

Seguiment de les extremitats dels ratolins i les vibrisses del musell.

TFM àrea Intel·ligència Artificial

Marc Górriz Marcelino
UOC. TFM. Curs 2013-2014
Consultor: Samir Kanaan Izquierdo

Agraïments

- A en David Masip Rodo per introduir-me en aquest projecte i tot el seu suport al llarg del desenvolupament d'aquest.
- A en Samir Kanaan Izquierdo pels consells i el suport en tots els aspectes del projecte, des de l'elecció d'aquest fins a la finalització.

Índex

Agraïments	2
1. Elecció del tema del TFM	7
2. Objectius del TFM.....	8
3. Introducció	9
3.1 Visió artificial	9
3.2 Experiments optogenètics.....	9
3.2.1 Optogenètica.....	9
3.3 Anatomia ratolins.....	10
3.3.1 Potes.....	10
3.3.2 Vibrisses	10
4. Planificació	11
4.1.Taula de fites	11
4.2.Lliurables	12
4.3. Diagrama de Gantt	13
5. Preparació	14
5.1 Eines	14
5.1.1 Open CV.....	14
5.1.2 Python	15
5.2 Algoritmes	15
5.2.1 Mètodes del valor llindar	15
5.2.2 Mascarees i convolucions.....	17
5.2.2.1 Filtres de HAAR.....	18
5.2.2.2 Gradient.....	19
5.2.2.3 Filtre de mitjana	20
5.2.3 Transformada de Hough.....	21
5.2.4 Scale-invariant feature transform	22
5.2.5 Canny edge detector	23
5.2.5 Optical flow	24
6. Execució.....	25
6.1 Detecció de potes.....	25
6.1.1 Fase 1.....	25
6.1.2 Fase 2.....	26

6.1.3 Fase 3.....	32
6.2 Detecció de vibrisses	33
6.2.1 Fase 1: Tria angle imatges	33
6.2.2 Fase 2: Tractament de la imatge	34
6.2.3 Fase 3.....	39
7. Resultats	41
7.1 Resultats potes	41
7.2 Resultats vibrisses	46
8. Conclusions del Treball.....	52
9. Problemes trobats.....	53
10. Possibles ampliacions.....	54
11.Bibliografia	55
Annex 1: Classes Python.....	56
Annex 2: Eines de Test	63

Índex de figures

Figura 1 Diagrama de Gantt	13
Figura 2. Valor Llindar Global	16
Figura 3. Valor Llindar Adaptatiu	17
Figura 4 Filtres de Haar	18
Figura 5 Gradient.....	19
Figura 6. Filtre de mitjana	20
Figura 7 Transformada de Hough.....	21
Figura 8 SIFT	22
Figura 9 Canny Edges	23
Figura 10 Optical flow	24
Figura 11. Angle imatge potes.....	25
Figura 12 Potes angle superior.....	25
Figura 13. Generador Haar Correctes	26
Figura 14. Imatge negativa Haar	27
Figura 15. Imatges Haar Detecció Correcte	27
Figura 16. Imatges Haar Detecció Incorrecte.....	27
Figura 17. Binarització Global Potes.....	28
Figura 18. Contorns Potes Binarització Global	29
Figura 19 Mostres training pota.....	30
Figura 20 Detecció correcta SIFT.....	31
Figura 21 Detecció errònia SIFT.....	31
Figura 22 Potes alta resolució	31
Figura 23. Optical Flow terra.....	32
Figura 24. Gràfica Optical Flow	32
Figura 25. Angle imatge vibrisses.....	33
Figura 26 Vibrisses angle lateral.....	33
Figura 27 Vibrisses angle superior.....	34
Figura 28. Binarització Adaptativa Vibrisses	34
Figura 29. Contorns Binarització Global.....	35
Figura 30. Contorns Binarització Adaptativa Gauss	35
Figura 31 Vibrisses modificació contrast.....	36
Figura 32 Canny edges	36
Figura 33 Transformada de Hough correcte	36
Figura 34 Trasnformada de Hough incorrecte.....	37
Figura 35 Vibrisses creuades	39
Figura 36. Vibrisses Optical Flow.....	39
Figura 37. Vibrisses Optical Flow.....	40
Figura 38. SIFT característiques altres objectes	43
Figura 39. Optical Flow roda ratolí.....	43
Figura 40. Gràfica Optical Flow roda ratolí 1.....	44
Figura 41. Gràfica Optical Flow roda ratolí 2.....	45

Figura 42. Eina marcatge manual vibrisses	46
Figura 43. Frame Resultats Vibrisses 1.....	47
Figura 44. Frame Resultats Vibrisses 2.....	48
Figura 45. Frame Resultats Vibrisses 3.....	49
Figura 46. Frame Resultats Vibrisses 4.....	50
Figura 47. Frame Resultats Vibrisses 5.....	51

1. Elecció del tema del TFM

El tema escollit pel TFM és el seguiment de dues parts de la anatomia dels ratolins, les potes i les vibrisses del musell (col·loquialment anomenades bigotis). Aquest projecte permet estudiar l'obtenció de dades motrius a partir d'un conjunt de gravacions de ratolins. Aquest procés s'engloba dins del camp de visió artificial i inclou processos de detecció d'objectes a imatges, seguiment d'objectes i tractament d'imatge entre d'altres. Addicionalment el projecte permet contribuir a l'estudi dels experiments optogenètics que s'estan realitzant en els vídeos, estudiant com automatitzar els processos de recollida de dades motrius dels subjectes.

El principal objectiu del projecte, és realitzar recerca sobre tècniques i algoritmes que permetin resoldre total o parcialment la necessitat de seguiment.

2. Objectius del TFM

El present TFM té per objectiu aplicar tècniques d'intel·ligència artificial per realitzar el seguiment de les extremitats dels ratolins i les vibrisses del seu musell. Aquest objectiu es deriva de la necessitat per part dels realitzadors d'experiments optogenètics de registrar els moviments dels ratolins. Com a fites concretes s'han fixat les següents:

- Estudiar i avaluar diferents mecanismes de visió artificial i la seva viabilitat per resoldre el problema tractat.
 - Investigar i avaluar quins mecanismes es poden usar per realitzar una detecció estàtica (a una imatge única) de les potes. Per la detecció de les potes es consideren important tres parts l'espatlla, el colze i el peu.
 - Investigar i avaluar quins mecanismes es poden usar per realitzar una detecció estàtica (a una imatge única) de les vibrisses.
 - Investigar i avaluar quins mecanismes es poden usar per realitzar el seguiment de les potes en un vídeo.
 - Investigar i avaluar quins mecanismes es poden usar per realitzar el seguiment de les vibrisses en un vídeo.
- Investigar quines millores en la captació dels vídeos es poden dur a terme per facilitar la detecció i el seguiment.
- Desenvolupar amb un llenguatge de programació els processos que usant un o més algoritmes, ofereixin resultats més satisfactoris.

Adicionalment, el present document pretén servir de guia per a futures investigacions i desenvolupaments. Proporcionant un relat de tot el estudi realitzat i diverses eines desenvolupades al llarg del treball.

3. Introducció

En aquest apartat es tracta de detallar els conceptes relacionats amb el projecte necessaris per a poder entendre'l correctament.

3.1 Visió artificial

La visió artificial o visió per computador, és un subcamp de la intel·ligència artificial. El propòsit d'aquesta àrea és programar un computador per a poder interpretar elements d'una imatge.

Els objectius clàssics de la visió artificial són:

- La detecció i localització de certs objectes d'una imatge.
- Seguiment d'un objecte en una seqüència d'imatges.
- Mapeig d'una escena per generar un model 3D.
- Estimació de les postures tridimensionals dels humans.
- Cerca d'imatges digitals.

Aquests objectius s'aconsegueixen per mitjà de reconeixement de patrons, aprenentatge estadístic, geometria de projecció, teoria de grafs o processament d'imatge entre d'altres. Dels objectius de la visió artificial, aquest projecte persegueix els dos primers esmentats, la detecció de les potes i les vibrisses en imatges estàtiques i posteriorment la capacitat de realitzar-ne el seguiment al llarg d'un vídeo.

3.2 Experiments optogenètics

La simulació optogenètica és un mètode útil per a l'estudi del sistema motor dels rosegadors. La disponibilitat d'opsines (grup de molècules responsables de les reaccions cel·lulars, involucrades en la percepció de la llum) inhibidores i excitatòries permet en subpoblacions neuronals selectives, avaluar la contribució de punts corticals individuals al moviment.

Durant el procés de mapeig motor basat en llum (LBMM), la regió cortical que per exemple representa els músculs de les potes davanteres, pot ser identificada mitjançant una estimulació seqüencial de diferents punts corticals i emmagatzemant les respostes generades.

El experiments tractats en aquest projecte tracten de monitoritzar les reaccions en vibrisses i potes, a les estimulacions de diferents punts corticals.

3.2.1 Optogenètica

L'optogenètica utilitza la llum per a controlar neurones que s'han modificat genèticament per ser més sensibles a la llum. És una tècnica de neuromodulació usada en neurociències que barreja tècniques òptiques i genètiques per controlar i monitoritzar les activitats de neurones individuals en teixits vius. En alguns casos, com al experiment que es tracta en aquest

projecte, s'apliquen sobre animals que es poden moure per mesurar amb més precisió els efectes sobre les diferents neurones en temps real.

3.3 Anatomia ratolins

3.3.1 Potes

Els ratolins tenen quatre potes i són quadrúpedes, les potes anteriors són curtes i amb quatre dits. Les posteriors són més llargues i amb cinc dits. A més són animals plantígrads que vol dir que posen a terra completament la planta del peu alhora de caminar.

La part superior de les potes presenten una marcada separació entre el cúbit i el radi. Aquestes potes tenen les falanges dels dits mitjanament desenvolupades, provocant que els dits es puguin creuar poc o res.

La part superior de les potes és de la mateixa coloració que la resta del pelatge, mentre que la part inferior al no tenir pel pren un color rosat.

3.3.2 Vibrisses

Les vibrisses són un tipus de pels especialitzats que posseeixen alguns animals. Generalment en els mamífers a mode de bigotis, i s'usen com un element sensorial tàctil. Les vibrisses tenen una arrel molt sensible que els permet detectar corrents d'aire el que juntament amb el olfacte permet detectar el origen de les olors.

Al entrar en contacte amb els objectes del entorn les vibrisses ajuden a percebre les distancies dels obstacles.

4. Planificació

4.1. Taula de fites

En aquest apartat es presenta una taula amb les diferents tasques realitzades i les dates d'inici i finalització. S'ha calculat una dedicació mitjana de 25h setmanals.

Fita	Durada	Inici	Fi
Elecció del tema del TFM	5h	24/02/2014	5/03/2014
Planificació del TFM	10h	5/03/2014	12/03/2014
Preparació	43h	17/03/2014	31/03/2014
Ampliació coneixements sobre el domini del TFM	11h	17/03/2014	24/03/2014
Estat de l'art dels algorismes i les eines a aplicar. Tria de les eines i algorismes adients.	32h	17/03/2014	31/03/2014
Execució	152h	31/03/2014	26/05/2014
Aplicació d'algorismes per a la detecció de les potes dels ratolins	50h	31/03/2014	19/05/2014
Test i avaluació dels resultats proporcionats pels algorismes aplicats sobre les potes	20h	19/05/2014	04/06/2014
Aplicació d'algorismes per a la detecció de les vibrisses dels ratolins	62h	31/03/2014	19/05/2014
Test i avaluació dels resultats proporcionats pels algorismes aplicats a les vibrisses dels ratolins	20h	19/05/2014	04/06/2014
Redacció de la memòria	40h	5/03/2014	04/06/2014
Preparació de la presentació i de la defensa	20h	26/05/2014	04/06/2014
TOTAL:	270 h	24/02/2014	04/06/2014

4.2.Lliurables

Fase	Lliurable	Data
PAC1 - proposta	Descripció, objectius i pla de treball temporalitzat	12/03/2014
PAC2 - primera versió	Part de la memòria explicant: <ul style="list-style-type: none">○ Introducció a l'àmbit del projecte i a les eines/conceptes relacionades.○ L'anàlisi dels algoritmes de tracking existents.○ Resultats dels primers algoritmes aplicats.	09/04/2014
PAC3 - resultats	<ul style="list-style-type: none">● Ampliació part de la memòria explicant els algoritmes i eines aplicades.● Resultats obtinguts amb els algoritmes avaluats fins al moment.	07/05/2014
Entrega final	<ul style="list-style-type: none">● Memòria, incloent la part de resultats, conclusions i possibles ampliacions a realitzar.● Implementacions dels algoritmes que proporcionin resultats útils.● Vídeo de presentació del projecte.	04/06/2014

4.3. Diagrama de Gantt



Figura 1 Diagrama de Gantt

5. Preparació

Aquest apartat té per objectiu recollir tots els passos previs a l'estudi i aplicació dels algoritmes sobre les imatges dels ratolins. Des de l'elecció d'eines fins a l'estudi dels algoritmes que s'usaran posteriorment.

5.1 Eines

5.1.1 Open CV

OpenCv(Open Source Computer Vision Library) és una llibreria de funcions de programació enfocada a processament d'imatges en temps real, desenvolupada inicialment per Intel i actualment, amb el suport de Willow Garage i Itseez, sota la llicència open source BSD license. La llibreria és lliure tant per ús comercial com acadèmic.

La versió 1.0 va ser llançada l'any 2006 i l'última versió és la 2.4.9 que data del 25 d'abril de 2014.

OpenCv esta disponible per a C++, C , Python i Java, però existeixen wrappers que permeten utilitzar les llibreries des d'altres llenguatges.

S'ha seleccionat OpenCv a causa de que és una de les llibreries més esteses per tractament d'imatge i disposa d'una gran quantitat d'algoritmes ja implementats el que permet estudiar-ne una major quantitat en menys temps.

Les funcions que ofereix OpenCv es poden classificar en el següents grups:

- **Funcions de interfície gràfica d'usuari:** Permeten crear interfícies per a visualitzar o modificar imatges i vídeos.
- **Funcions de processament d'imatge:** Aquest grup inclou funcions com cerca de contorns, transformacions geomètriques, canvis d'espai de color o binarització.
- **Detecció de característiques:** Detectors de cantonades, algoritmes de SIFT (Scale-Invariant Feature Transform), SURF (Speeded-Up Robust Features), BRIEF (Binary Robust Independent Elementary Features) i ORB (Oriented FAST and Rotated BRIEF).
- **Anàlisi de vídeo:** Tracking basat en color , Optical Flow o subtracció del fons.
- **Calibració de càmeres i reconstrucció 3D:** Disposa d'algoritmes de calibració de càmeres o d'extracció de profunditat a partir d'imatges 2D.
- **Aprenentatge automàtic:** Conté algoritmes de classificació com KNN (K-Nearest Neighbour), SVM (Support Vector Machines) o K-Means Clustering.
- **Detecció d'objectes:** Engloba algoritmes com els filtres de Haar.

5.1.2 Python

Python és un llenguatge de programació de propòsit general, amb una ampla base d'usuaris. Python posa especial èmfasi en la facilitat de lectura del codi i la capacitat de poder programar amb menys línies de codi que en altres llenguatges.

Aquest llenguatge suporta varis paradigmes de programació com programació orientada a objectes, programació imperativa, funcional i procedimental. Addicionalment disposa d'un sistema de tipatge dinàmic, gestió automàtica de memòria i una gran llibreria de mètodes estàndard. Per últim cal remarcar que suporta diferents sistemes operatius com Linux, Windows o Mac.

Els motius de la tria d'aquest llenguatge de programació per a dur a terme el estudi dels algoritmes s'ha basat en diversos aspectes. Principalment s'ha valorat positivament el seu caràcter multiplataforma, rendiment, el suport natiu de OpenCv i la seva ampla utilització en l'àmbit de la intel·ligència artificial.

5.2 Algoritmes

5.2.1 Mètodes del valor llindar

Els mètodes del valor llindar són un grup d'algoritmes que tenen la finalitat de separar alguns elements d'una imatge de la resta. Aquesta separació es fa en base al valor en escala de grisos de cadascun dels píxels.

Generalment aquest mètode s'usa per binaritzar les imatges, procés que consisteix en que si el valor del píxel supera el llindar el píxel serà d'un color (blanc o negre) i si no el supera serà del altre color.

Els algoritmes de valor llindar usats es poden dividir en dos grups:

- **Valor llindar global:** S'usa un únic valor llindar per tota la imatge. Aquest algoritme és més senzill, però no és adequat per casos on per exemple hi hagi variacions d'il·luminació.

