ÍNDEX

1.- Introducció i objectius 3

1.1.- Els models DTI (Diffusion Tensor Imaging) 4
1.2.- La Plataforma DTIWeb 6
1.3.- Objectius 6
1.4.- Pla de treball 7
1.5.- Metodologia a seguir 8

2.- Estudi teòric 11

2.1.- Obtenció de les dades de DTI 12
2.2.- Tècniques de visualització de dades DTI 13
2.3.- Clustering de fibres 18
2.4.- Descripció general d'algorisme de clustering. 19
2.5.- Classificació dels algorismes de clustering 21
2.5.1.- Algorismes de clustering exclusius 21
2.5.1.1.- L'algorisme k-means 21
2.5.2.- Algorismes de clustering de solapament 23
2.5.2.1.- L'algorisme Fuzzy c-means 23
2.5.3.- Algorismes de clustering jeràrquic 25
2.5.4.- Algorismes de clustering probabilístic 27
2.6.- Mètodes de distàncies entre fibres 28
2.6.1.- Distància dels punts més propers 28
2.6.2.- Mitjana de les distàncies de punts més propers 29
2.6.3.- Distància de Hausdorff 30
2.6.4.- Distància dels punts finals 31
2.6.5.- Distància sobre el llindar 32
2.7.- Visualització de clústers 32

3.- Estudi de la plataforma DTIWEB 35

3.1. Estructura de la plataforma DTIWeb 36
3.1.1.- Capa de dades 37
3.1.2.- Capa d'aplicació 37
3.1.3.- Capa de presentació 37
3.2. Llenguatge de programació 38

4.- Implementació 39

4.1.- Anàlisis i requeriments generals 40
4.2.- Implementació dels mètodes de distància 49
4.2.1.- Model utilitzat per realitzar les proves 49

4.2.2.- Distància dels punts més propers 50
4.2.3.- Mitjana de distàncies dels punts més propers 51
4.2.4.- Distància de Hausdorff 53
4.2.5.- Distància dels punts finals 55
4.2.6.- Distància sobre el llindar 56
4.3.- Implementació dels mètodes de clustering 58
4.3.1.- Models utilitzats per a realitzar les proves 58
4.3.1.1.- Models sintètics 58
4.3.1.2.- Models reals 63
4.3.2.- L'algorisme de clustering jeràrquic 65
4.3.2.1.- Implementació sobre models sintètics 65
4.3.2.2.- Resultats sobre models reals 77
4.3.3.- L'algorisme K-means 78
4.3.3.1.- Implementació sobre models sintètics 78
4.3.3.2.- Resultats sobre models reals 83
4.3.4.- L'algorisme Fuzzy C-Means 84
4.3.4.1.- Implementació sobre models sintètics 84
4.3.4.2.- Resultats sobre models reals 92
4.4.- Optimització del clustering a partir de la simplificació de les fibres 94
4.4.1.- Explicació de l'optimització del clustering 94
4.4.2.- Resultats sobre models reals 98
4.5.- Resultats dels temps d'execució 100
4.5.1.- Resultats sobre models sintètics 100
4.5.2.- Conclusions (models sintètics) 106
4.5.3.- Resultats sobre models reals 106
4.5.4.- Conclusions (models reals) 107
4.6.- Implementació de la visualització de clústers 108
4.6.1.- Visualització de clústers fibra a fibra 108
4.6.2.- Visualització de clústers per bandes 112
4.6.2.1.- Implementació simple 112
4.6.2.2.- Implementació sobre models sintètics 114

5.- Conclusions 117

5.1.- Conclusions generals 118
5.2.- Treball futur 120
5.3.- Conclusions personals 120

6.- Bibliografia 120

1.- INTRODUCCIÓ I OBJECTIUS

Actualment, el diagnòstic mitjançant la imatge mèdica, s'ha convertit en una eina fonamental en la pràctica clínica diària, ja que permet, entre d'altres coses, reconstruir a partir d'un conjunt d'imatges 2D, obtingudes a partir d'aparells de captació, qualsevol part de l'organisme d'un pacient i representar-lo en un model 3D. Sobre aquest model 3D poden realitzar-se diferents operacions que faciliten el diagnòstic i la presa de decisions als especialistes.

La captació de dades del pacient es realitza mitjançant algun dispositiu físic que permet mesurar un o més paràmetres determinats en diferents punts del cos del pacient. Entre aquests dispositius de captació trobem la tomografia computeritzada, la ressonància magnètica, l'ecògraf, etc.

En aquest projecte ens centrarem en el processament i la visualització d'una nova modalitat de ressonància magnètica anomenada tensor de difusió (DTI). A diferència dels altres dispositius de captació, la informació que obté el DTI és molt més precisa i, per tant, més difícil de processar.

