0 i -10. Cal remarcar que aquestes proves s'han fet localitzant manualment la posició d'ulls i nas.

9.3 ANNEX III: ENTRENAMENT AUTOMATIC DE MODELS ESTADISTICS DE FORMA

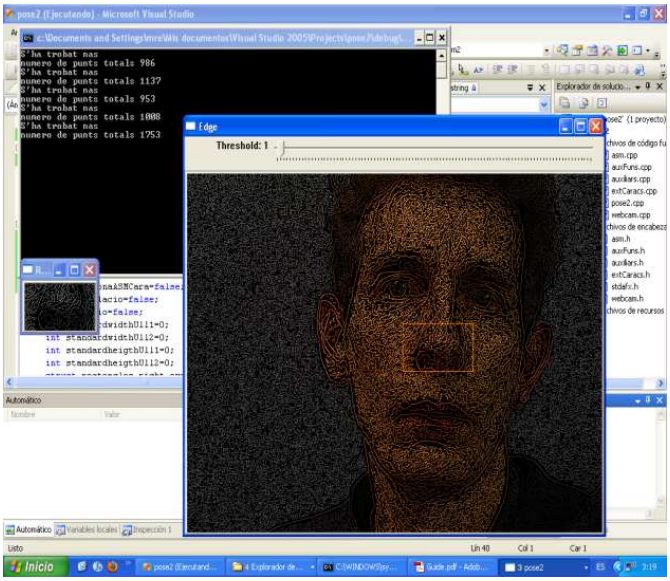
En moltes ocasions, precisem d'una descripció molt exacta de la forma que estem tractant si volem fer un anàlisi comportament molt detallat. Es per això que necessitem que els models a tractar han de ser molt fidels al conjunt d'entrenament. Una de les formes més eficients per a estudiar el comportament d'un exemple de test, es que aquest ja formi part del conjunt d'entrenament. Dit d'altra manera, seria bo de que la pròpia aplicació aprengués models extrets de les imatges de test que tingui una bona correspondència.

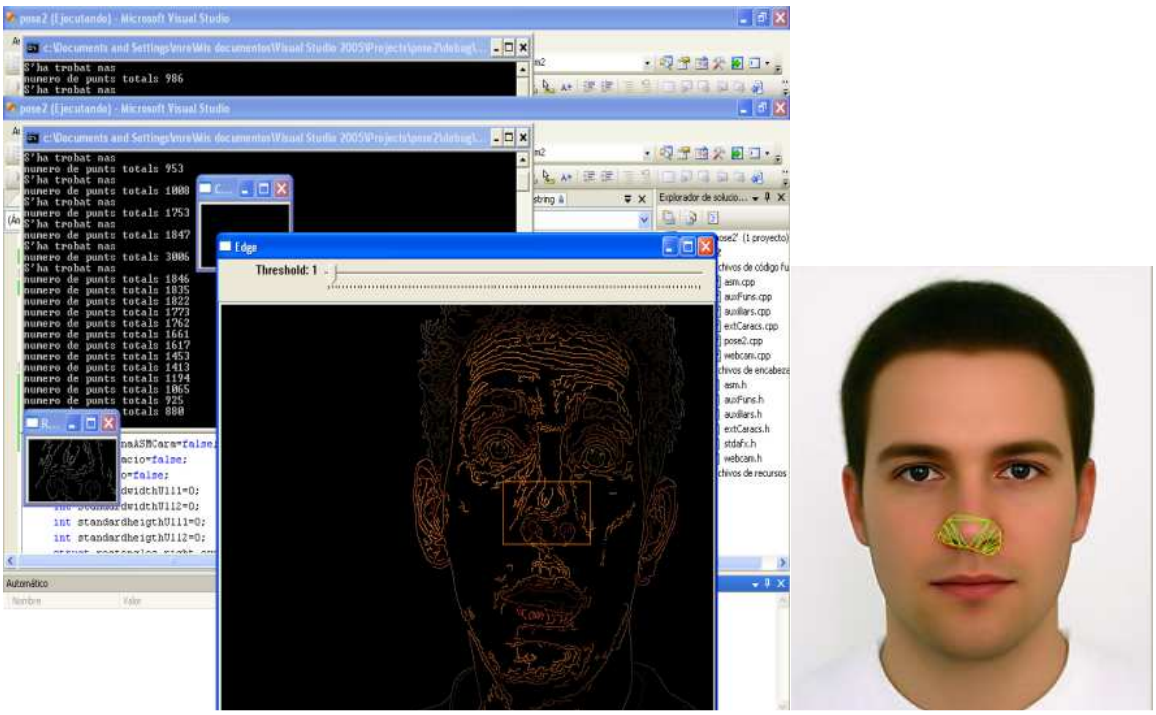
Per aconseguir aquest fi hem de ser capaç de definir una heurística prou solida capaç de considerar que un model suficientment plausible, com per afegir-ho al conjunt d'entrenament. Si aquesta funció no esta ben formulada, o existeix una manca de components discriminants, el nostre sistema pot tendir cap al fracàs estrepitosament, aprenent models que no s'ajusten al conjunt d 'aprenentatge original i considerant-los ,posteriorment, com a vàlids.