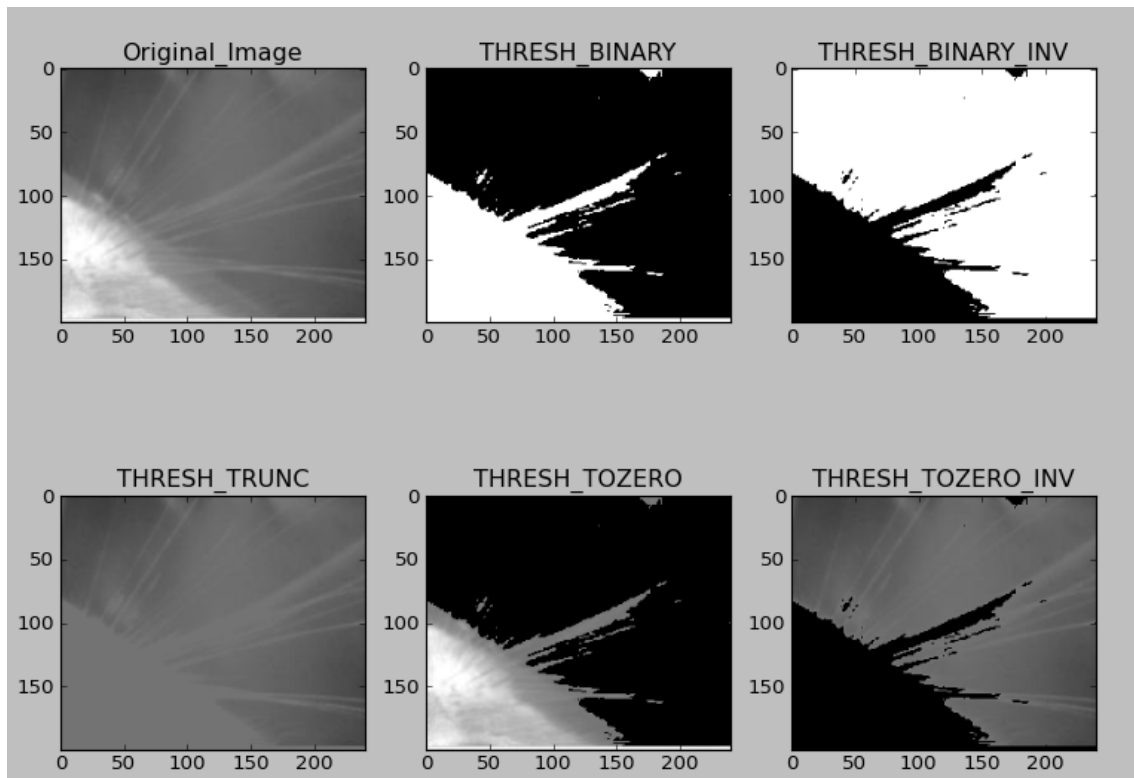


Figura 2. Valor Llindar Global

A la imatge anterior es poden observar els diferents tipus de binarització global que ofereix OpenCv. L'únic valor que es pot configurar en aquest tipus de binarització és el llindar que ha de ser un valor de 0 a 255 en el cas de la imatge és un valor de 115.

- **Valor llindar adaptatiu:** En aquest cas la imatge es tracta en diferents regions cadascuna amb un valor llindar diferent. Aquest tipus permet binaritzar millor les imatges amb variacions en la il·luminació.

El mètodes de valor llindar adaptatiu tenen els següents paràmetres:

- **Tipus binarització:** Es pot escollir binarització normal i invertida.
- **Mida de bloc:** Mida del bloc de píxels veïns a tenir en compte per calcular el valor de binarització.
- **C:** Constant estreta de la mitjana calculada.

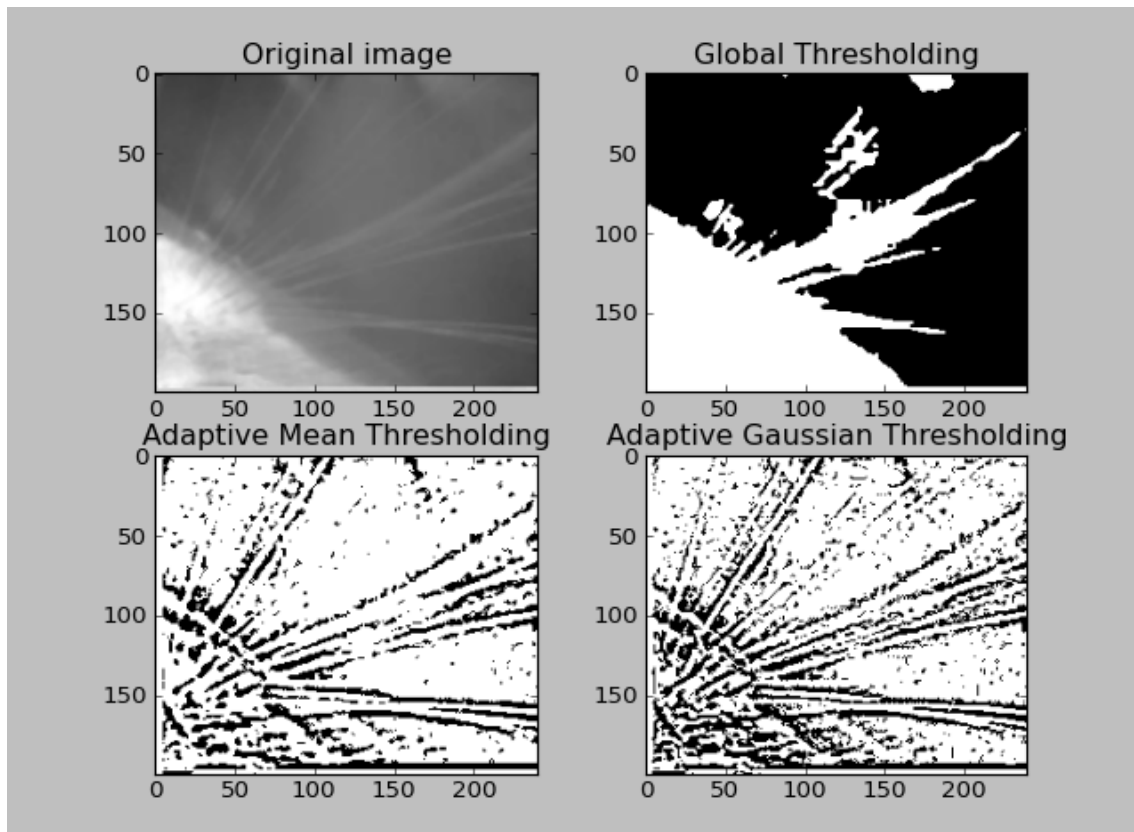


Figura 3. Valor Llindar Adaptatiu

A la imatge superior es pot observar la imatge original, el resultat de la binarització amb un valor de llindar global i dos mètodes de valor llindar adaptatiu. Aquests mètodes són:

- El valor llindar mitjà, el valor llindar es calcula a partir de la mitjana de píxels d'una zona.
- El valor llindar gaussià, el valor llindar s'obté com la suma ponderada dels valors dels píxels d'una zona. Les ponderacions dels diferents elements venen donats per una finestra gaussiana.

5.2.2 Mascarees i convolucions

Les mascarees són usades en visió artificial per realitzar operacions amb imatges. El procés d'aplicar les mascarees sobre una imatge s'anomena convolució. Aquest procés consisteix en que dues funcions es combinen per crear una funció resultant.

La primera funció es correspon a la mascara que es convoluciona sobre la segona funció, la imatge. La funció resultant serà una imatge amb valors modificats respecte l'original. Per combinar la mascara amb la imatge es multiplica el valor de cada píxel de la mascara pel seu corresponent valor de la imatge i el resultat és la imatge convolucionada.

5.2.2.1 Filtres de HAAR

Els filtres de haar usen mascarees i convolucions sobre diferents regions de les imatges. Els resultats d'aquestes convolucions permeten categoritzar subseccions de les imatges.

La detecció per filtres de Haar esta constituïda per dues fases:

- **Aprenentatge:** en aquesta fase l'algoritme necessita rebre imatges on l'objecte que es vol detectar estigui present i imatges en les que el objecte no estigui present. Aquest entrenament permet obtenir el classificador que permet distingir el objecte desitjat de la resta d'elements. Es recomana per objectes complexos usar una quantitat d'imatges de l'ordre de centenars o varis milers.
- **Detecció:** es recorre tota la imatge amb els filtres i es realitza el càlcul del valor de cada subsecció. Seguidament es comparen els valor amb el classificador obtingut en la fase d'aprenentatge per separar les subseccions que contenen el objecte de les que no.

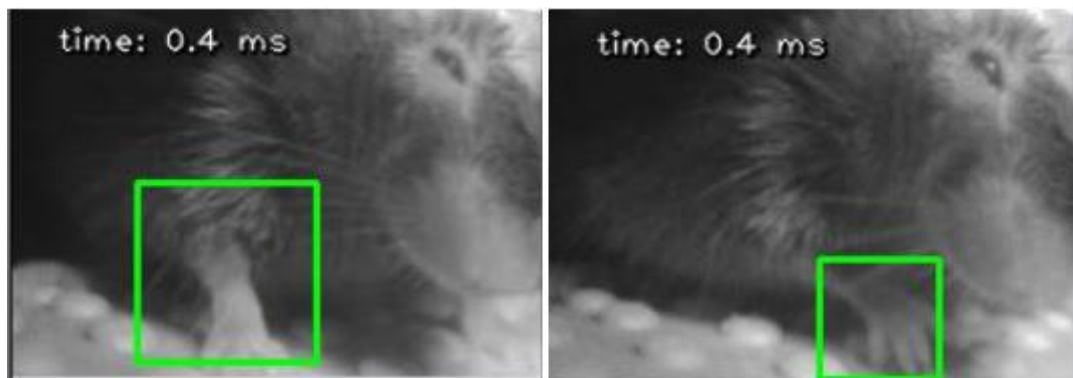


Figura 4 Filtres de Haar

En aquest mètode es poden configurar principalment tres paràmetres:

- **Factors d'escala:** El paràmetre especifica quant es redueix la imatge en cada escala.
- **Mínim nombre de veïns:** Determina quants veïns ha de tenir com a mínim dintre del conjunt de training un element de la imatge de test. Si s'augmenta aquest valor es redueix el nombre d'elements detectats però s'augmenta la precisió.
- **Mida mínima i màxima:** Permet determinar la mida mínima i màxima que poden tenir els elements candidats a ser l'objecte cercat.

5.2.2.2 Gradient

El gradient representa els canvis d'intensitat d'una imatge, aquests canvis es representen com un vector per cada píxel que defineix en quina direcció el canvi és més representatiu i amb quina intensitat. Aquesta tècnica esta dissenyada per trobar els contorns en una imatge.

Per aconseguir la informació del gradient es proporciona una mascara per cada dimensió existent. Depenent de la mascara els canvis entre intensitats poden ser més o menys importants.

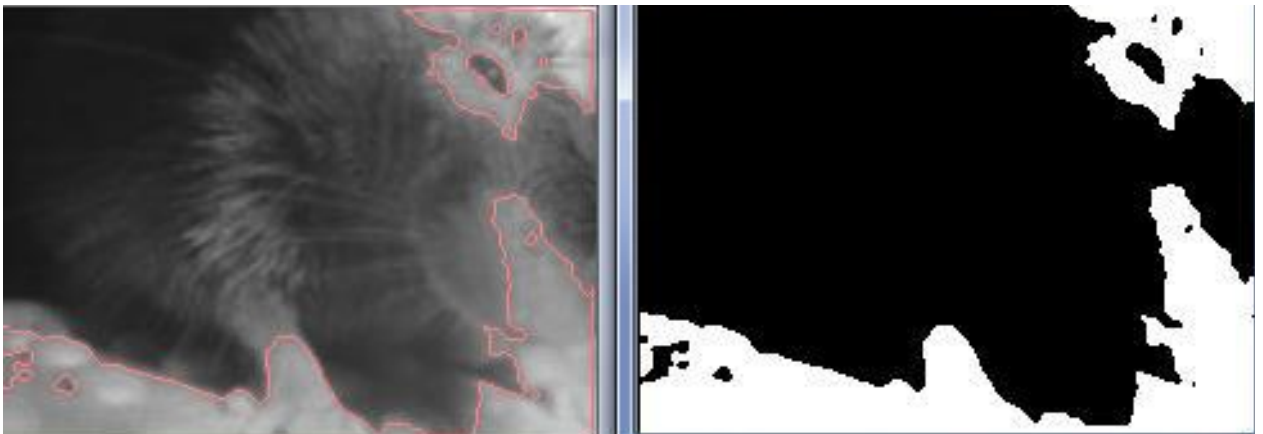


Figura 5 Gradient

A la imatge superior es pot observar a la dreta la imatge binaritzada i a l'esquerra els contorns extrems pel gradient.

Els paràmetres usats per calcular el gradient són:

- **Mode:** Aquest paràmetre defineix el mode de recuperació dels contorns. Existeixen els següents mètodes:
 - **CV_RETR_EXTERNAL:** Recupera únicament els perfils dels extrems externs.
 - **CV_RETR_LIST:** Recupera tots els contorns sense jerarquies.
 - **CV_RETR_CCOMP:** Recupera tots els contorns i els separa en una jerarquia de dos nivells.
 - **CV_RETR_TREE:** Recupera tots els contorns i crea una jerarquia de contorns encadenats.
- **Mètode:** Estableix el mètode d'aproximació dels contorns. Els mètodes disponibles són:
 - **CV_CHAIN_APPROX_NONE:** Emmagatzema absolutament tots els punts dels contorns.
 - **CV_CHAIN_APPROX_SIMPLE:** Redueix el nombre de punts del contorn ajuntant els punts propers.
- **Offset:** Paràmetre opcional, determina un valor pel qual tots els punts del contorn seran desplaçats.

5.2.2.3 Filtre de mitjana

El filtre de mitjana és un procés que realitza la mitjana entre els píxels de la imatge afectats per la mascara, aconseguint un efecte de suavitzat. Aquest efecte permet reduir el soroll o suavitzar els contorns d'una imatge. A la següent imatge es pot observar com la imatge de la dreta presenta molt menys detall que la de la esquerra ja que se li ha aplicat un valor alt de filtratge.

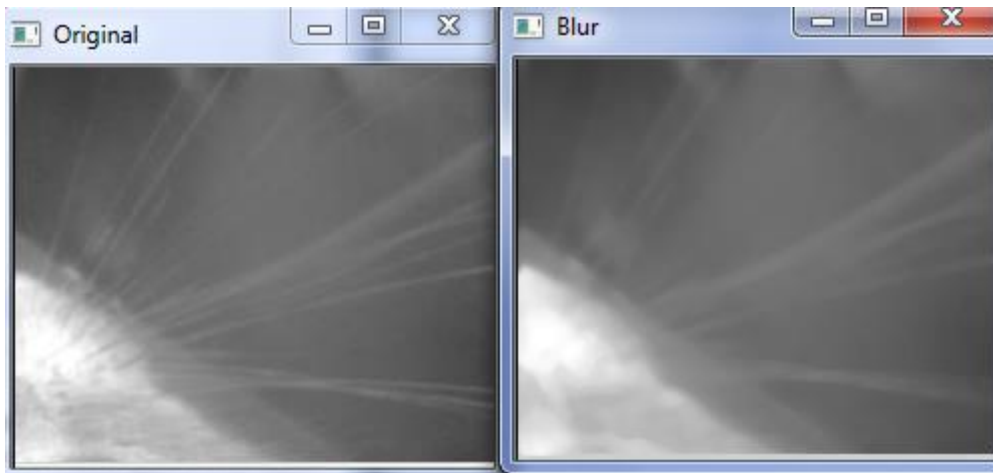


Figura 6. Filtre de mitjana

Aquest mètode té un únic valor modificable que és el nivell de blur, com més alt és aquest valor major és el nivell de suavitzat.

5.2.3 Transformada de Hough

La transformada de Hough té com a objectiu trobar figures geomètriques a una imatge. El seu ús clàssic és per la detecció de rectes a una imatge. Per detectar les rectes el que es fa és representar totes les rectes que passen per cadascun dels punts de les imatges i cercar les interseccions. Per cadascun dels punts s'obté una corba, si les corbes de diferents punts interseccionen vol dir que pertanyen a la mateixa recta.

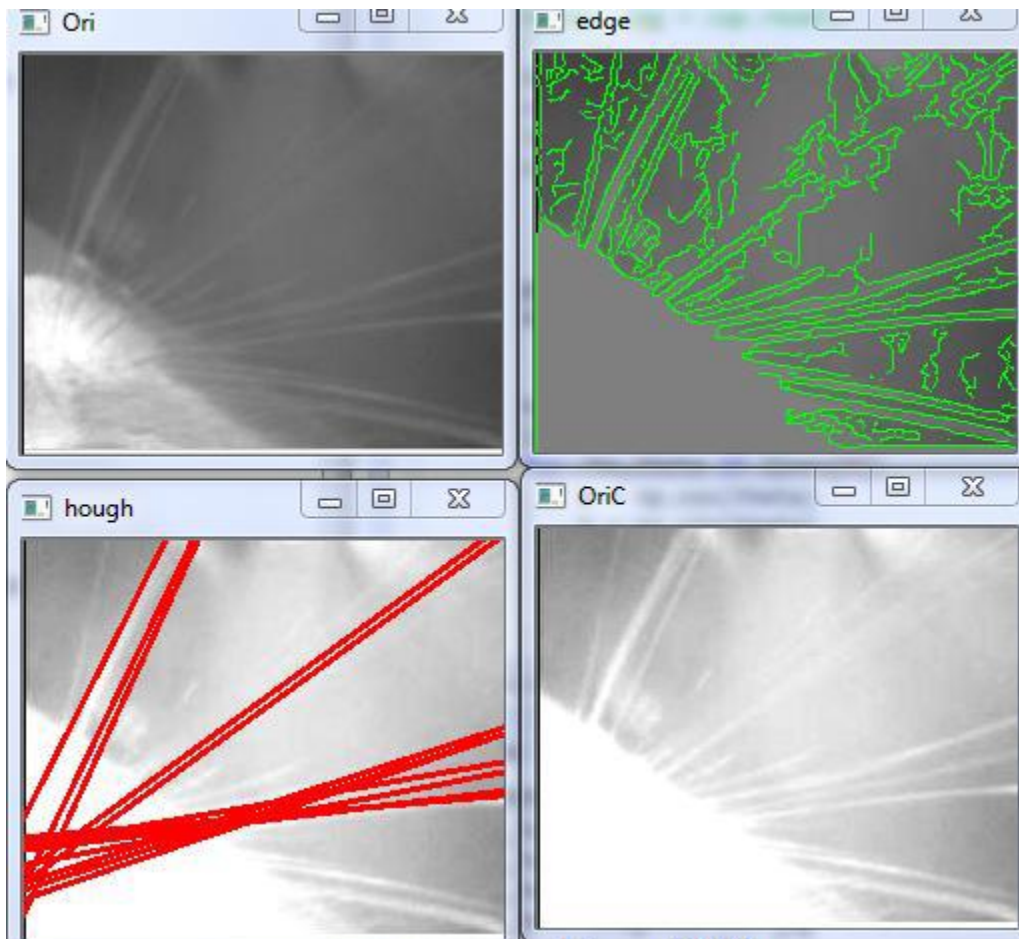


Figura 7 Transformada de Hough

La transformada de Hough depèn d'un únic paràmetre, aquest és el treshold. El threshold o llindar, defineix el valor mínim de interseccions per considerar que és una recta, com més alt és el valor llindar menor nombre de rectes s'obtidran.