1.1.- Els models DTI (Diffusion Tensor Imaging)

Els DTI són una nova modalitat de ressonància magnètica (MRI) a partir de la qual es pot obtenir la informació necessària per a determinar com circula la matèria blanca a les fibres del cervell. A través de l'aparell de ressonància es captura informació del pacient en diferents direccions. Aquesta informació es representa en forma d'imatges que es codifiquen en un fitxer seguint el protocol fixat pel dispositiu de captació. A partir d'aquesta informació s'apliquen diferents tècniques de processament i visualització d'informació que permeten generar els mapes de connectivitat. Actualment, tot i que s'han proposat diferents tècniques, encara no existeix un mètode estàndard que pugui aplicar-se en la pràctica clínica. El principal problema en el que es troben els experts és que no tenen forma de validar si les imatges que obtenen són correctes o no i, per tant, no poden validar si el mètode usat és el més apropiat o no.
La informació del DTI es representa mitjançant un tensor de difusió, que representa la difusió de les molècules d'aigua en una unitat mínima de volum anomenada vòxel. Tot i no existir un mètode d'interpretació i de visualització de DTI estàndard, la majoria de tècniques proposades segueixen els mateixos passos. A partir de la representació d'aquest tensor, s'obté informació de com són les fibres a l'interior del vòxel.
Aplicant aquest procés vòxel a vòxel sobre tot el volum, podem saber com estan connectades totes les fibres. Finalment el que obtenim és com la matèria blanca connecta les zones de matèria gris.

A continuació podem veure algunes de les utilitats clíniques de DTI:

- Localització específica d'extensió de lesions de matèria blanca, com el trauma.
- Localització de tumors en relació amb les extensions de matèria blanca (la infiltració, la desviació).
- Localització de les principals extensions de matèria blanca per a la planificació de neurocirurgia.
- Avaluació de matèria blanca en el desenvolupament, la patologia i la degeneració.

El DTI també té usos en la caracterització de múscul esquelètic i cardíac.

La informació que s'obté del DTI es representa en un model de vòxels. La difusió a nivell de vòxel es representa a partir d'un tensor (matriu simètrica de 3x3).

Un dels principals problemes que es plantegen relacionats amb el processament de dades de DTI és com representar la informació dels tensors en la pantalla.

Per resoldre aquest problema s'han proposat diferents tipus de tècniques. Aquestes tècniques van lligades amb la informació que es pot obtenir del tensor. Això ens porta a parlar dels vectors i valors propis i diferents coeficients que es poden obtenir a partir d'aquests valors. D'entre les diferents tècniques que s'han proposat, nosaltres ens centrarem en els mètodes de *tracking*. Aquests mètodes simulen el traçat de les fibres del cervell partint d'un punt inicial i reconstruint el camí que segueix la fibra al llarg del cervell. La base d'aquestes tècniques és la simplificació del tensor en un vector que segueixi la direcció principal de difusió. Una vegada s'ha reconstruït aquest camí s'aplica un algorisme de visualització per poder-lo veure en pantalla. Les imatges que s'obtenen són com les que es mostren en les figures 1 i 2.