Un dels mètodes proposats consisteix en aprendre la forma dels contorns d'una certa regió. Reduint el valor llindar dels contorns que apareixen sobre una imatge, molt semblant al procediment de piràmide de gaussianes, trobarem contorns on es manté la forma de cert objecte o figura que vulguem modelar.
Aquest mètode també pot servir per a modelar objectes que conformen un estructura molt complexa i seria una tasca feixuga amb el mètode de col·locació manual dels landmarks.

L'auto aprenentatge es basaria en examinar aquests contorns, elaborant estructures de contorns, i si aquestes es consideren suficientment similars al conjunt d'aprenentatge, passarien a formar part d'aquest.

A les següents imatges veiem com s'ha auto après la forma d'un nas.




pose2 (Ejecutando) - Microsoft Visual Studio
Edge
Threshold: 1

9.4 **ANNEX IV**: CONTINGUT DEL CD

Els continguts del cd son els següents:

1. Conjunt d'entrenament del prototipus I
2. Conjunt d'entrenament del prototipus II
3. Codi font del prototipus I
4. Aplicació executable del prototipus II per a web-cam
5. Aplicació executable del prototipus II per a entrada de vídeo
6. Aplicació per a l'entrenament de patrons amb Active Appereance model
7. Aplicació Matlab per al càlcul d'angles d'Euler
8. Memòria en format PDF

A cada una de les aplicacions cal llegir l'arxiu readme.txt per a entendre el seu funcionament.

10. BIBLIOGRAFIA

[1] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models–their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995

[2] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001

[3] J. Xiao, T. Moriyama, T. Kanade, and J. Cohn, "Robust full-motion recovery of head by dynamic templates and registration techniques," Int'l. J. Imaging Systems and Technology, vol. 13, no. 1, pp. 85–94, 2003

[4]Tania Mezzadri Centeno, Heitor Silvério Lopes, Marcelo Kleber Felisberto and Lúcia Valéria Ramos de Arruda CPGEI/CEFET-PR, Av. 7 de setembro, 3165, CEP: 80230-000 Curitiba-PR, Brazil

[5] "Object Detection for Computer Vision Using a Robust Genetic Algorithm Jianbo Shi and Jitendra Malik, Member, IEEE Normalized Cuts and Image Segmentation IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 22, NO. 8, AUGUST 2000

[6]Image Processinghttp://www.efg2.com/Lab/Library/ImageProcessing/Algorithms.htm

[7] PAUL VIOLA Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA MICHAEL J. JONES Mitsubishi Electric Research Laboratory, 201 Broadway, Cambridge, MA 02139, USA Robust Real-Time Face Detection Received September 10, 2001; Revised July 10, 2003; Accepted July 11, 2003

[8] H. Rowley, S. Baluja, T. Kanade, K. Sung and T. Poggio: CMU Face Database, upright set (tests A, B and C). The Vision and Autonomous Systems Centre, Carnegie Mellon University (VASC/CMU) - Massachusetts Institute of Technology (MIT), 1997.

[9] Distinctive Image Features from Scale-Invariant Keypoints David G. Lowe January 5, 2004 International Journal of Computer Vision, 2004.

[10] SURF: Speeded Up Robust Features Herbert Bay1, Tinne Tuytelaars2, and Luc Van Gool12

[11]Nonlinear Mean Shift for Robust Pose Estimation Raghav Subbarao Yakup Genc‡ Peter MeerECE Department Real-time Vision and Modeling Department Rutgers University Siemens Corporate Research Piscataway, NJ 08854 Princeton, NJ 08540

[12]ENHANCED SUPPORT REGION FOR SCALE-SPACE BLOB DETECTION Cattleya Duanggate,Bunyarit Uyyanonvara, Stanislav S. Makhanov and Sarah Barman School of Information, Computer and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University 131 Moo 5, Tiwanont Road, Bangkadi, Muang, Pathumthani, 12000, Thailand

[13]] Michael Kearns and Leslie G. Valiant. Learning Boolean formulae or _nite automata is as hard as factoring. Technical Report TR-14-88, Harvard University Aiken Computation Laboratory, August 1988.