5.2.4 Scale-invariant feature transform

Scale-invariant feature transform (SIFT) és un algoritme de visió per computador amb l'objectiu de detectar característiques locals d'una imatge. Aquest algoritme va ser publicat per David Lowe a l'any 1999.

Per qualsevol objecte d'una imatge es poden extreure els punts interessants per aconseguir una descripció de les característiques del objecte. Una vegada extreta aquesta descripció es pot tractar de buscar aquestes característiques en una imatge de test per d'aquesta manera detectar l'objecte.

L'algoritme SIFT pot identificar objectes amb una oclusió parcial, canvis d'escala, orientació, il·luminació i distorsió.

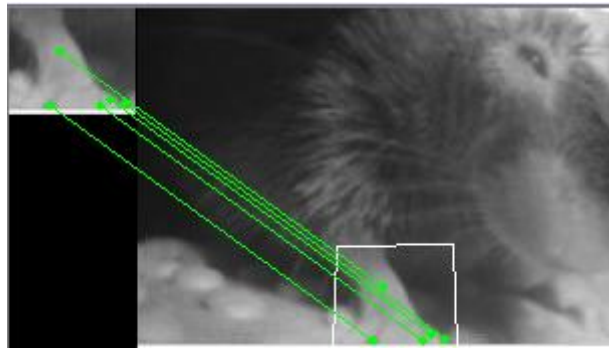


Figura 8 SIFT

Els paràmetres d'aquest algoritme són els següents:

- **NFeatures:** Indica la quantitat de característiques que es poden recuperar, es recuperen les n millors.
- **NOctaveLayers:** El nombre de capes en cada octava. El nombre d'octaves es calcula automàticament a partir de la resolució de la imatge.
- **ContrastThreshold:** Valor per filtrar les característiques en zones de baix contrast. Com més gran és aquest valor menys característiques es detecten.
- **EdgeThreshold:** Valor per filtrar les característiques relacionades amb contorns. Com més gran és aquest valor més característiques es detecten.
- **Sigma :** El valor sigma de la gaussiana aplicada a la imatge.

5.2.5 Canny edge detector

Aquest operador és un detector de contorns que utilitza un algoritme de diverses fases per detectar un ampli ventall de contorns a les imatges. Aquest detector va ser desenvolupat per John F.Canny al 1986.

Les etapes d'aquest algoritme són les llistades seguidament:

- **Reducció de soroll:** Inicialment s'aplica una convolució amb un filtre gaussià per reduir el soroll mitjançant un efecte "blur".
- **Cerca del gradient d'intensitat de la imatge:** Un contorn pot existir en diferents direccions, per això el detector utilitza quatre filtres per detectar contorns verticals, horitzontals i diagonals.
- **Supressió dels no màxims:** En aquesta etapa s'eliminen els punts no desitjats dels contorns, usant una sèrie de regles en base als valors dels gradients.
- **Cerca de contorns a la imatge i filtratge per histèresi:** Els gradients amb una alta intensitat, és més probable que es corresponguin a contorns que els de baixa intensitat. A causa de la dificultat de filtrar per un valor concret de gradient per determinar si és un contorn, es realitza el filtratge per histèresi.

Una vegada finalitzades aquestes etapes s'obté una imatge binaritzada on els píxels blancs són els contorns i els negres els no contorns.



Figura 9 Canny Edges

Els paràmetres necessaris per usar aquest mètode són els següents:

- **Threshold1:** Primer valor de filtratge per el procés de histèresi.
- **Threshold2:** Segon valor de filtratge per el procés de histèresi.
- **Mida de obertura:** Mida de la obertura pel operador Sobel.
- **Gradient L2:** Indica si es vol usar una norma L2 més acurada, per calcular el gradient de la imatge que la usada per defecte.

5.2.5 Optical flow

Aquest mètode de visió per computador no té per objectiu identificar objectes sinó detectar els moviments dels píxels entre frames consecutius. Aquest sistema és útil per realitzar seguiment d'objectes entre d'altres aplicacions. L'optical flow es basa en dues premisses:

- La intensitat dels píxels d'un objecte no es modifiquen entre frames consecutius.
- Els píxels adjacents tenen un moviment similar.

Aquest mètode permet per tant identificar en quins punts de la imatge hi ha moviment apart d'identificar-ne el sentit i la direcció.

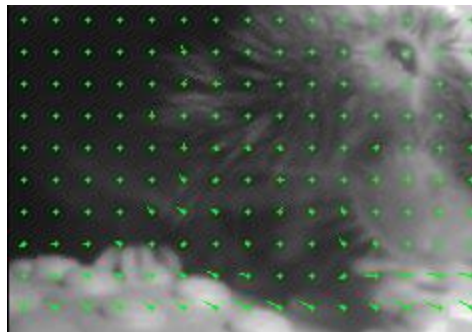


Figura 10 Optical flow

L'algoritme d'optical flow requereix els següents paràmetres:

- **Escala Pyr:** Paràmetre especificant l'escala de la imatge per construir piràmides d'aquesta. La escala pren valors entre 0 i 1, un valor de 0,5 és una piràmide clàssica on la mida de cada capa és la meitat de l'anterior.
- **Nivells:** Nombre de capes de la piràmide, incloent la capa de la imatge inicial. És a dir si el valor és 1, només s'usarà la imatge original.
- **Mida de finestra:** Si s'augmenta la mida de la finestra es millora la robustesa davant del soroll de les imatges i els moviments ràpids. D'altra banda produeix un camp de moviment més borrós.
- **Iteracions:** Nombre d'iteracions que realitza l'algoritme per cada nivell de la piràmide.
- **Poly_n:** Mida de les rodalies de cada píxel usades per a cercar l'expansió polinomial a cada píxel. Valors majors es tradueixen en que la imatge serà aproximada amb superfícies més suaus. Això resulta en un algoritme molt robust i un camp de moviment més borrós. Els valors més comuns són el 5 i el 7.
- **Poly_sigma:** Desviació estàndard de la Gaussiana usada per suavitzar les derivatives de la expansió polinomial. Per un valor de poly_n=5 es recomana el valor 1.1 i per poly_n=7 el valor 1.5.

6. Execució

En aquest apartat es detallen tots els algorismes analitzats al llarg del desenvolupament del projecte i els seus resultats. Per cada algorisme es justifica perquè s'ha utilitzat i/o descartat. La fase d'execució esta separada en dues parts diferenciades detecció de potes i detecció de vibrisses, atès que precisen d'un tractament diferent a causa de la seva naturalesa.

6.1 Detecció de potes

La detecció de potes inclou la detecció de tres punts clau el inici, el final de la extremitat i la articulació intermitja. A causa del seu color distintiu i la seva forma es decideix començar per la detecció del final de la extremitat, on es troben els dits i les urpes.

6.1.1 Fase 1

En la primera etapa es tria l'angle de visualització de les imatges entre diversos disponibles finalment s'opta per una visió lateral del ratolí. En la següent imatge es pot observar el angle triat.

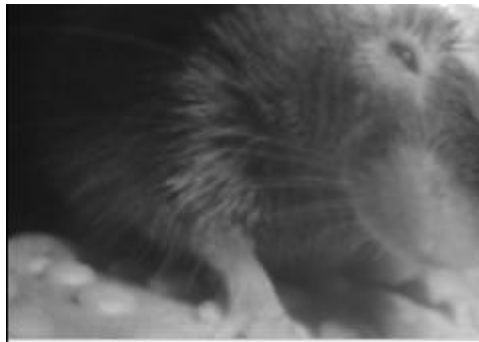


Figura 11. Angle imatge potes

Aquest angle no permet visualitzar correctament ni l'espatlla, ni el colze tot i així es considera el millor dels angles disponibles ja que el peu es diferencia de manera relativament clara. La resta d'angles disponibles són els mostrats tot seguit:



Figura 12 Potes angle superior

Com es pot veure a les imatges amb aquests angles és encara més difícil apreciar els moviments de la pota, ja que els moviments verticals són difícils de veure. A més el colze i l'espatlla són menys visibles que amb l'angle triat.

6.1.2 Fase 2

En aquesta segona etapa es tracta de trobar el final de l'extremitat com a base per trobar la resta de parts d'aquesta. Per aquest fi s'opta bàsicament per dos enfocaments:

Filtres de Haar

Inicialment es realitza la fase d'entrenament amb un conjunt de 100 imatges positives en les quals cal marcar les coordenades de les potes. Aquestes imatges s'han extret amb la mateixa quantitat de cadascun dels quatre vídeos disponibles amb l'angle triat (25 imatges per cada vídeo) i amb frames no consecutius per una major variabilitat. Per facilitar el procés s'ha creat una eina gràfica que permet seleccionar la localització de la pota, n'extreu les coordenades i les emmagatzema en el format esperat per OpenCv. La interfície d'aquesta eina es mostra a la següent imatge.



Figura 13. Generador Haar Correctes

Una vegada obtingut el conjunt de dades de localització per cadascuna de les imatges correctes, cal obtenir un conjunt d'imatges on no aparegui l'element. Aquestes imatges s'han extret dels propis vídeos dels ratolins però en angles en els que no era visible cap pota. La imatge mostrada a sota és un exemple de imatge negativa.

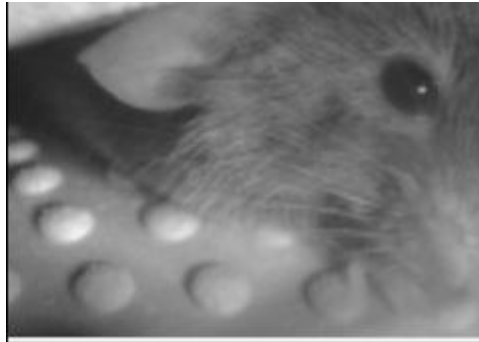


Figura 14. Imatge negativa Haar

Per últim es realitza el training sobre les imatges de mostra per obtenir un fitxer .xml, que permet realitzar la detecció sobre les imatges en les que es vol realitzar el seguiment.

Alhora de realitzar l'execució de l'algoritme es va haver d'augmentar el valor mínim de veïns necessaris per a considerar que és una pota, per evitar falsos positius. Addicionalment es defineix una mida mínima de la pota per evitar que es detectin per exemple els grànuls del terra. Després d'aquesta calibració es va aconseguir la correcta detecció en varis dels frames dels vídeos. Si bé, no en tots ja que en alguns moments la visibilitat de les potes es redueix molt.

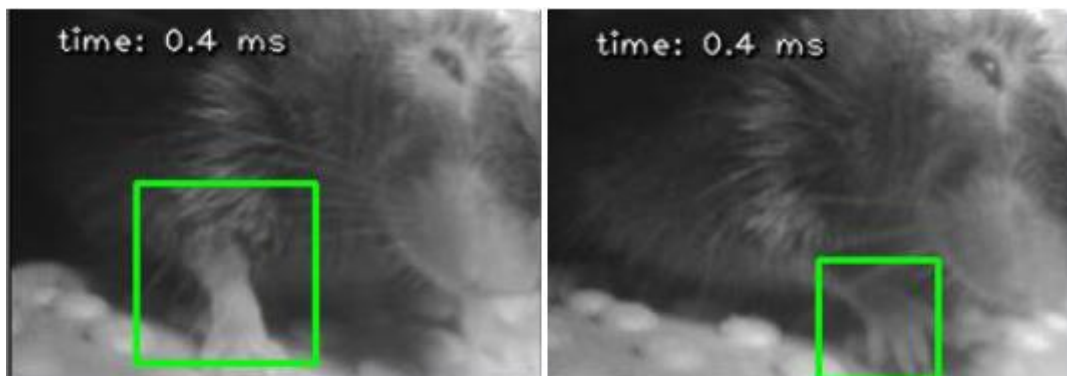


Figura 15. Imatges Haar Detecció Correcte



Figura 16. Imatges Haar Detecció Incorrecte

Cerca de contorns

Paral·lelament als filtres de Haar es va estudiar la cerca de contorns per a trobar la forma característica de les extremitats.

Aquest enfocament té un problema amb el pas previ a la cerca dels contorns, la binarització. Alhora de binaritzar resulta difícil separar els extrems de les potes del terra i la part superior de les potes de la resta del cos.

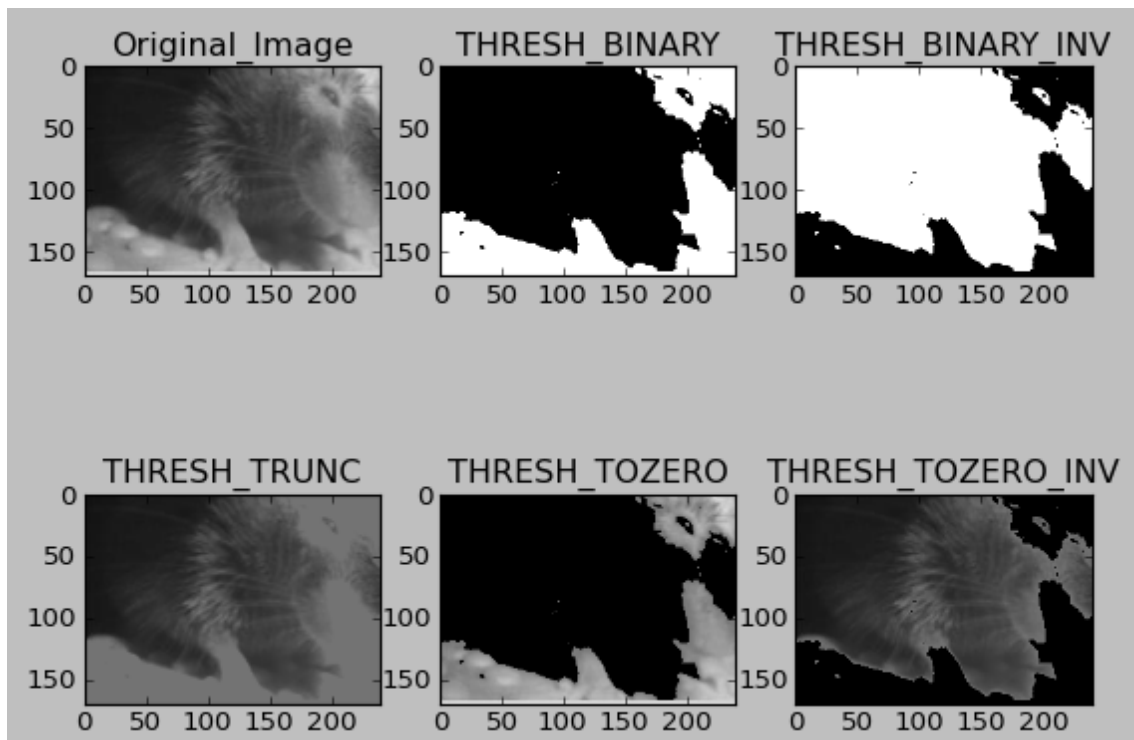


Figura 17. Binarització Global Potes

La imprecisa binarització mostrada a la imatge superior, provoca que sigui complex extreure informació útil dels contorns. Una extracció dels contorns a partir d'aquestes imatges es pot observar a continuació.

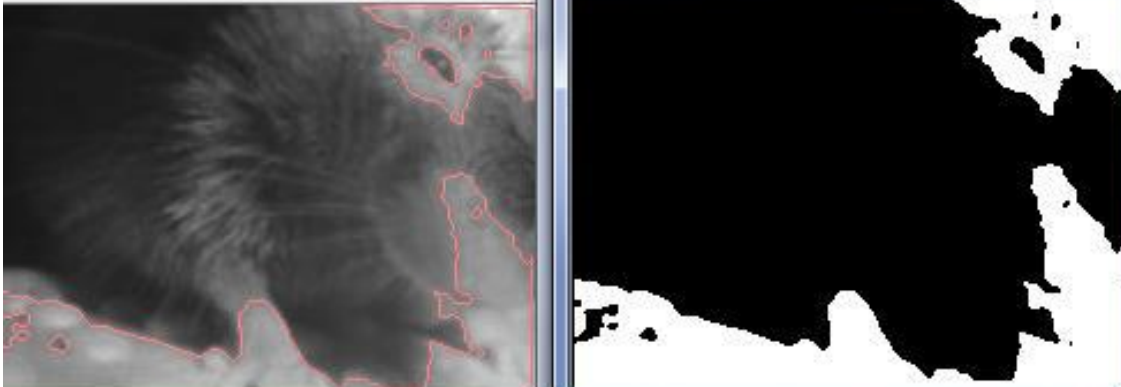


Figura 18. Contorns Potes Binarització Global

Es pot veure com l'extrem de la pota es binaritzava de color blanc i la resta de la pota de color negre. Però no es diferencia el peu del terra blanc obtenint uns contorns poc precisos. Si el terra fos fosc aquest sistema podria proporcionar resultats útils ja que gairebé tota la imatge es marcaria de color negre excepte el peu i una part del cap. Aproximant els contorns a polígons i avaluant aspectes com l'àrea o el nombre de vèrtexs es podria determinar quin contorn és la pota.