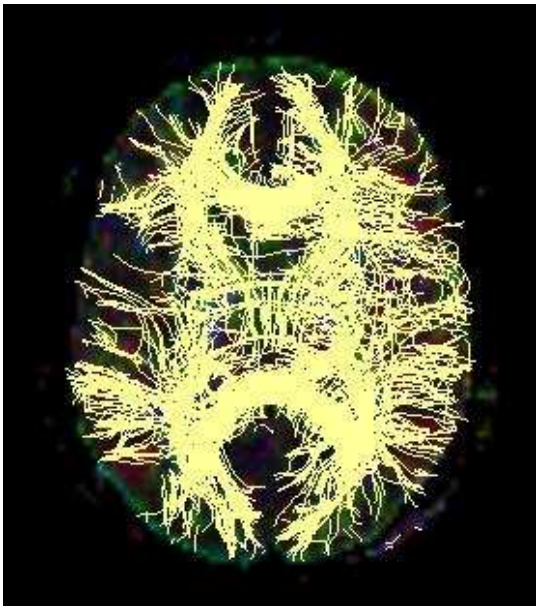


***FIGURA 1**. Imatge on es pot apreciar la gran quantitat de fibres que hi ha al cervell.*

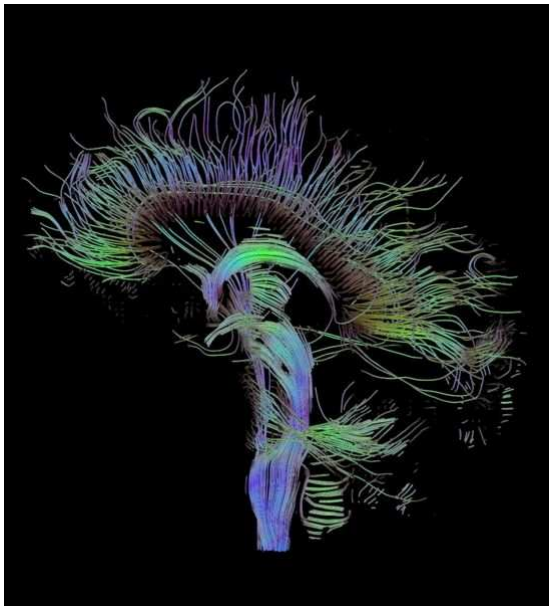


***FIGURA 2**. Reconstrucció tractogràfica de les connexions neuronals via DTI.*

La interpretació d'aquestes imatges no és fàcil donat l'elevat nombre de fibres que hi ha en el cervell. Per resoldre aquest problema es proposen tècniques d'agrupament de fibres o tècniques de clustering. El que fan aquests algorismes és agrupar les fibres en funció de les zones que connecten.

1.2.- La Plataforma DTIWeb

La plataforma DTIWeb és una aplicació Java robusta, portable i extensible per a la visualització i el processament de les dades DTI. El DTIWeb integra estratègies de visualització, diferents tècniques de traçat de fibres i tècniques d'interrogació que doten d'una informació molt valuosa per al diagnòstic.

L'estructura proposada es basa en la plataforma de programació de Java 3D que proporciona un model de programació orientat a objectes i independència tant de la configuració del hardware de la màquina com del sistema operatiu.

El DTIWeb ha estat dissenyat per treballar en l'entorn d'Internet i només requereix un navegador.

Les principals característiques de l'entorn proposat són la portabilitat, el fet que sigui reutilitzable i extensible i la flexibilitat.

Aquesta plataforma es desenvolupa conjuntament entre el Laboratori de Gràfics i Imatge de la Universitat de Girona i l'Hospital Dr.Josep Trueta de Girona.

1.3.- Objectius

L'objectiu principal d'aquest projecte es integrar a la plataforma DTIWeb diferents algorismes de clustering i tècniques de visualització de clústers de fibres de forma que faciliti la interpretació de dades de DTI als especialistes.

Per assolir aquest objectiu ens caldrà:

- Estudiar, dissenyar i implementar diferents **tècniques d'agrupament de fibres (clustering).**

Els mètodes d'agrupament de fibres agrupen les fibres en diferents clústers o grups. Per a això es necessita d'algun criteri per a decidir si la fibra que s'està tractant pertany a un grup de fibres o a un altre. Per a prendre aquesta decisió s'utilitza el concepte de distància entre fibres. Hi ha diferents mètodes de distàncies entre fibres.

En aquest projecte s'implementaran diferents mètodes de distància així com diferents algorismes de clustering, que es podran combinar, i així podrem fer una avaluació de quins obtenen un millor resultat o en quines situacions caldrà escollir uns o altres.

- **Estudiar, dissenyar i implementar una tècnica de visualització de clústers de fibres.**

Fins ara disposàvem d'una visualització de totes les fibres. El que es pretén en aquest projecte és fer que la visualització dels clústers sigui de tota una banda enlloc de veure cadascuna de les fibres i que doni una sensació de profunditat. Obtenir aquesta sensació d'estar visualitzant objectes en tres dimensions és un dels principals problemes.

- **Integració dels diferents mòduls implementats en la plataforma DTIWeb**

Tant la part d'implementació dels mètodes d'agrupament de fibres com la part de visualització ha de quedar integrada a la plataforma DTIWeb.

Per assolir aquest objectiu seguirem el pla de treball que es presenta en la següent secció.

1.4.- Pla de treball

El pla de treball a seguir durant el curs del projecte consta de dues fases, una d'estudi previ i una altra d'implementació.

Fase d'estudi previ

Adquisició dels coneixements previs necessaris per poder assolir els objectius del projecte. Aquest estudi inclou una part teòrica, relacionada amb el DTI i els mètodes de clustering i una part pràctica relacionada amb la plataforma DTIWeb

Una vegada finalitzada aquesta fase podem començar l'etapa d'implementació.

Fase d'implementació

Aquesta és la fase on es farà l'anàlisi de requeriments, el disseny i la implementació de l'aplicació.
Avaluarem els requeriments funcionals del projecte i farem servir el llenguatge de modelat unificat (UML) per a representar-los.
Tindrem, doncs, un conjunt de diagrames de classes, de seqüència, fitxes de cas d'ús, etc.
En aquesta fase, un cop analitzats els requeriments, és on es farà la implementació dels algorismes de clustering, les distàncies entre fibres i la part que fa referència a la visualització.
Aquesta fase acabarà amb un testeig sobre models de dades reals.

1.5.- Metodologia a seguir

La metodologia que utilitzarem a l'hora de fer la implementació estarà basada en l'Extreme Programming.