[14] Michael Kearns and Leslie G. Valiant. Cryptographic limitations on learning Boolean formulae and _nite automata. Journal of the Association for Computing Machinery, 41(1):67.95, January 1994.

[15] using sequential minimal optimization. Advances in Kernel Methods – Support Vector Learning (pp. 185–208). MIT Press.

[16] IST-2003-511598 (NoE) COGAIN Communication by Gaze Interaction Network of Excellence Information Society Technologies D5.2 Report on New Approaches to Eye TrackingActual submission date: 23.10.2006

[17] Computer Vision a Modern Approach, Forsyth, Ponce. Prentice Hall 2007.

[18] Hartley, R; Ziserman, A: Multiple View Geometry in Computer Vision, Cambridge University Press, 2000.

RESUM

Durant els últims anys s'estan duent a càrrec aplicacions prou sòlides i robustes en quant a sistemes de detecció. Degut a aquest fet, dia rere dia se'ns ofereixen noves solucions i aplicacions que treuen el màxim profit a aquestes sistemes, com és el cas de la identificació biomètrica. Avui en dia existeixen aplicacions totalment funcionals basades en sistemes de detecció i identificació. Aquest fet, ens planteja l'idea de donar pas endavant, obrint una nova línea d'investigació, amb l'objectiu d'arribar a implementar aplicacions que siguin capaç d'analitzar el comportament d'un espai determinat. Sobre aquest teixit es desenvolupa la nostra aplicació, exercint primerament, un anàlisi i estudi exhaustiu sobre les tècniques i metodologies de detecció pròpies de visió per computador. Aquest estudi ens aportarà una capacitat d'anàlisi per afrontar el repte de la classificació de la posició del cap. Al llarg del desenvolupament, també ens endinsarem en elements molt comuns, i necessaris, per a la consecució d'aquests tipus de sistemes, utilitzant les tècniques de Machine Learning. A més, creiem que ens trobem en un context oportú per a endinsar-nos en aquestes línies d'investigació, ja que moltes de les robustes i sòlides formulacions de l'àrea de la intel·ligència artificial, origen de la visió per computador, van ser creades durant la dècada dels 80 i 90, que aleshores sovint eren consideres computacionalment intractables, en canvi, a l'actualitat les podem dur a terme amb força agilitat gracies als avenços dels sistemes computacionals actuals.

Resumen

Durante los últimos años se están llevando a cargo aplicaciones suficientemente sólidas y robustas en cuanto a sistemas de detección. Debido a este hecho, día tras día se nos ofrecen nuevas soluciones y aplicaciones que sacan el máximo rendimiento a estos sistemas, como es el caso de la identificación biométrica. Hoy en día existen aplicaciones totalmente funcionales basadas en sistemas de detección e identificación. Este hecho, nos plantea la idea de dar paso adelante, abriendo una nueva línea de investigación, con el objetivo de llegar a implementar aplicaciones que sean capaces de analizar el comportamiento de un espacio determinado. Sobre este tejido se desarrolla nuestra aplicación, ejerciendo primero, un análisis y estudio exhaustivo sobre las técnicas y metodologías de detección propias de visión por computador. Este estudio nos aportará una capacidad de análisis para afrontar el reto de la clasificación de la posición de la cabeza. A lo largo de su desarrollo, también nos adentraremos en elementos muy comunes, y necesarios, para la consecución de estos tipos de sistemas, empleando las técnicas de Machine Learning. Además, creemos que nos encontramos en un contexto oportuno para adentrarnos en estas líneas de investigación, ya que muchas de las robustas y sólidas formulaciones del área de la inteligencia artificial, origen de la visión por computador, fueron creadas durante la década de los 80 y 90, cuando entonces, a menudo, eran consideras computacionalmente intratables, en cambio, en la actualidad las podemos llevar a cabo con mucha agilidad gracias a los avances de los sistemas computacionales actuales.

ABSTRACT

In recent years are being implemented sufficiently strong and robust applications in terms of detection systems. Because of this, day after day we offer new solutions and applications that take full advantage of these systems, such as biometric identification. Today there are fully functional applications based on the detection and identification systems. This fact raises the idea of giving step forward, opening a new line of research, aiming to reach deploy applications that are able to analyze the behavior of a given space. About this tissue develops our application, putting first, a thorough analysis and study on the detection techniques and methodologies specific to computer vision. This study will test the capacity to meet the challenges of the classification of the head pose. Throughout its development, also we enter a very common and necessary, to achieve these types of systems, using Machine Learning techniques. Furthermore, we believe we are in a context to get into this line of research, since many of the robust and solid formulations of the area of artificial intelligence, the source of computer vision, were created during the 80s and 90, when then, often, were considered computationally intractable, however, we can now perform with agility thanks to advances in current computer systems