SIFT

Observant que amb els filtres de Haar s'assoleix un cert grau d'encert es decideix provar amb SIFT. SIFT és un detector basat en característiques, però teòricament amb uns nivells de precisió majors que els filtres de Haar. Aquesta millora de precisió ve donada per la capacitat de SIFT per suportar canvis d'escala, orientació, distorsió i oclusions parcials.

Per aplicar SIFT s'ha creat inicialment un conjunt d'imatges de training que són potes retallades de diferents imatges dels vídeos. Tot i que SIFT tolera diverses variacions en les imatges cal usar diferents potes ja que aquestes canvien de posició variant notablement les seves característiques. A continuació es presenten tres mostres de training.



Figura 19 Mostres training pota

Quan ja s'ha recopilat el conjunt de training, s'utilitza una modificació del algoritme de SIFT que realitza la comparació de cada frame del vídeo amb totes les mostres de training i dona per bona la comparació de menor distància. A més la distància es pondera amb el nombre de punts per donar preferència a aquelles comparacions que ofereixin més punts de coincidència detectats.

Per últim per evitar algunes de les deteccions errònies, que es produïen, es compara la distància entre la posició de la pota en el frame actual amb la posició de la pota detectat en el frame anterior. Es parteix del supòsit que entre dos frames consecutius pot haver-hi cert moviment de la pota però per exemple no pot passar d'un extrem a l'altre de la imatge.

Aplicant aquests mètodes s'aconsegueix cert grau de detecció, encara que dista de tenir una gran precisió. La granularia i el color del terra és el que més ho dificulta.

Adicionalment a causa de la baixa resolució de les imatges les potes tenen poc detall i com es pot veure a les imatges següents (Figura 20 i 21), el nombre de característiques detectat és molt baix. D'altra banda a la figura 22 es pot veure la diferència amb una pota en una major resolució.

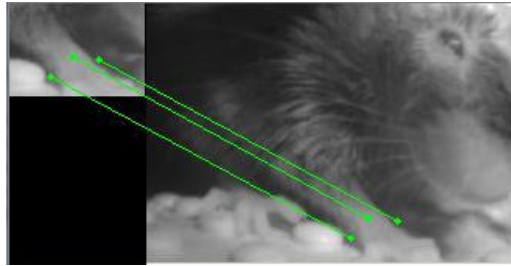


Figura 20 Detecció correcta SIFT



Figura 21 Detecció errònia SIFT



Figura 22 Potes alta resolució

6.1.3 Fase 3

Després d'observar que per culpa de les diverses limitacions exposades prèviament, no es podran aconseguir de moment resultats amb una bona fiabilitat. Es decideix buscar un altre enfocament.

Per això s'aplica la tècnica d'optical flow a la superfície sobre la que es troben els ratolins. Aquesta superfície es mou quan el ratolí tracta de caminar, d'aquesta manera es pot saber quan el ratolí camina endavant o endarrere i quan resta al mateix lloc. A la figura mostrada a sota es pot veure com l'algoritme detecta el moviment de la roda.

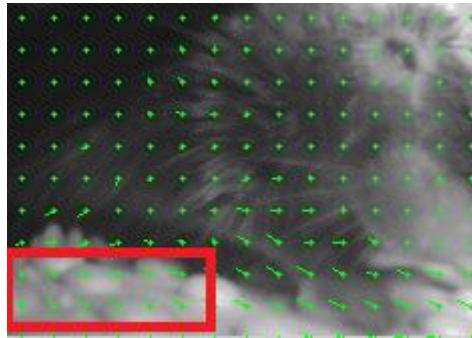


Figura 23. Optical Flow terra

Amb aquesta tècnica s'aconsegueix uns resultats basics però bastant precisos, tot i que es detecta que el sistema falla quan hi ha un frame de color negre intercalat al vídeo. Per això s'afegeix una validació, per no tenir en compte els valors de l'algoritme si el frame actual esta format únicament per píxels negres. Quan ja s'ha resolt aquest problema es genera una gràfica amb el següent format.

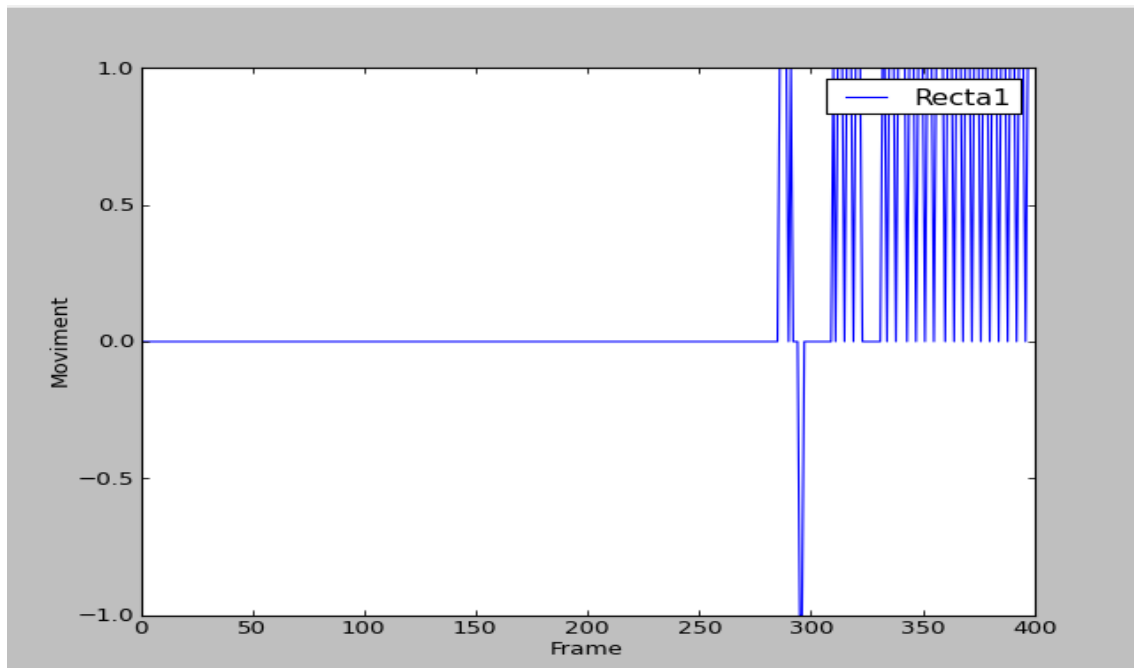


Figura 24. Gràfica Optical Flow

6.2 Detecció de vibrisses

Aquest apartat inclou la detecció de totes les vibrisses del ratolí i la capacitat de realitzar el seguiment d'una vibrissa concreta.

6.2.1 Fase 1: Tria angle imatges

En la primera etapa es tria l'angle de visualització de les imatges entre diversos disponibles. Finalment s'opta per una visió des de dalt de les vibrisses. A la següent imatge es pot observar el angle triat.



Figura 25. Angle imatge vibrisses

S'ha triat aquest angle perquè permet visualitzar els moviments horitzontals de les vibrisses i es disposa d'un fons uniforme encara que no proporciona un gran contrast amb les vibrisses.

A continuació es presenten altres angles disponibles:

-Angle lateral amb visualització parcial del cap, aquest angle s'ha descartat perquè les vibrisses són poc visibles i a més es poden confondre amb la resta de pelatge.

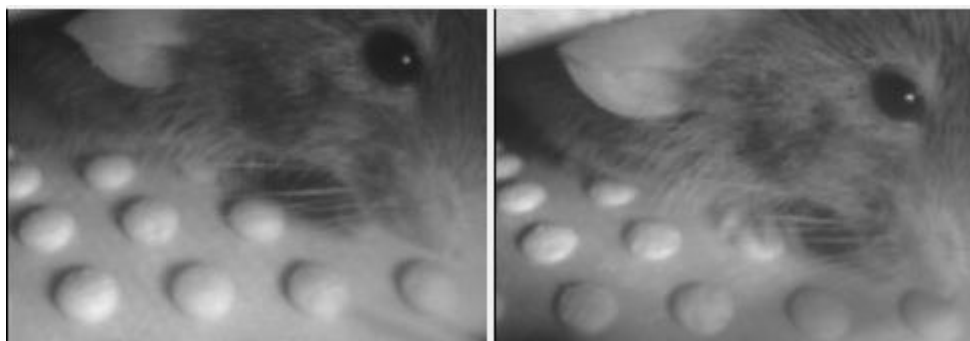


Figura 26 Vibrisses angle lateral

-Angle superior amb visualització parcial del cap, en aquest angle les vibrisses es diferencien millor de la resta de pelatge però tot i així són poc visible.



Figura 27 Vibrisses angle superior

6.2.2 Fase 2: Tractament de la imatge

6.2.2.1 Detecció de contorns

Considerant a la forma distintiva de les vibrisses s'ha decidit aplicar un detector de contorns. La seva forma, els agrupament i encreuaments que es duen a terme entre les vibrisses dificulten aplicar algoritmes com els filtres de Haar o SIFT.

Inicialment s'aplica els diferents tipus de binarització, com es mostra a la següent imatge.

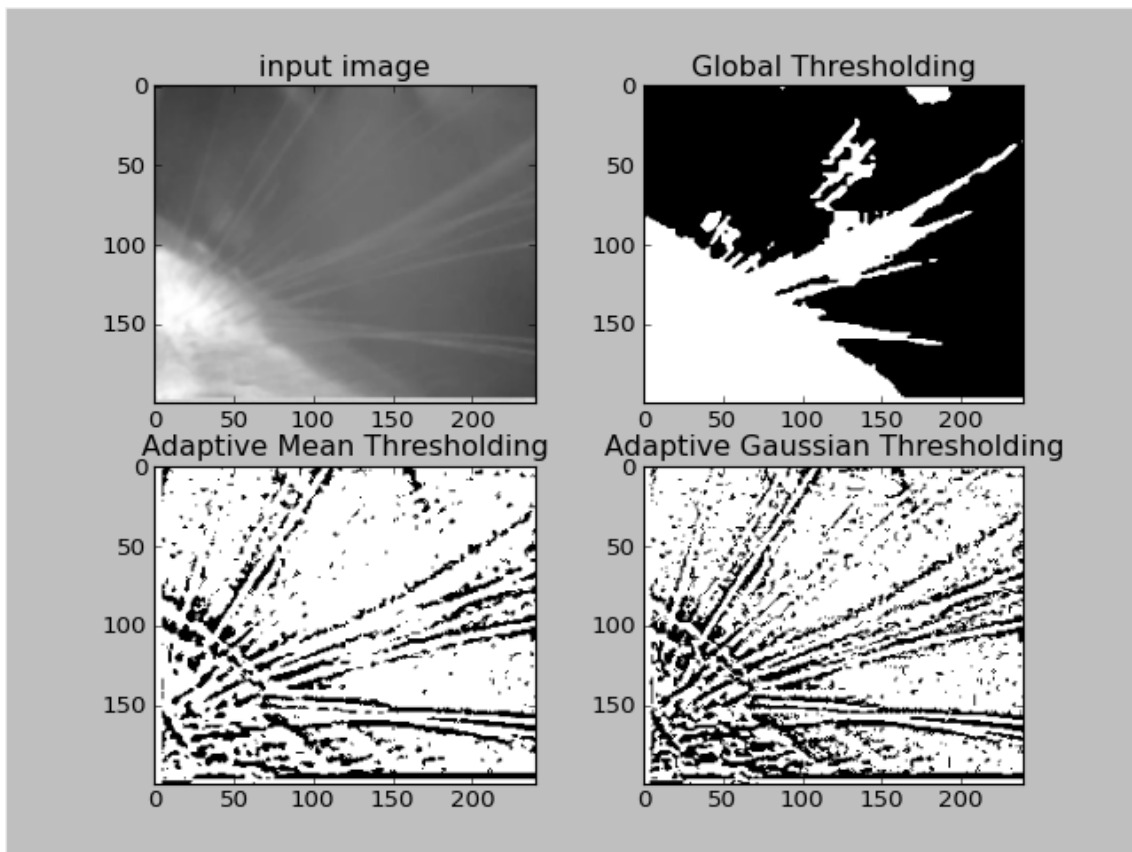


Figura 28. Binarització Adaptativa Vibrisses

Per últim s'aplica la cerca de contorns sobre les imatges binaritzades. La imatge global thresholding no obté bons resultats ja que la binarització no separa correctament les vibrisses del fons.

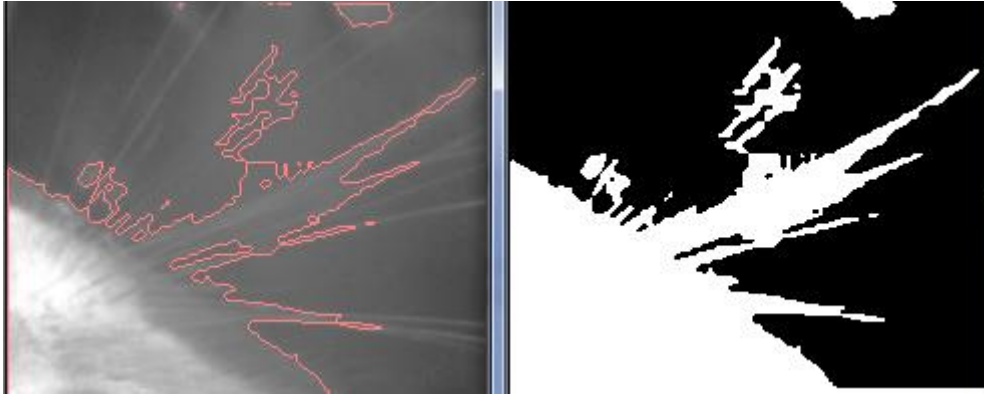


Figura 29. Contorns Binarització Global

D'altra banda al aplicar el mateix algoritme de contorns sobre les imatges adaptatives, tampoc s'aconsegueixen bons resultats. Això és causat pel soroll que tenen aquestes imatges, encara que la binarització adaptativa aconsegueix marcar les vibrisses amb una precisió molt major que la global.

A continuació es mostra la cerca de contorns sobre la adaptativa gaussiana i amb blur per reduir el soroll.

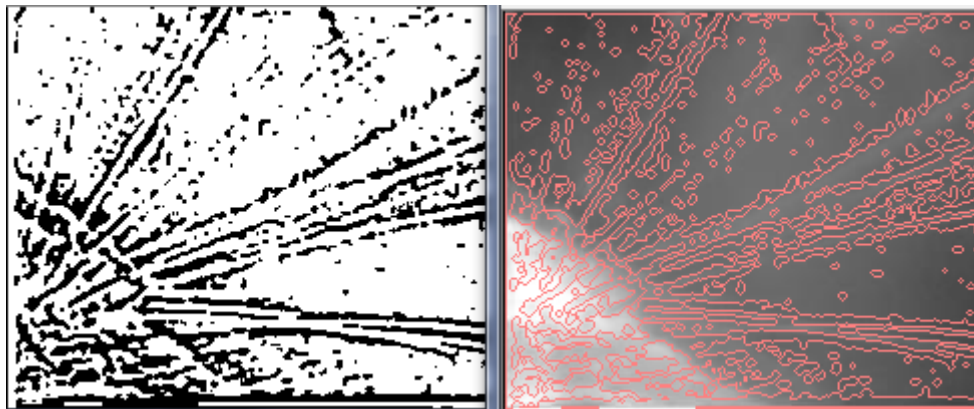


Figura 30. Contorns Binarització Adaptativa Gauss

6.2.2.2 Transformada de Hough

Com que a les imatges de les vibrisses usades, aquestes són gairebé rectes, es va decidir usar un detector de rectes. Per a aplicar la transformada de Hough cal aplicar prèviament una binarització.

Inicialment es va aplicar la transformada de Hough sobre la binarització adaptativa de Gauss però finalment es va usar l'algoritme de "Canny Edges" que ofereix uns millors resultats. Per

millorar el resultat de "Canny Edges " inicialment s'augmenta el contrast de la imatge com es presenta a sota.

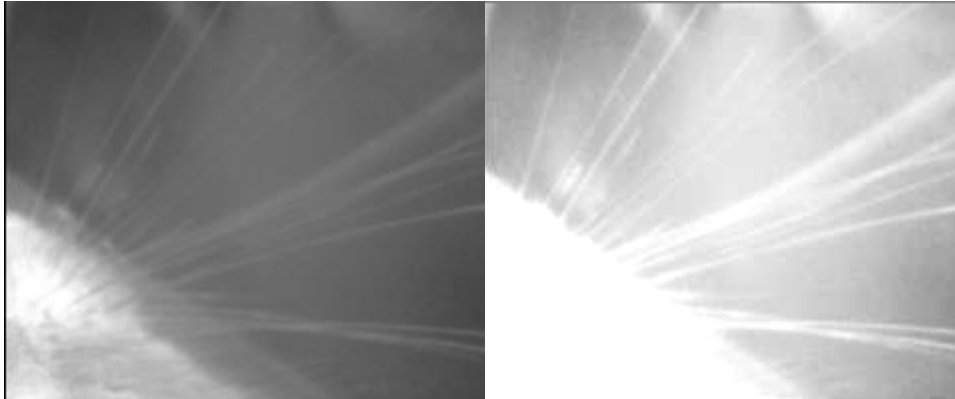


Figura 31 Vibrisses modificació contrast

Seguidament es mostra el resultat d'aplicar Canny sobre la figura anterior.