L'Extreme Programming (també anomenat XP) és una metodologia d'enginyeria del software que condueix a un procés de desenvolupament que es més receptiu a les necessitats del client que no pas els mètodes tradicionals i permet crear un software de major qualitat ja que permet canvis en els requeriments per part de l'usuari, cosa que els defensors d'aquesta metodologia consideren que és un aspecte normal, inescapable i desitjable dels projectes de desenvolupament de software.
És més realista pensar en l'adaptació al canvi de requeriments en qualsevol punt de la durada del projecte i una millor aproximació que no pas intentar definir tots els requeriments a l'inici del projecte i després haver de perdre molt de temps en fer adaptacions als requeriments inicials en cas de que hi hagi canvi de plans.

Característiques fonamentals de l'Extreme Programming:

- Desenvolupament iteratiu i incremental. Es fan petites millores, una rere l'altra.
- Proves unitàries contínues. Freqüentment repetides i automàtiques, incloent proves de regressió. S'aconsella escriure el codi de la prova abans de la codificació.
- Programació per parelles. Es recomana que les tasques de desenvolupament es duguin a terme entre dues persones en un mateix lloc. El codi és revisat i

discutit mentre s'escriu aconseguint una major qualitat del codi escrit que se suposa és més important que la possible pèrdua de productivitat immediata. A més, a diferència de les metodologies tradicionals en que la feina és repartida per mòduls, en aquesta metodologia es promou que qualsevol membre de l'equip pugui corregir i estendre qualsevol part del projecte amb la qual cosa es garanteix poder detectar més errors.

- Freqüent interacció de l'equip de programació amb el client o l'usuari. Es recomana que un representant del client treballi juntament amb l'equip de desenvolupament.
- Correcció de tots els errors abans d'afegir una nova funcionalitat, així com fer entregues freqüents.
- Refactoreig del codi, és a dir, reescriure certes parts del codi per tal d'augmentar la seva llegibilitat i mantenibilitat però sense modificar el seu comportament. Les proves han de garantir que en el refactoreig no s'ha introduït cap errada.
- Simplicitat en el codi ja que aquesta és la millor manera per a que les coses funcionin.

En la implementació d'aquest projecte s'intentarà seguir totes les premisses esmentades, no essent possible el compliment d'alguna d'elles, com seria, la programació per parelles, per un motiu obvi, ja que el projecte s'ha fet de forma individual. També, val a dir, que la figura del client serà substituïda per la tutora del projecte, l'Imma Boada.

El joc de proves que s'utilitzarà per a fer els repetits testejos seran els models de DTI adquirits per l'Hospital en diferents voluntaris i també models sintètics.

Resumint, l'Extreme Programming ens permetrà desenvolupar un software que sigui fàcil d'adaptar en cas de canvis o ampliacions.

2.- ESTUDI TEÒRIC

En aquest apartat es fa un recull de la informació necessària per a entendre el context i per a assolir els objectius que ens hem proposat.

Primer veurem com s'obtenen les dades DTI i quines tècniques de visualització s'hi poden aplicar.

Els següents apartats ja fan referència a la informació que ens servirà per a la implementació: algorismes de clustering de fibres, mètodes de distància, etc.

2.1.- Obtenció de les dades de DTI

El primer pas que cal fer per a fer possible la visualització és obtenir el conjunt de dades que caldrà interpretar. Aquestes dades s'obtenen mitjançant simulacions o bé a través de dispositius especialitzats de captació que mesuren un o més paràmetres determinats (la intensitat de la imatge dels quals en determina el valor) en diversos punts del cos del pacient, guarden les imatges captades en format DICOM i treballen amb un tipus de dades que es distribueixen regularment per l'espai i sobre plans.
El procés de captació simula el seccionament del cos en talls plans.

És important tenir en compte que les resolucions i l'extensió espacial dels conjunts de dades poden ser diferents depenen del dispositiu que s'hagi fet servir.

Hem parlat de la captació de dades en general, però ara ens centrarem en el cas de les dades DTI, les quals mesuren la difusió molecular, que és el moviment translacional de les molècules per la seva energia tèrmica. Aquest moviment varia en funció de les característiques de l'ambient en el que es troben les molècules. En el cas concret del cervell, el moviment està dirigit per les zones blanques. A partir d'aquesta informació podem saber l'estructura que té el cervell.

Darrera d'aquest procés de captació hi ha una base teòrica que permet el càlcul del tensor de difusió a partir d'un conjunt de fórmules, però, que no explicarem, ja que, queden fora de l'abast d'aquest projecte.