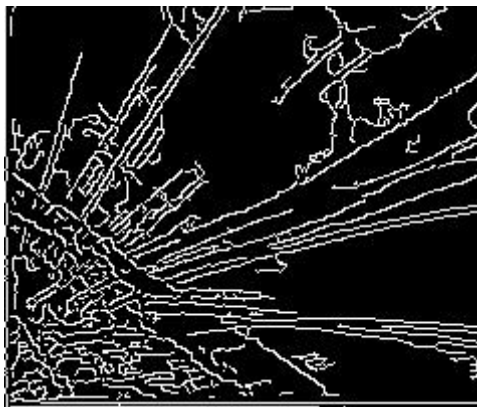


Figura 32 Canny edges

Tot seguit es va aplicar l'algoritme de la transformada de Hough sobre la imatge binaritzada amb Canny Edges. Amb aquest procés s'aconsegueix una detecció parcial de les vibrisses.

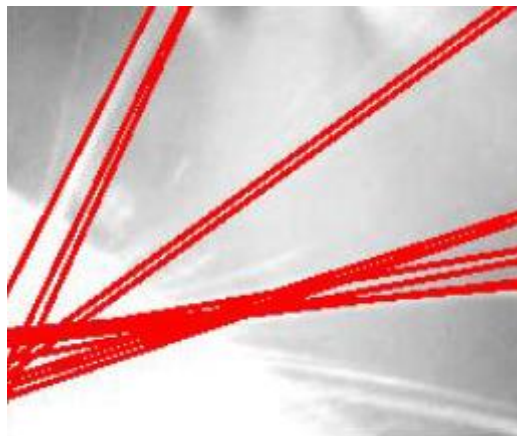


Figura 33 Transformada de Hough correcte

En algunes imatges es perden vibrisses ja que són poc visibles i no s'aconsegueixen separar del fons de la imatge. En altres imatges degut a la velocitat de moviment d'aquestes no es poden visualitzar correctament. En aquests casos com a la imatge presentada a continuació la detecció és nul·la.

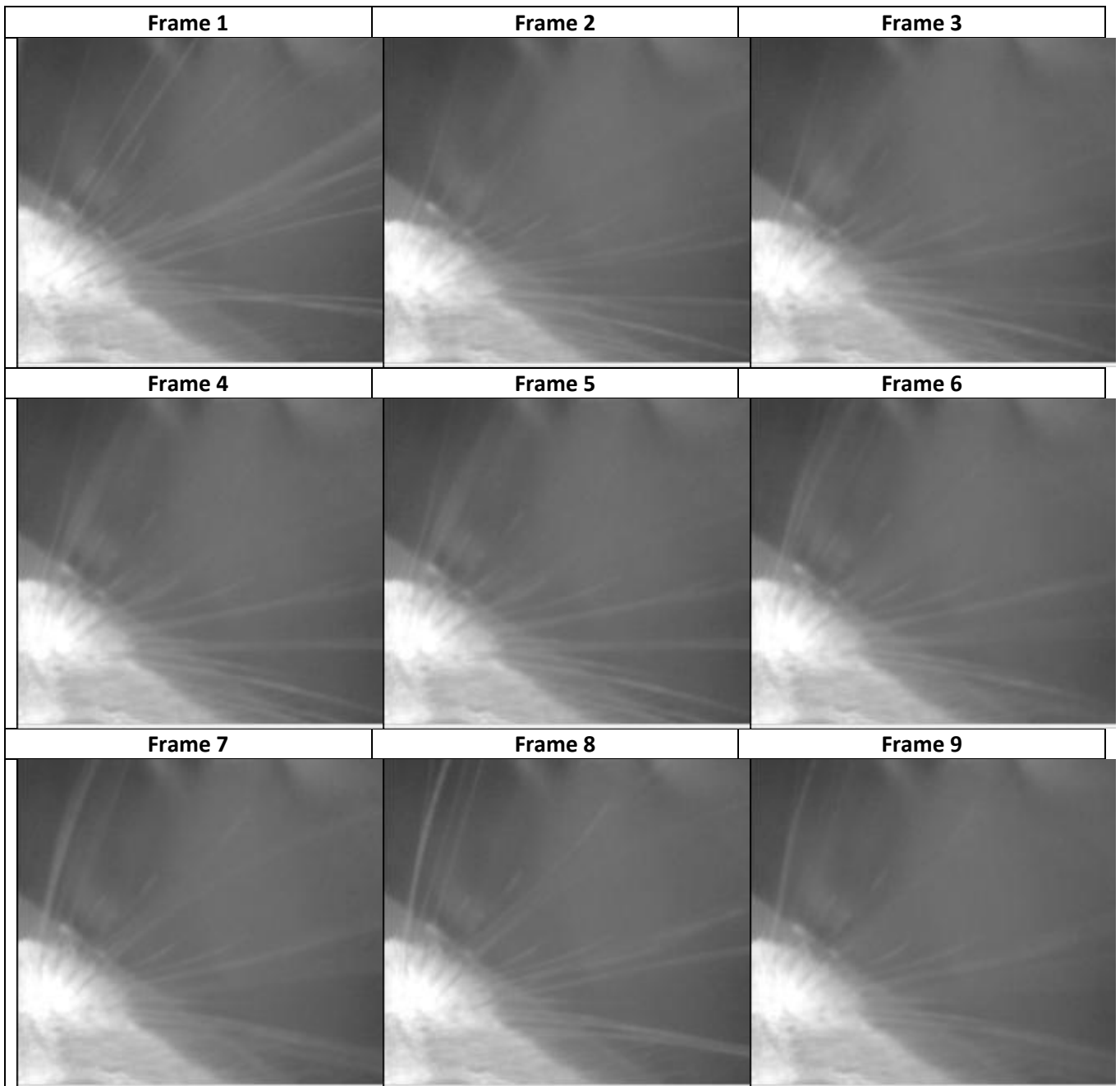


Figura 34 Transformada de Hough incorrecte

Tot i que amb la transformada de Hough s'aconsegueix detectar a cada frame on estan algunes de les vibrisses no es possible realitzar un seguiment del moviment de les vibrisses.

Aquesta impossibilitat es deu als motius presentats tot seguit:

1. Els frames per segon de captura de la càmera utilitzada (120 fps) són insuficients per la velocitat de moviment de les vibrisses. Això provoca que en algunes imatges les vibrisses apareguin borroses, i que entre dos frames consecutius el moviment sigui molt gran. En la següent taula es mostren nou frames consecutius, es pot apreciar que inclús a simple vista resulta difícil poder seguir el moviment d'una vibrissa.



2. Les vibrisses s'entrellacen i es creuen entre si, el provoca que mitjançant Hough sigui complicat diferenciar els grups de vibrisses de les vibrisses úniques.

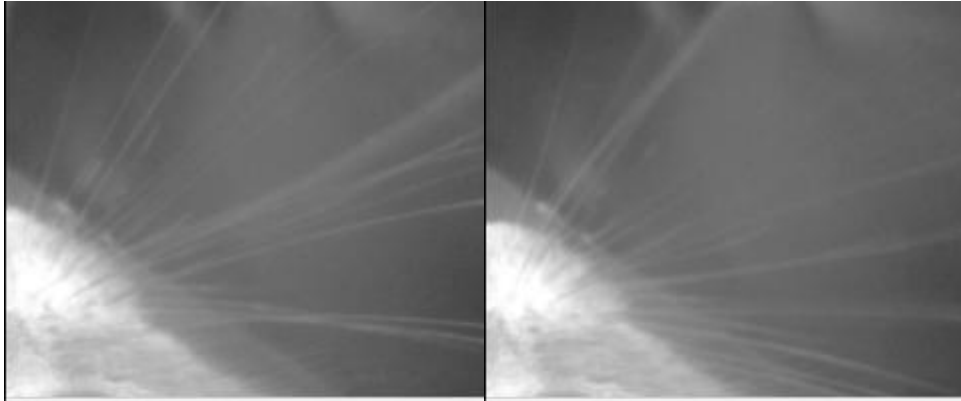


Figura 35 Vibrisses creuades

6.2.3 Fase 3

Després d'observar que per culpa de les diverses limitacions exposades prèviament, no es podran aconseguir de moment resultats amb una bona fiabilitat. Es decideix buscar un altre enfocament.

Per això es prova d'aplicar la tècnica d'optical flow, sobre els bigotis per determinar de manera global la direcció majoritària de les vibrisses. El resultat no és satisfactori, ja que no detecta el moviment de les vibrisses.

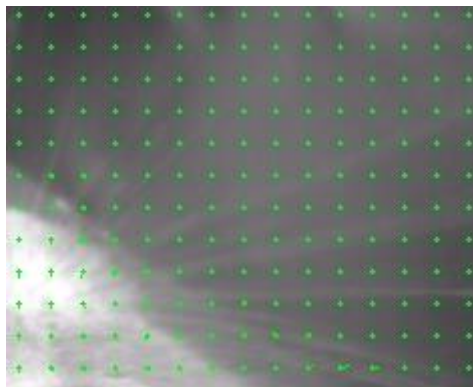


Figura 36. Vibrisses Optical Flow

Modificant el contrast de la imatge i tornant a aplicar Optical flow milloren els resultats captant alguns moviments. Però atès el baix rati d'imatges per segon, es considera que només es pot determinar si els bigotis es mouen o no. La informació de la seva direcció no és considerada totalment fiable. Això es causat perquè com s'ha comentat prèviament en varis frames es perden els bigotis.

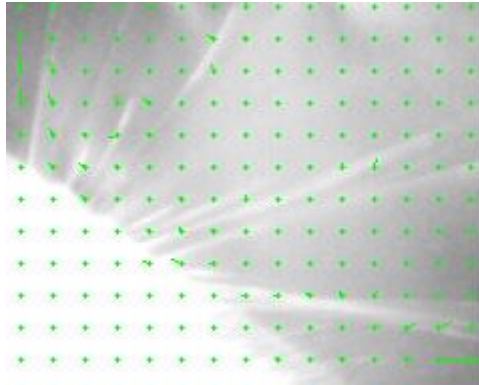


Figura 37. Vibrisses Optical Flow

7. Resultats

En aquest apartat s'avaluen els resultats obtinguts pels algoritmes més importants usats al llarg del projecte.

7.1 Resultats potes

Per poder avaluar la taxa d'encert dels algoritmes de Haar i SIFT alhora de detectar les potes, i a causa de la impossibilitat de trobar un sistema automàtic que pugui validar els resultats. S'ha desenvolupat un sistema de validació semiautomàtic. Aquest sistema es basa en que primerament de forma manual, es marca a cadascun dels frames la posició de la pota i s'emmagatzemen les coordenades a una base de dades. Posteriorment es compara aquests valors amb l'àrea seleccionada per l'algoritme. Per considerar que l'algoritme ha encertat, cal que el rectangle seleccionat manualment i el rectangle trobat per l'algoritme interseccionin com a mínim un 50%.

Inicialment s'ha valorat l'algoritme de Haar amb dos vídeos i amb diferents combinacions dels seus paràmetres, aquests paràmetres s'han descrit en apartats anteriors. El resultat és un percentatge d'encert en base als 400 frames que componen cadascun dels vídeos. L'algoritme de Haar s'ha entrenat únicament amb 100 imatges positives extretes dels 4 vídeos disponibles, per culpa de la petita quantitat de mostres.

Vídeo 1

Scale Factor	Num neighbours		
	5	10	15
1.3	78%	81%	75%
1.5	80%	76%	72%
1.7	79%	77%	72%

Vídeo 2

Scale Factor	Num neighbours		
	5	10	15
1.3	58%	64%	64%
1.5	52%	50%	38%
1.7	56%	62%	25%

El primer vídeo obté uns resultats millors, ja que en els primers frames d'aquest el moviment és gairebé nul. A més en el segon vídeo, en diversos moments la part inferior de la pota se surt del pla de la càmera.

Posteriorment, s'ha fet la prova sobre els mateixos vídeos amb l'algoritme de SIFT. A continuació es presenten els resultats:

Vídeo 1

NFeatures	Contrast thresholding	Sigma	Percentatge d'encert
15	0.07	16	76%
15	0.07	18	81%
15	0.09	16	73%
15	0.09	18	79%
20	0.07	16	77%
20	0.07	18	81%
20	0.09	16	73%
20	0.09	18	79%

Vídeo 2

NFeatures	Contrast thresholding	Sigma	Percentatge d'encert
15	0.07	16	11%
15	0.07	18	17%
15	0.09	16	1%
15	0.09	18	5%
20	0.07	16	11%
20	0.07	18	64%
20	0.09	16	1%
20	0.09	18	5%

Es pot observar que els millors resultats de l'algoritme de SIFT són iguals que els millors de HAAR tot i ser teòricament un algoritme més complert. A més, en el vídeo 2 la majoria de resultats són pitjors, es considera que aquest problema es deu a que SIFT cerca característiques de l'interior de l'objecte com al exemple de sota.

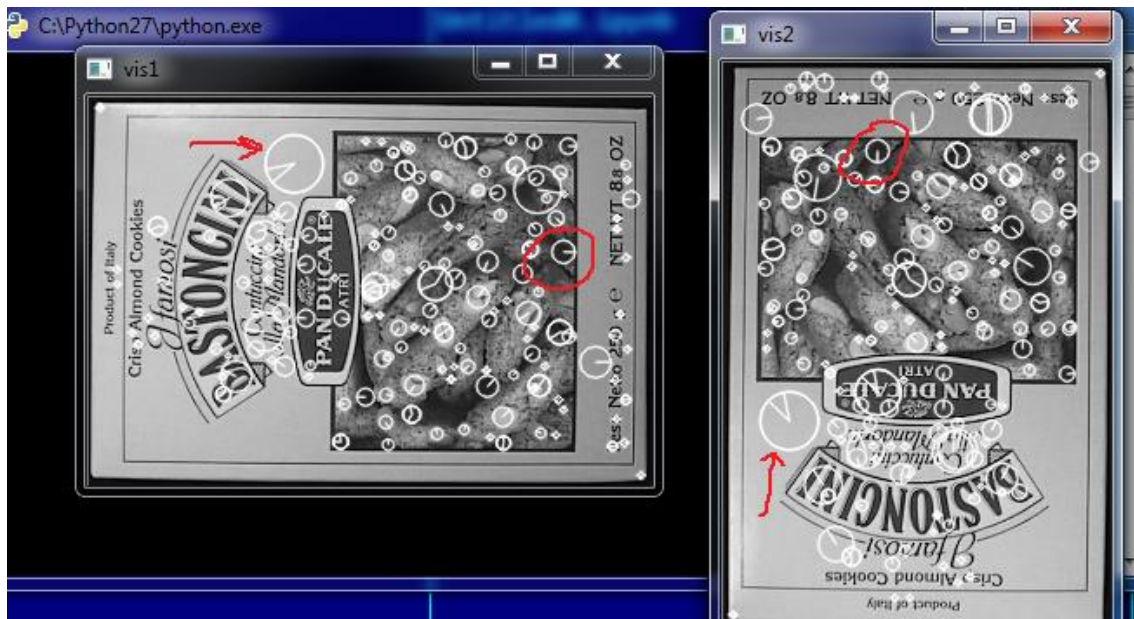


Figura 38. SIFT característiques altres objectes

En canvi a les potes, a causa de la baixa resolució no detecta gairebé cap característica clara, el que provoca que el nivell de detecció no sigui òptim.

Per finalitzar els resultats de les potes, es presenta els resultats d'aplicar l'algoritme d'optical flow als dos vídeos usats en la resta de proves. L'optical flow s'ha aplicat sobre la zona de la superfície sobre la que es troba el ratolí i no tenint en compte els frames negres intercalats en els vídeos.

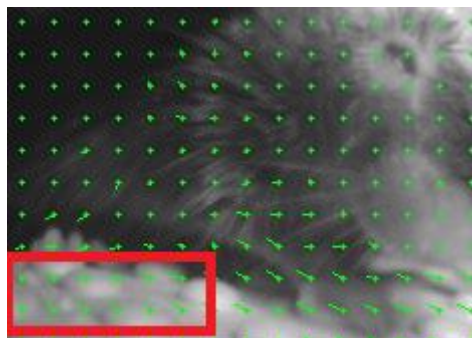


Figura 39. Optical Flow roda ratolí

A la segona gràfica el ratolí comença a caminar al frame 210 i va realitzant moviments intermitents fins al final del vídeo.

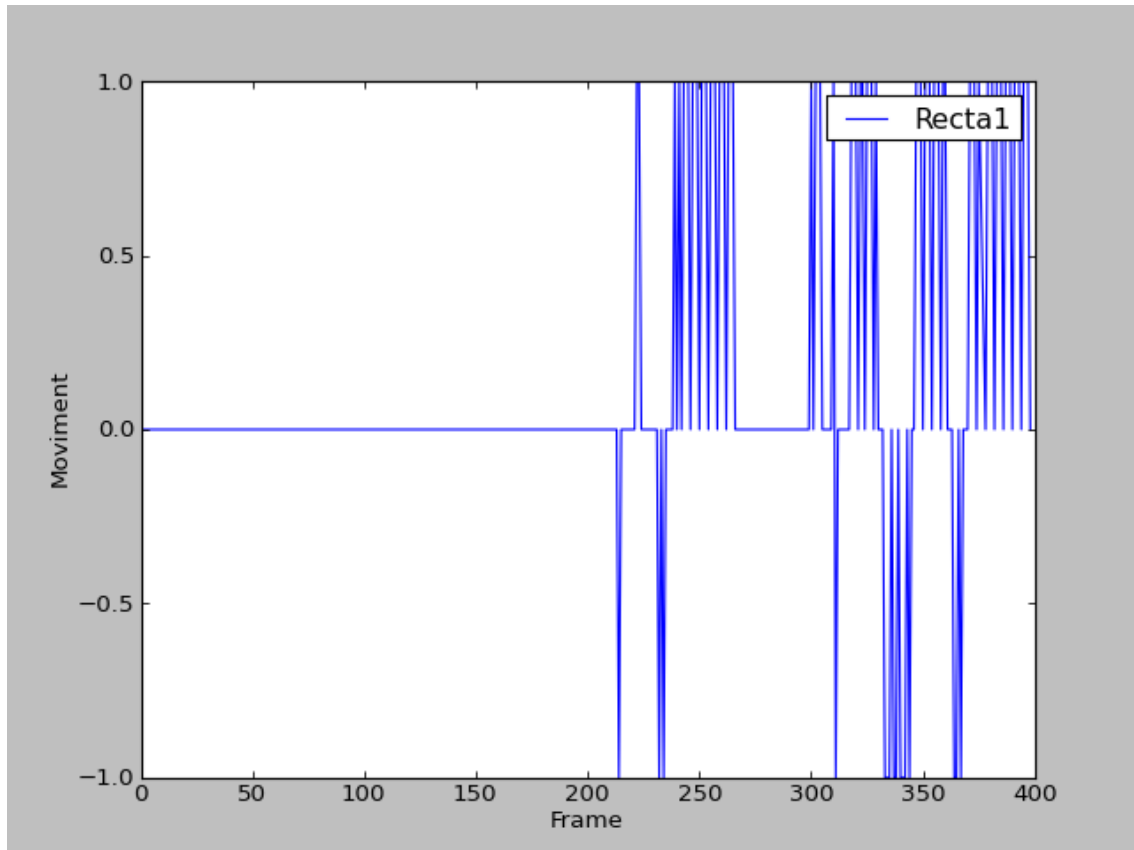


Figura 41. Gràfica Optical Flow roda ratolí 2

Els resultats d'aquest algoritme tot i ser més senzills, permeten obtenir una informació amb un nivell més alt de fiabilitat que la resta.

7.2 Resultats vibrisses

Per calcular els resultats de les vibrisses s'ha desenvolupat una aplicació que permet marcar manualment les rectes a cadascun dels frames d'un vídeo i els emmagatzema en una BBDD.

Posteriorment s'ha aplicat l'algoritme de Hough sobre els mateixos frames i amb una altra eina s'han comparat els resultats. L'eina de comparació de resultats comprova per cada una de les rectes marcades manualment, si es corresponen amb alguna de les que ha trobat l'algoritme automàtic. Aquesta eina retorna un percentatge del total de vibrisses trobades respecte les marcades, a més retorna un valor informant de la quantitat de vibrisses addicionals que ha trobat l'algoritme.

Les vibrisses addicionals detectades no tenen perquè ser incorrectes, ja que manualment només s'han marcat les que es veuen de manera clara i per això la precisió es valora sobre aquestes. En cas que hi hagi un grup de vibrisses si no es veu clarament quantes hi ha només se n'ha marcat una.

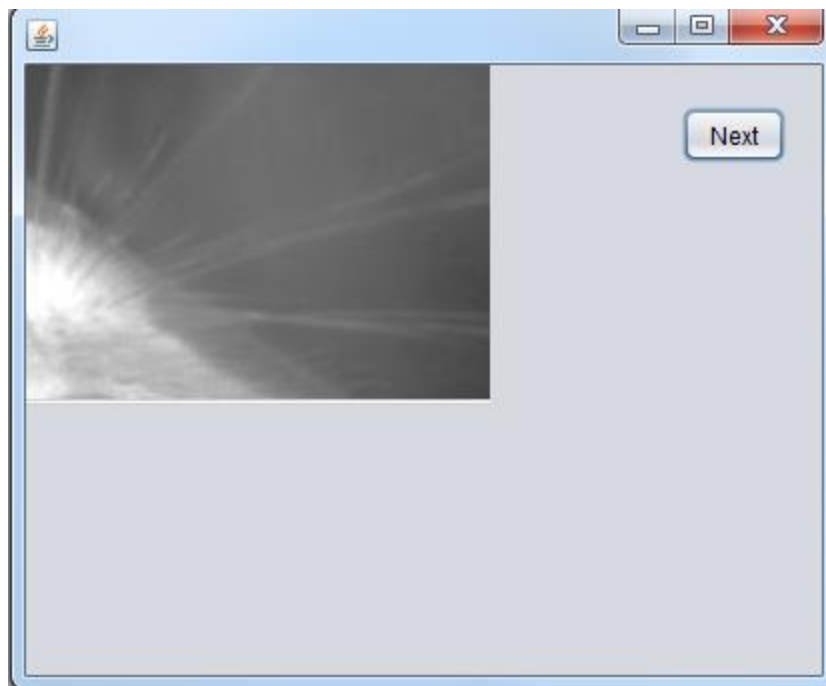


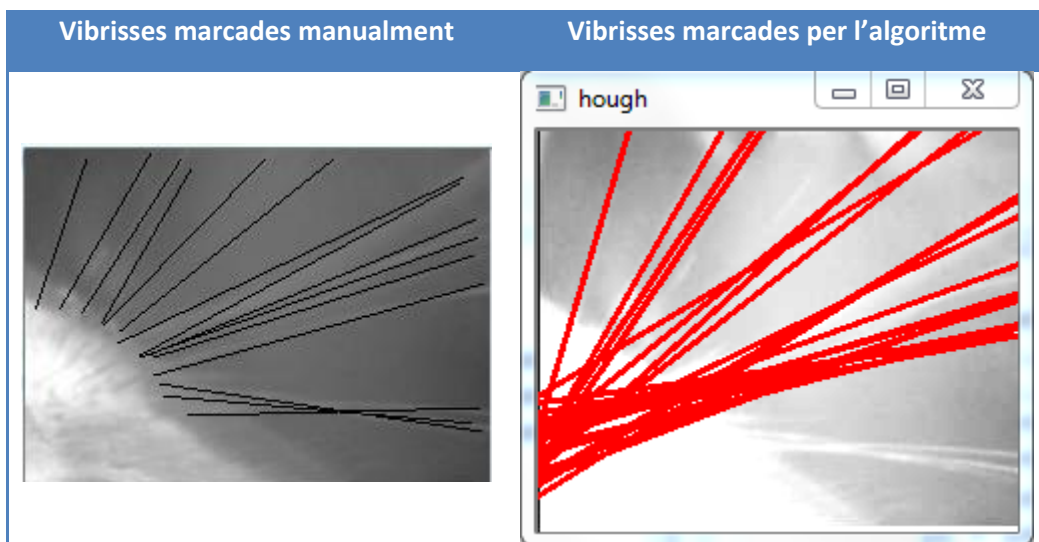
Figura 42. Eina marcatge manual vibrisses

A continuació es mostra l'encert de l'algoritme de Hough, amb Canny Edges i modificació de contrast en 5 frames diferents. S'ha usat un nombre reduït de frames enlloc de vídeos complets, perquè aquests resultats són més complexos que els de les potes i han de ser avaluats d'una manera més detallada.

Frame 1



Figura 43. Frame Resultats Vibrisses 1

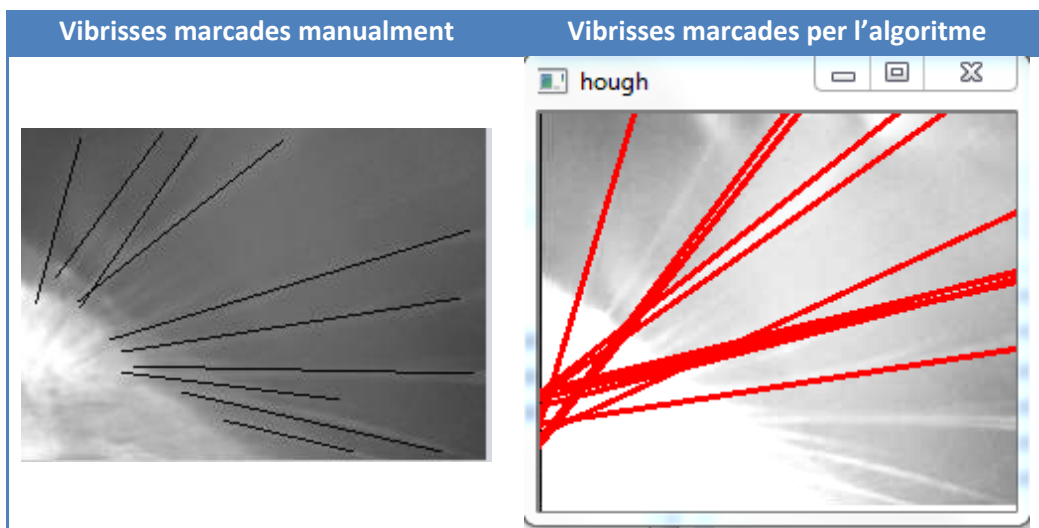


Percentatge d'encert	53%
Nombre d'encerts	8
Vibrisses detectades no marcades manualment	9
Comentari	En aquest frame no es detecten els bigotis de l'extrem dret i algun del centre poc visible respecte els marcats. Sobre els frames addicionals és difícil discernir si en sobra algun, ja que a les imatges no s'aprecia correctament quants hi ha en alguns grups.

Frame 2



Figura 44. Frame Resultats Vibrisses 2

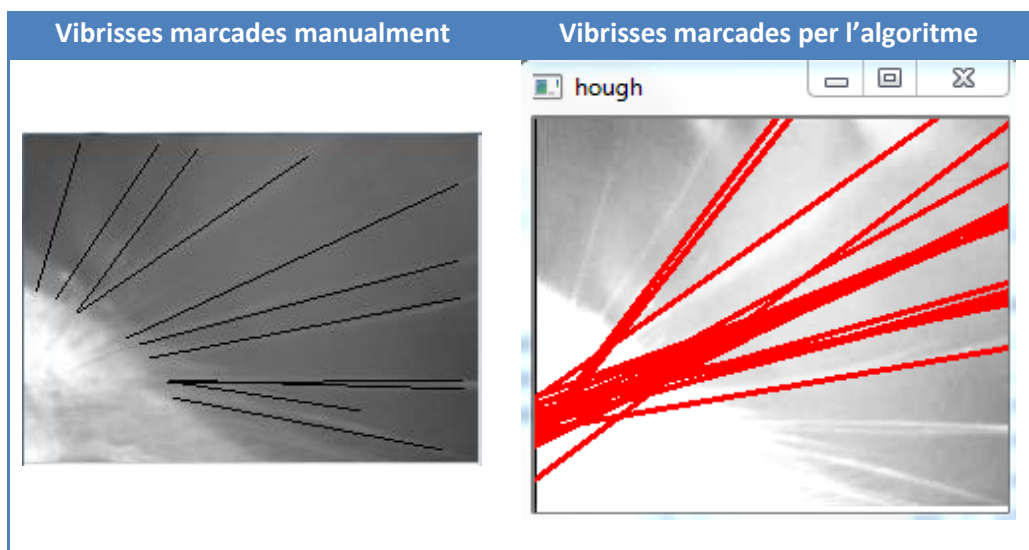


Percentatge d'encert	50%
Nombre d'encerts	5
Vibrisses detectades no marcades manualment	2
Comentari	En aquest frame s'han detectat menys vibrisses no marcades, això es deu a que a la imatge es veuen pocs grups de bigotis. D'altra banda també hi ha problemes amb la detecció dels bigotis dels extrems, ja que són més curts.

Frame 3



Figura 45. Frame Resultats Vibrisses 3

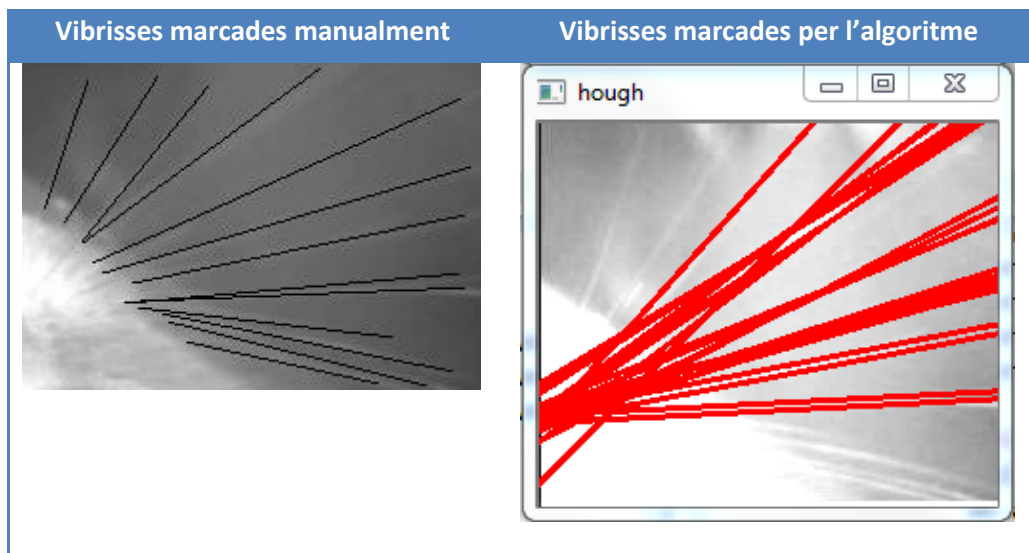


Percentatge d'encert	45%
Nombre d'encerts	5
Vibrisses detectades no marcades manualment	10
Comentari	En aquest frame a diferència dels altres sembla clar que s'han detectat algunes vibrisses més de les que realment hi ha. A més les vibrisses dels extrems no les ha trobat.

Frame 4



Figura 46. Frame Resultats Vibrisses 4

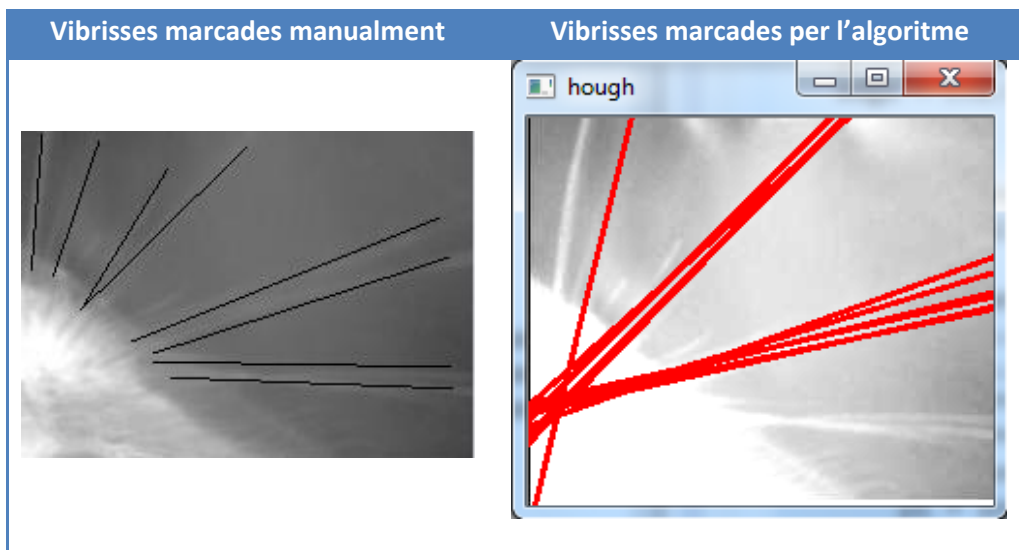


Percentatge d'encert	54%
Nombre d'encerts	7
Vibrisses detectades no marcades manualment	11
Comentari	En aquest frame dels mostrats és en el que hi ha més vibrisses detectades que no s'havien marcat. Però la majoria poden ser correctes ja que es visualitzen diversos grups de vibrisses, en les zones on ha detectat les vibrisses addicionals.

Frame 5



Figura 47. Frame Resultats Vibrisses 5



Percentatge d'encert	38%
Nombre d'encerts	3
Vibrisses detectades no marcades manualment	5
Comentari	En aquesta ultima imatge els resultats són similars a la resta de proves. Es detecta generalment els principals bigotis centrals però té dificultats per detectar les vibrisses dels extrems ja que són més curtes.

8. Conclusions del Treball

En aquest apartat s'avalua la consecució dels diferents objectius establerts a l'inici del treball. En termes generals es fa una valoració positiva dels resultats obtinguts.