Un cop tenim les dades del pacient, per a poder-les tractar cal definir un esquema de representació que ens permeti accedir-hi d'una manera senzilla. El model més utilitzat en el camp de la medicina és el model de vòxels que el que fa és subdividir l'espai en un conjunt de cubs o paral·lelepípedes de les mateixes dimensions seguint una malla regular, cadascun dels quals rep el nom de vòxel.
Les dades que tenim guardades en el fitxer DICOM segueixen una distribució regular sobre un pla podent-se interpretar com un tall de la part del cos que s'està examinant. Els diferents plans es posaran un sobre l'altre intentant reconstruir de nou la part 3D del cos. D'aquesta manera obtenim una malla regular 3D formada per un conjunt de cel·les. Als vèrtexs d'aquestes cel·les hi haurà la informació de cada un dels plans.
Els equipaments de captació generen imatges monocromàtiques de talls normalment paral·lels i uniformement espaiats donada una certa regió 3D. En el cas de la medicina, aquests talls són, generalment, longitudinals, cobrint una certa regió del cos del pacient que conté un òrgan o fenomen que es vol estudiar.

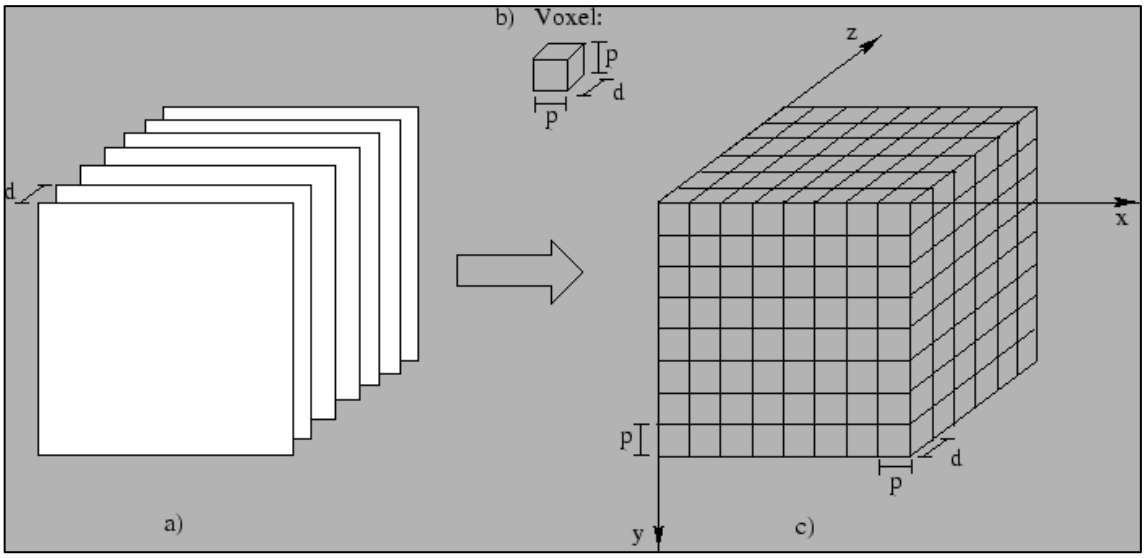


FIGURA 3. *Captació de la imatge i transformació de la informació*

A la figura 3 podem veure'n un exemple on la imatge *a* representa els talls obtinguts per l'aparell de captació amb una distància *d* de separació entre ells.
Llavors, considerant les dimensions *p x p* d'un píxel i el desplaçament entre els talls, la extensió del píxel en 3D forma un petit paral·lelepípede de dimensions *p x p x d*, que és l'anomenat vòxel (la paraula vòxel prové de l'anglès volum element). El centre dels vòxels coincideix amb el centre dels píxels. Els vòxels representen punts de mostreig d'algun fenomen físic i són utilitzats per a reconstruir a l'ordinador la forma o funció d'estructures tridimensionals. En aquest cas, la imatge pot ser representada com una matriu 3D.
De cadascun dels vòxels se'n guardarà el tensor de difusió.

2.2.- Tècniques de visualització de dades DTI

Per a obtenir la imatge final s'ha d'aplicar alguna tècnica de visualització sobre el model de dades. En la tècnica de visualització es determina quins atributs gràfics s'assignen als diferents valors de la propietat representada en el model, el quals varien en funció del tipus d'informació que vulguem representar i també dependran de la informació que tinguem representada en el model.

En el cas de les dades DTI les tècniques de visualització van lligades als valors que ens és possible calcular a partir del tensor del que disposem de cada vòxel.
Podem classificar les tècniques de visualització de la següent manera:

- **Mètodes 1D**. Basats en calcular un valor escalar reduint la informació del tensor a un valor. Com s'ha explicat en l'apartat anterior per a cada vòxel disposem d'un tensor de difusió que pot ser representat com una matriu de 3x3. Gràcies a això, podem determinar les direccions principals de difusió de cada vòxel, a través de la diagonalització d'aquesta matriu. El que ens proporciona la diagonalització són els valors i vectors propis que ens donen com a resultat les tres direccions principals de difusió. Aquestes tres direccions es poden representar sobre una el·lipsoide. Finalment, a partir d'una fórmula que escollim, els reduïm a un únic valor que ens servirà per a representar mapes de color. A la *figura 4* es veu un exemple d'aquest procés.

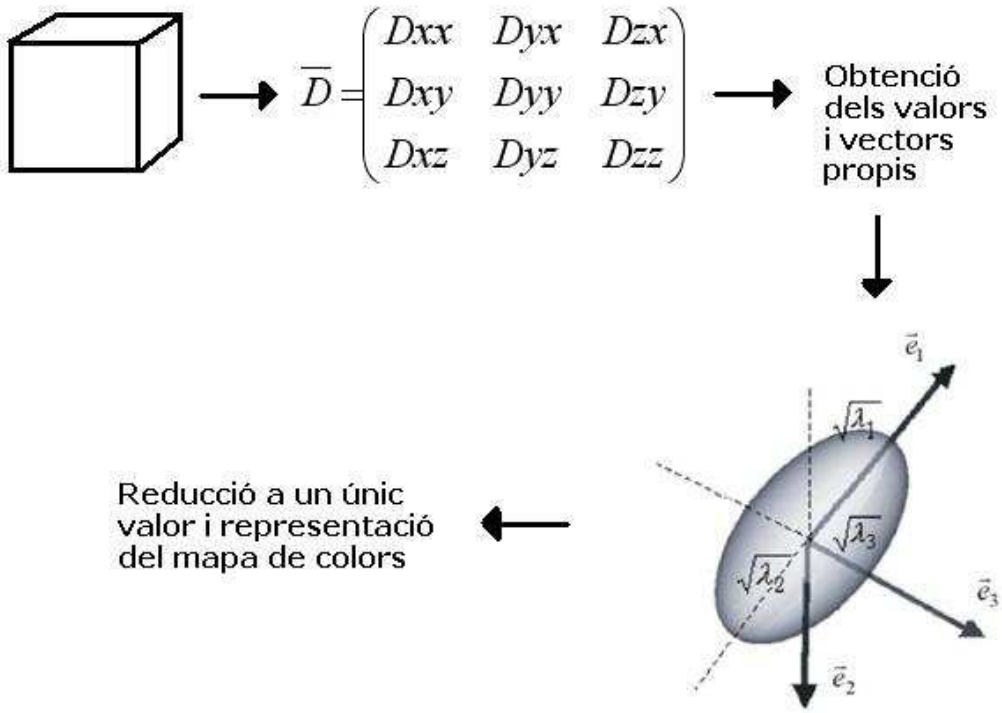


FIGURA 4. *Funcionament de la tècnica de visualització 1D.*

La visualització d'aquests valors es fa en escala de grisos tot i que, per a obtenir més informació, es poden reemplaçar els nivells de grisos per valors RGB fent ús de diferents tècniques. Els valors que es poden utilitzar per a obtenir aquests tipus de visualització són diversos i es calculen mitjançant equacions que s'apliquen sobre els valors de la matriu. La *figura 5* ens mostra un exemple d'aquesta visualització sobre un model de cervell humà utilitzant diferents paràmetres.

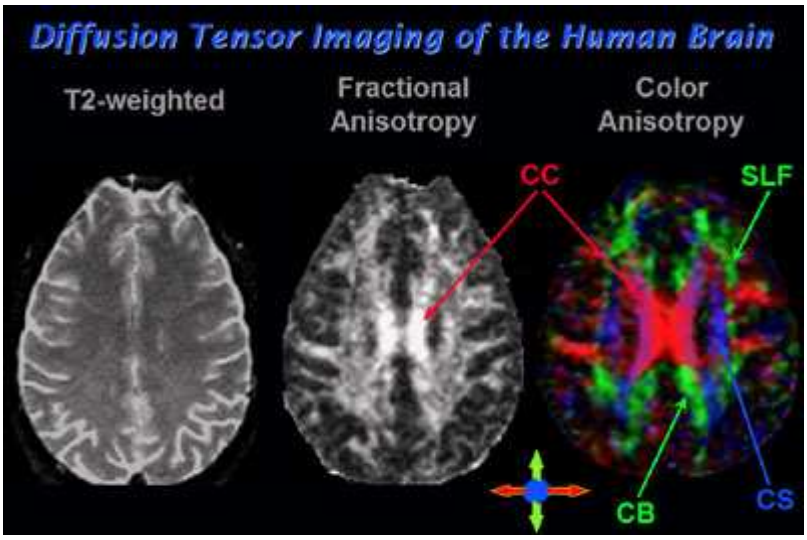


FIGURA 5. *Exemple de visualització 1D*

- **Mètodes 2D**. Intenten reduir la informació que proporciona el tensor de difusió en una icona 2D que s'anomena glyph.
  S'intenta representar la pèrdua d'informació que es deriva d'utilitzar una tècnica de visualització 1D. Per a aquest tipus de visualització s'utilitzen, normalment, cuboides i el·lipsoides.