- **Detecció estàtica potes:** Per la detecció estàtica de les potes els principals algorismes han estat Haar i SIFT, si bé cap dels dos ofereix uns nivells d'encert aplicables a una aplicació real. Es pot observar com ambdós tenen potencial per poder arribar a ser una bona solució si es corregeixen certs aspectes dels vídeos. Aquests aspectes són resolució, quantitat de vídeos (per poder augmentar la mida dels conjunts d'entrenament) i contrast de les potes amb el terra. A més és necessari un angle més lateral per poder detectar el colze i l'espatlla.
- **Detecció estàtica vibrisses:** Per la detecció estàtica de les vibrisses s'han descartat els algorismes usats a les potes ja que tenen molt poca superfície. Per això s'ha usat un detector de contorns sumat a un detector de rectes per tal d'obtenir la localització d'aquestes. Per millorar aquesta detecció es considera necessari augmentar la taxa de captura de la càmera per evitar frames borrosos. D'altra banda es considera necessari modificar el fons de la captura, per tal d'obtenir un major contrast entre les vibrisses i el fons.
- **Seguiment potes:** Per assolir el seguiment de les potes es pot usar els mateixos mecanismes que per la detecció estàtica, ja que les potes es troben clarament separades en l'espai. Aquesta separació espacial permet diferenciar de manera relativament senzilla la pota que s'està seguint.
- **Seguiment Vibrisses:** Amb les vibrisses la detecció estàtica no permet realitzar un seguiment encara que s'augmentés el rati d'encert d'aquest mètode. Això es deu a la gran quantitat de vibrisses i els creuaments d'aquestes, per això és necessari complementar aquest mètode amb altres que permetin tenir informació del moviment de cada vibrissa. Per exemple l'optical flow, per poder fer correctament el seguiment quan es produeixin creuaments.
- **Detectar possibles millores tècniques:** S'han detectat diverses millores necessàries ja exposades en els punts anteriors de les conclusions. Aquestes millores són de resolució, quantitat d'imatges per segon, angle de la càmera i contrast del ratolí amb el fons de la captura.
- **Codificació dels algorismes:** Tots els algorismes detallats en el projecte, s'han adaptat amb Python, depenent de les previsions d'èxit de l'algoritme han estat més modificats i millorats.
- **Proporcionar guia per futures investigacions i desenvolupaments:** Es considera que aquest objectiu s'ha assolit totalment, el treball ofereix una introducció als diferents algorismes a més de detallar tot el procés realitzat i els resultats obtinguts. A més s'han desenvolupat eines de validació de la precisió dels algorismes de detecció de potes i vibrisses.

9. Problemes trobats

En aquest apartat es presenten els problemes trobats al llarg del desenvolupament del treball. La major part dels problemes estaven previstos en el moment de planificació del projecte.

- **Domini nou:** Desconeixement del domini de la optogenetica. S'ha previst una fase d'estudi d'aquest domini a l'inici del TFM.
- **Vídeos en format Matlab:** Problemes alhora de tractar els vídeos degut a estar en format propietari Matlab. Es soluciona creant un script que permet convertir els vídeos a avi i un altre script que converteix un avi a imatges jpg.
- **Poc coneixement de Python:** Aprenentatge durant la fase d'implementació millorant el codi paral·lelament a l'adquisició del coneixement.
- **Poca experiència en visió per computador:** Pocs coneixements sobre algoritmes de detecció d'objectes a imatges. Per resoldre-ho es va planificar un temps per la recerca d'aquests.
- **Dificultat per calcular ratis d'encert dels algoritmes:** Per resoldre-ho s'han programat eines amb Java per poder validar els resultats.
- **Vídeos gravats amb un frame rate insuficient (120 FPS):** Aquest problema ha fet impossible el seguiment de les vibrisses al llarg dels vídeos.
- **Baixa resolució de les imatges:** Dificulta l'aplicació d'algoritmes com SIFT que es basen en cercar característiques del objecte. Per resoldre aquest problema s'han usat algoritmes alternatius.
- **Poc contrast entre els elements a detectar i l'entorn:** Ha estat necessari descartar determinats algoritmes, com la cerca de contorns per les potes i cercar alternatives com els filtres de Haar.

10. Possibles ampliacions

A continuació es presenten algunes possibles ampliacions, algunes subjectes a millores en la captura dels vídeos.

- **Generar una quantitat major de vídeos amb un angle lateral**, per millorar el conjunt d'imatges d'entrenament dels mètodes Haar i SIFT.
- En cas de millorar el "frame rate" dels vídeos de vibrisses **aplicar el Optical flow i la transformada de Hough per realitzar el seguiment de les Vibrisses**. Per aplicar el Optical flow també seria necessari millorar el contrast de les vibrisses amb el fons.
- **Detecció de colze i espatlla**, aquesta ampliació estaria subjecte a uns vídeos amb un angle més lateral. Per dur-ho a terme es podria usar com a base la posició trobada de la part inferior de la pota.
- Si es modifica el color del terra dels vídeos de les potes es pot usar una cerca basada en contorns.
- **Substituir SIFT per SURF** per obtenir un millor rendiment. SURF es un algoritme equivalent a SIFT, però amb certes millores que permeten un millor rendiment en la seva execució.
- **Usar en els algoritmes valors de training preprocessats**, enlloc de processar el conjunt en cada execució per reduir el seu temps d'execució.

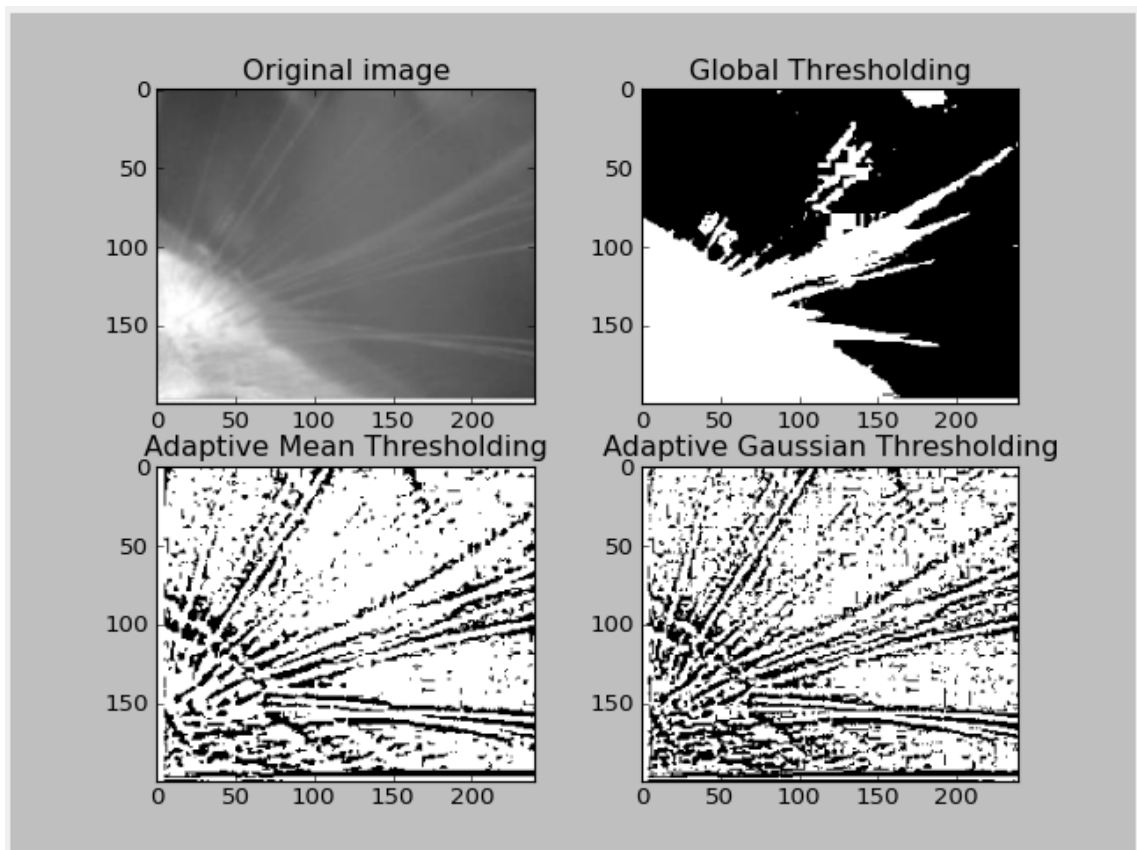
11. Bibliografia

- [1] mathworks.es/products/matlab/, MathWorks, Informació software Matlab. Febrer 2014.
- [2] python.org, Python, Lloc oficial llenguatge de programació Python. Febrer 2014.
- [3] dev.mysql.com, MySQL, Lloc oficial de la base de dades MySQL. Febrer 2014.
- [4] societyofrobots.com/programming_computer_vision_tutorial_pt4.shtml, Society of Robots, Documentació sobre algoritmes de visió per computador. Febrer 2014.
- [5] javaya.com.ar/detalleconcepto.php?codigo=130&inicio=40, JavaYa, Classe gràfics i els seus mètodes.
- [6] coding-robin.de/2013/07/22/train-your-own-opencv-haar-classifier.html, Coding Robin, Entrenar un classificador Haar amb OpenCV. Març 2014.
- [7] note.sonots.com/SciSoftware/haartraining.html, Naotoshi Seo, Entrenament de Haar amb OpenCV. Març 2014.
- [8] answers.opencv.org/question/7141/about-traincascade-parameters-samples-and-other/, OpenCV, Informació adicional classificador Haar. Març 2014.
- [9] opencvpython.blogspot.com.es, OpenCV Python, Guia d'ús d'OpenCv amb Python. Març 2014.
- [10] opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_tutorials.html, OpenCV-Python Tutorials, Guia d'ús d'OpenCv amb Python. Març 2014.
- [11] docs.opencv.org/index.html, OpenCv, Documentació Oficial OpenCv. Març 2014.
- [12] "Automated Tracking of Whiskers in Videos of Head Fixed Rodents", Nathan G. Clack, Daniel H. O'Connor, Daniel Huber, Leopoldo Petreanu, Andrew Hires, Simon Peron, Karel Svoboda, Eugene W. Myers. Març 2014
- [13] "Improved methods for chronic light-based motor mapping in mice", Gergely Silasi, Jamie D. Boyd, Jeff LeDue i Timothy H. Murphy. Març 2014
- [14] "Visual Object Tracking using Adaptive Correlation Filters", David S. Bolme, J. Ross Beveridge, Bruce A. Draper, Yui Man Lui. Abril 2014

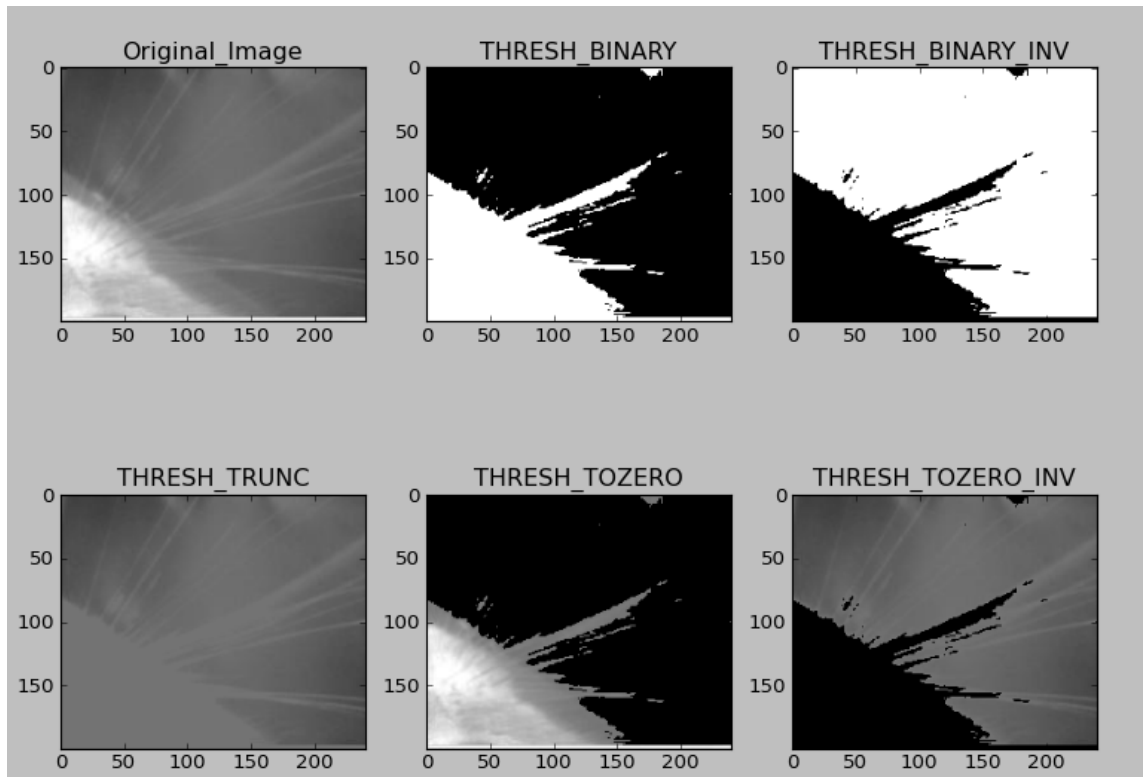
Annex 1: Classes Python

En aquest annex es detallen les classes Python adjuntes a la memòria del projecte i la funció que realitza cadascuna. Totes les classes s'entreguen preparades per executar sense cap modificació prèvia. D'altra banda cal disposar de Python 2.7.3 i Open CV 2.4.8 instal·lats i correctament configurats. A més cal disposar de les llibreries de Python Matplotlib i Numpy.

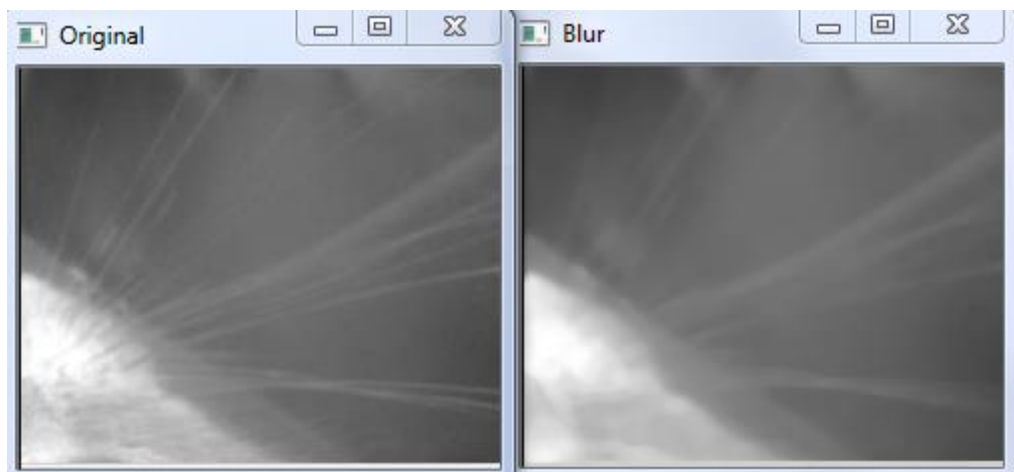
- **binarizationAdaptative.py**: Aplica els diferents tipus de binarització adaptativa disponibles a OpenCv sobre la imatge seleccionada. Tot seguit es presenta el resultat d'aquest.



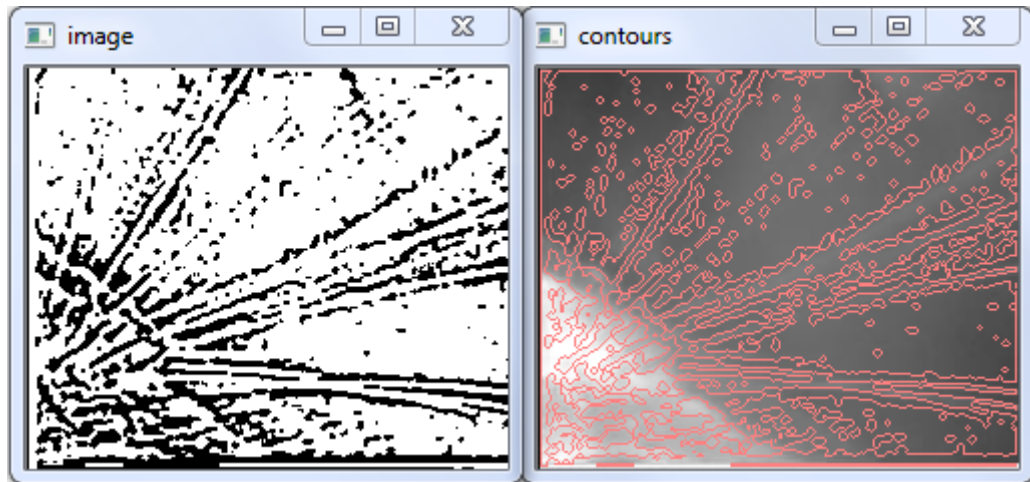
- **binarizationGlobal.py:** Aplica els diferent tipus de binarització global disponible a OpenCv sobre la imatge seleccionada. A continuació es presenta el resultat d'aquest.



- **blur.py:** En aquesta classe cal definir la imatge i el nivell de difuminat que es vol aplicar. El resultat és el que es mostra a continuació.