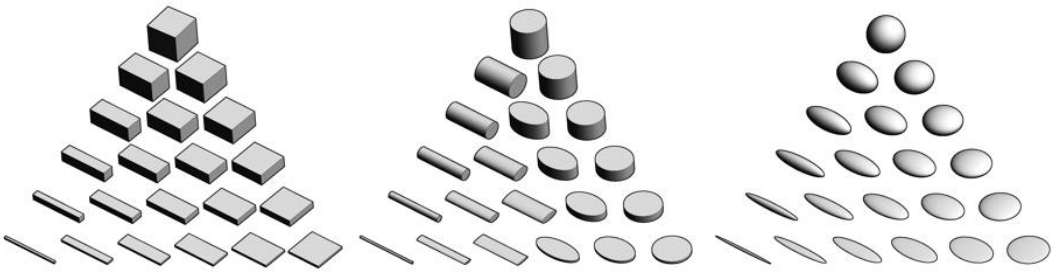


FIGURA 6. *Exemples de glyphs. Formes del tensor representades a partir de cuboides, cilindres i el·lipsoides respectivament.*

Per a obtenir més informació, es poden pintar els glyphs, al igual com passava en la representació 1D.
Aquests mètodes són eficaços quan es tracten regions petites, ja que si s'ha de tractar una regió massa gran, l'acumulació d'informació faria que es fes molt difícil d'interpretar els resultats.

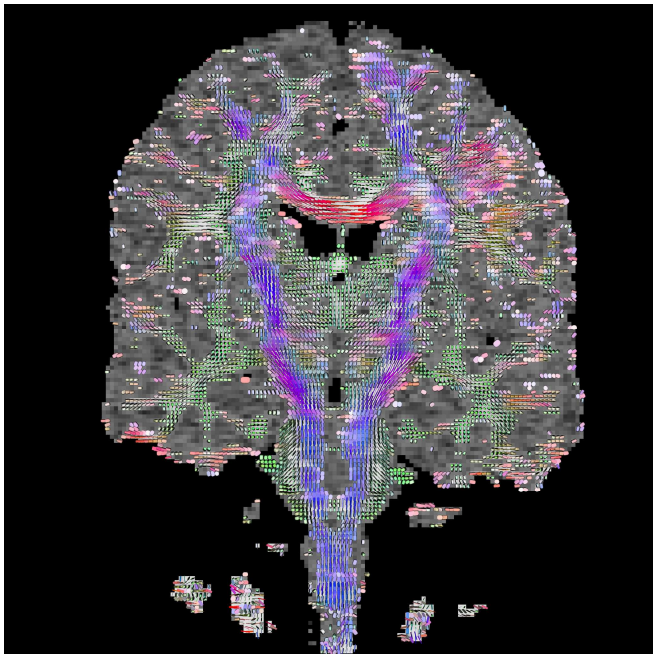


FIGURA 7.

Exemple de visualització 2D, on es poden apreciar els glyphs dels que es parla sobre aquesta imatge.

- **Mètodes 3D**. Aquests mètodes són també anomenats mètodes de tracking de fibres. Els mètodes 3D són els que simulen el traçat de fibres del cervell amb l'objectiu de donar una visió 3D de la fibra.
  Les extensions de fibres en el cervell estan caracteritzades per una difusió anisotrópica i relativament alta en la direcció de la fibra. Per aquest motiu és possible extreure les extensions de fibra com a línies (*streamlines*) per a integrar-les en la direcció de la màxima difusió en regions amb una alta anisotropia.
  A partir d'un punt inicial reconstrueixen el camí que segueix una fibra al llarg del cervell i un cop s'aconsegueix aquest camí s'aplica un algorisme de visualització.

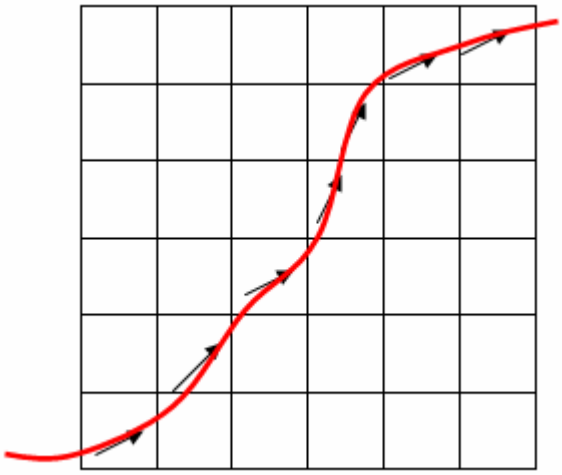


FIGURA 8. *Mapa de vectors*

En l'exemple de la *figura 8* es mostra el mapa de vectors principals on el punt inicial seria la cel·la 0,0 i, a partir d'aquesta, aniríem seguint per la cel·la següent en funció de la direcció del vector principal.
Podem veure pintada de vermell el que seria la fibra resultant d'aplicar l'algorisme de tracking.