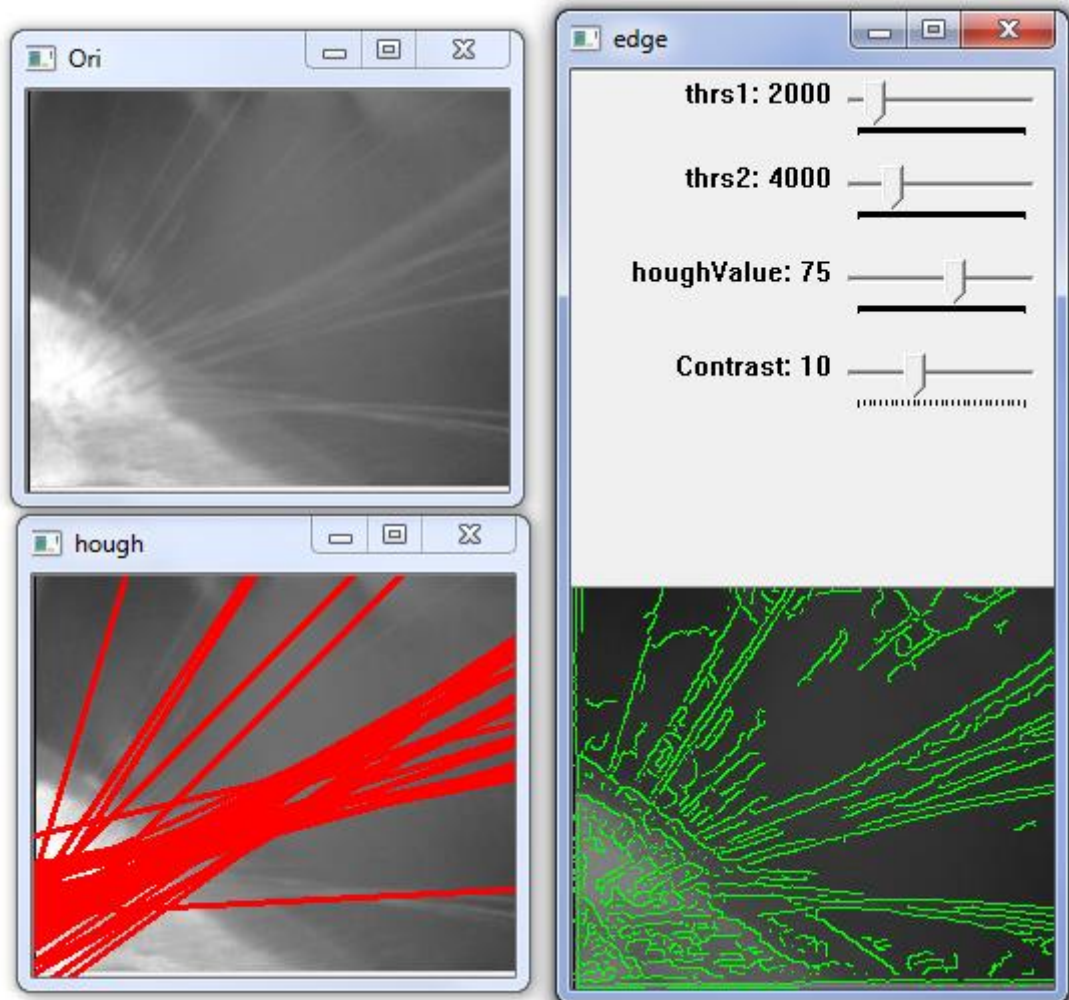
- **contornsGradient.py:** Classe que aplica el mètode de binarització adaptativa i posteriorment cerca de contorns d'OpenCv sobre la imatge seleccionada. El que s'obté al usar aquesta classe es correspon amb la imatge mostrada seguidament.



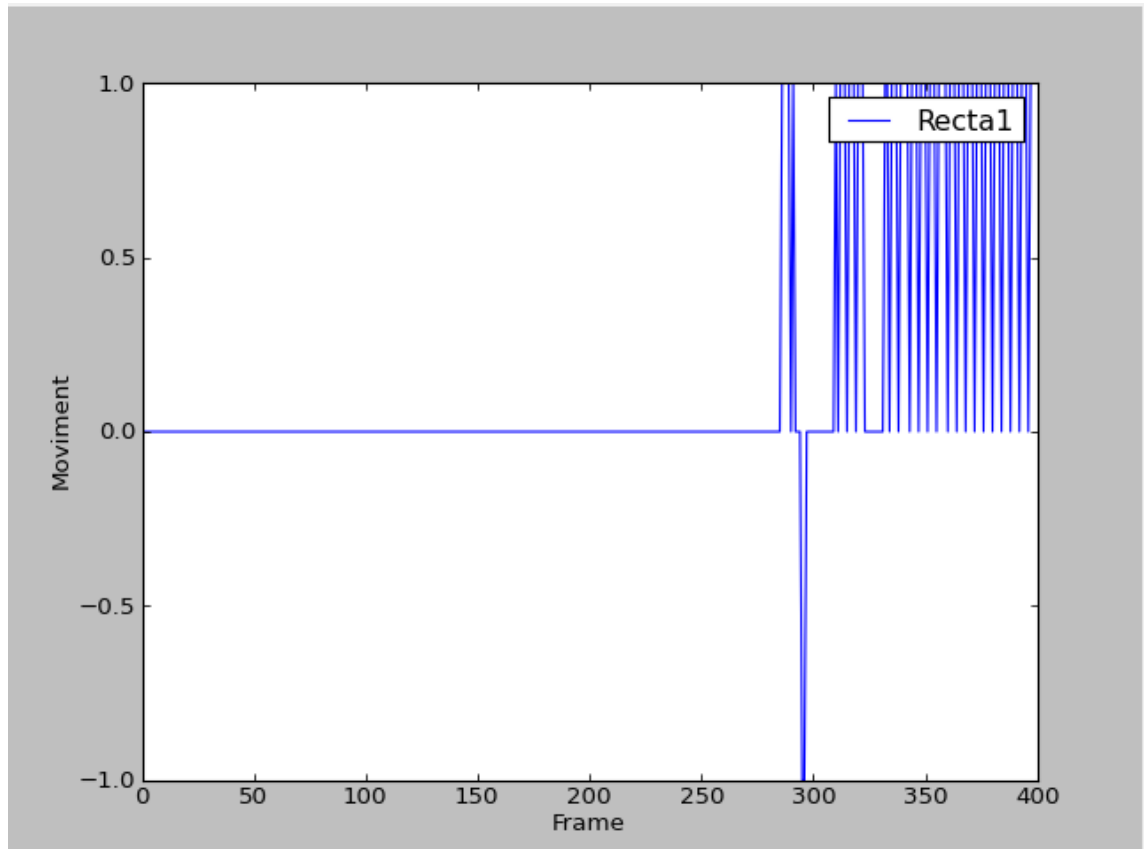
- **contrast.py:** Aplica el nivell seleccionat de contrast a la imatge triada. El resultat d'aquesta classe és el següent.



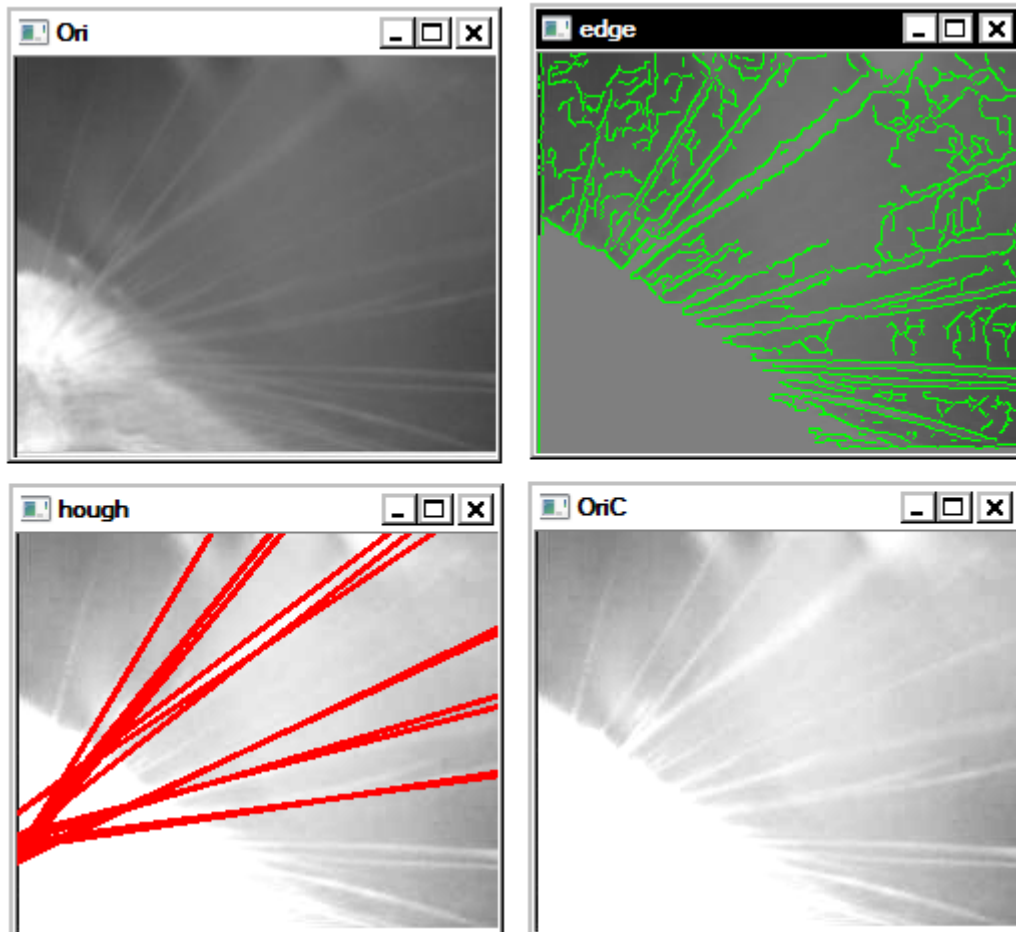
- **HoughLinesTest.py**: Interfície gràfica que permet provar diferents combinacions de paràmetres i veure els resultats en l'algoritme de Canny i Hough.



- **Opy_flow_rodaPota.py:** Aquest mètode calcula el Optical flow de la roda del vídeo seleccionat. El mètode proporciona el resultat amb una gràfica com la presentada tot seguit.



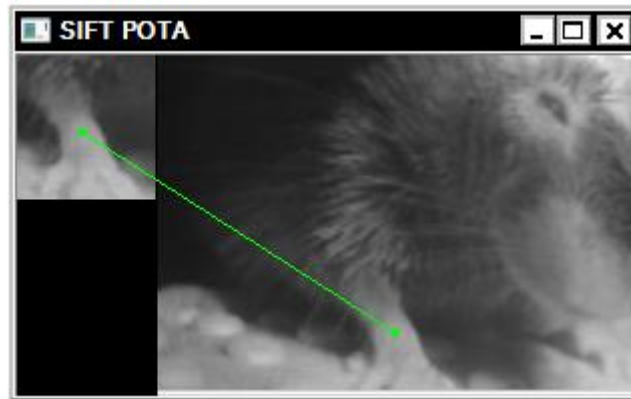
- **videoHoughLines.py:** Aquest codi aplica l'algorisme de HoughLines, sobre el vídeo de vibrisses seleccionat. La sortida és el vídeo marcant a cada frame les vibrisses trobades. També es mostra la imatge original, la imatge amb el contrast modificat i el resultat de la cerca de contorns de Canny.



- **Video of Haar.py:** Software que permet aplicar Haar sobre el vídeo seleccionat. El codi es pot configurar perquè emmagatzemi els resultats en una base de dades.



- **Video of SIFTFLANN.py:** Codi que aplica l'algoritme de SIFT sobre el vídeo de potes seleccionat. El codi es pot configurar perquè emmagatzemi els resultats en una base de dades.

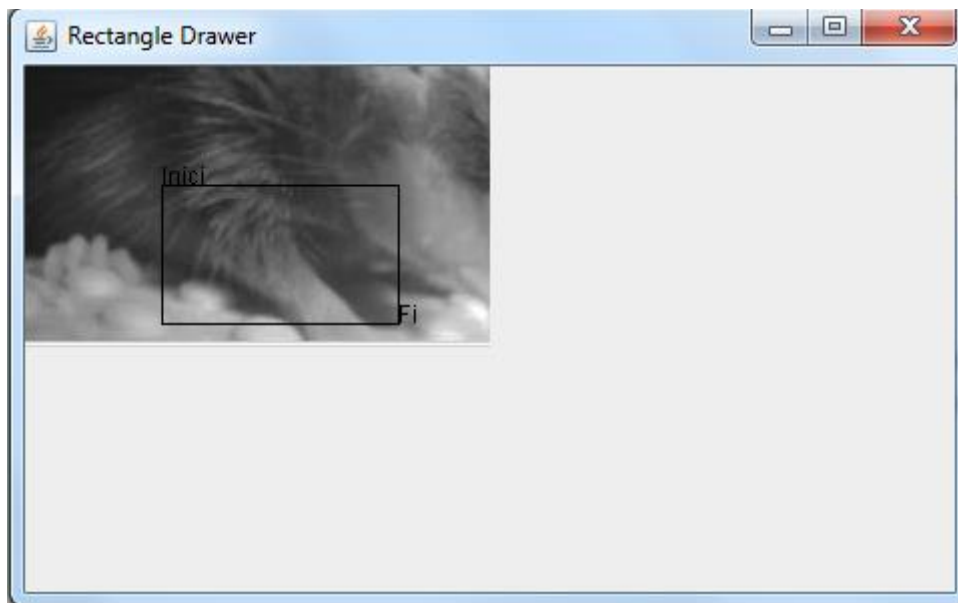


Annex 2: Eines de Test

En aquest segon annex es detallen les eines realitzades per validar els resultats dels algorismes, aquestes eines s'han lliurat conjuntament amb aquesta memòria. S'ha adjuntat el projecte Netbeans amb Java 1.7.

Validació de les potes

Per validar les potes s'han creat dues classes Java. La primera, `DibuixaRectanglesMouse.java`, és una eina gràfica que permet recorre els frames d'un vídeo i marcar manualment la posició de la pota.



Les coordenades marcades s'emmagatzemen en una taula anomenada "posició_potes_videos" d'una Base de dades MySQL amb l'estructura següent.

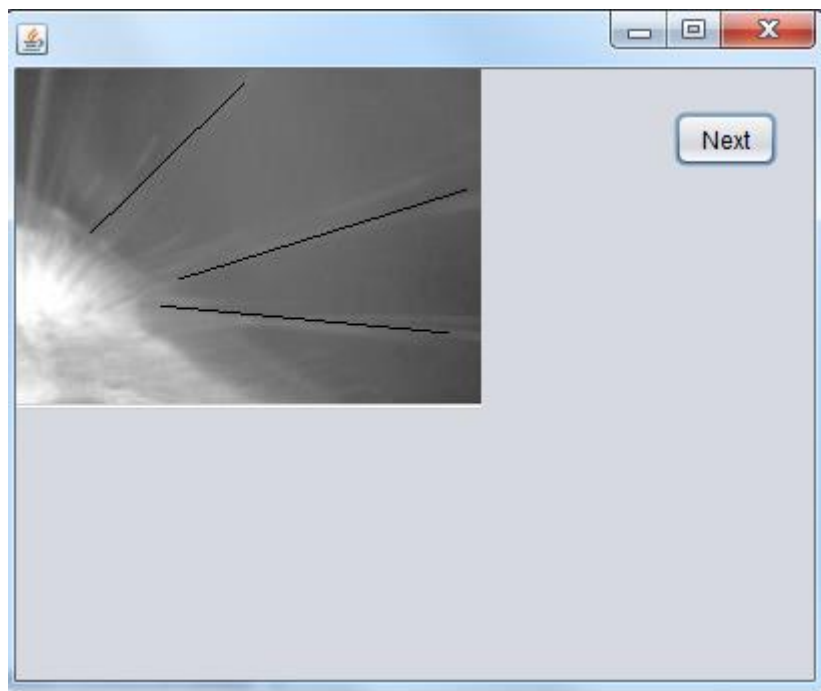
Column	Type	Default Value	Nullable
◇ Video	varchar(250)		NO
◇ FrameNumber	int(11)		NO
◇ X	int(11)		YES
◇ Y	int(11)		YES
◇ Height	int(11)		YES
◇ Width	int(11)		YES

La segona classe, `CheckEncertsPotes.java`, rep el nom del identificador dels resultats obtinguts manualment i dels resultats obtinguts amb l'algorisme usat. Mitjançant aquests dos valors consulta la base de dades, i realitza el càlcul del percentatge d'encert al llarg del vídeo.

Aquesta comparació es duu a terme comprovant si l'àrea seleccionada manualment i la de l'algoritme tenen un 50% o més d'intersecció.

Validació de les Vibrisses

Igual que en el cas de les potes s'han realitzat dues classes java. La primera, MarcaVibrisses.java, és una eina gràfica que permet recorre els frames d'un vídeo i marcar manualment la posició de les diferents vibrisses.



Les coordenades marcades s'emmagatzemen en una taula anomenada "posició_vibrisses_videos" d'una Base de dades MySQL amb l'estructura següent.

Column	Type	Default Value	Nullable
Video	varchar(250)		NO
FrameNumber	int(11)		NO
NumVibrissa	int(11)		NO
OriX	int(11)		YES
OriY	int(11)		YES
FinX	int(11)		YES
FinY	int(11)		YES

La segona classe rep el identificador del nom del identificador dels resultats obtinguts manualment i dels resultats obtinguts amb l'algoritme usat. Mitjançant aquests dos valors realitza el càlcul del percentatge d'encert a un únic frame. Aquesta comparació es duu a terme comprovant si el origen i el destí de la recta seleccionada manualment, estan entre els dos a menys de 12 píxels d'algun punt de la recta trobada per l'algoritme.