En aquesta figura només hi ha marcades les direccions dels vectors per on passa la fibra.

Existeixen diverses formes de representar les fibres un cop han estat reconstruïdes. La forma més simple és representar-les a través de línies (*streamlines*), que donaria com a resultat el que s'explicava a la figura anterior. Hi ha altres tècniques més complexes, com podria ser representar-les mitjançant tubs (*streamtubs*). En aquesta tècnica s'aplica una distància que serà el diàmetre necessari per a la visualització del tub. Si es vol es pot visualitzar en forma de prisma, enlloc de fer-ho en forma cilíndrica. Una altra tècnica semblant és la tècnica de visualització amb *hyperstreamtubes* agafa el segon vector principal per a determinar quin serà el diàmetre del tub, de manera que el tub no tindrà una amplada uniforme, sinó que variarà en funció del tensor de difusió.

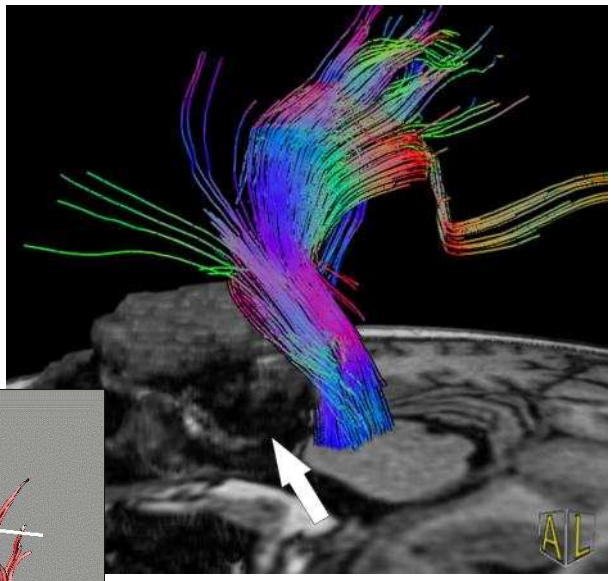


FIGURA 9.
Exemple de visualització 3D utilitzant streamlines.

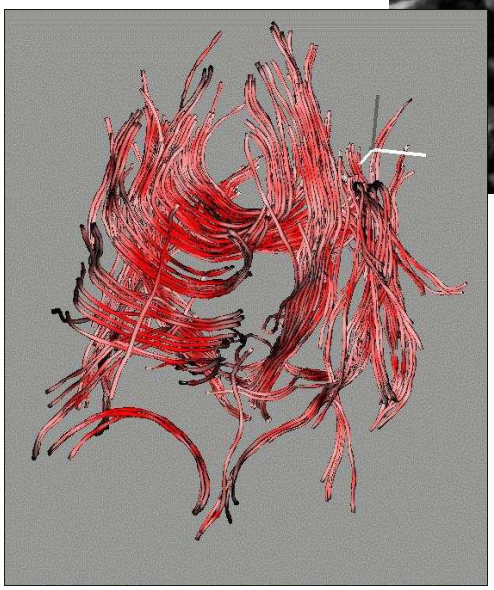


FIGURA 10.
Exemple de visualització 3D utilitzant streamtubes.

Com s'ha explicat anteriorment, el cervell conté una quantitat molt elevada de fibres, i és per aquest motiu que s'utilitzen algorismes d'agrupament de fibres segons les zones que connecten, ja que, sense aquests, els resultats poden ser molt confusos. Els algorismes de clustering s'expliquen en l'apartament de segueix.

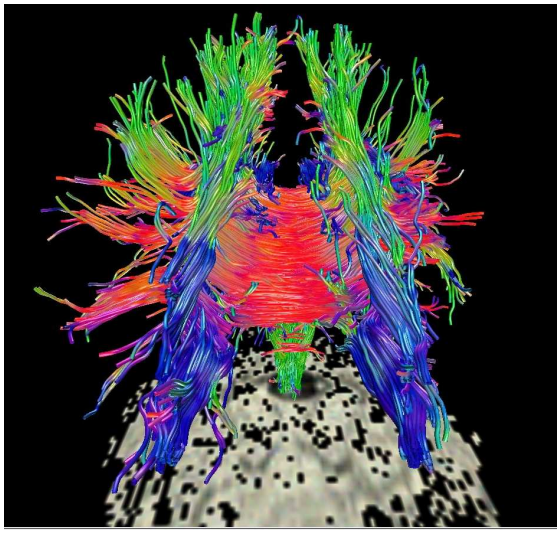


FIGURA 11.
Exemple de visualització 3D

2.3.- Clustering de fibres

Com s'ha explicat anteriorment, al cervell hi ha un nombre molt gran de fibres i, sense fer un tractament previ, aquesta visualització no ens aporta massa informació, ja que el resultat que ens dóna el tracking de fibres és la visualització de les fibres una a una, però no veurem cap relació entre elles.

La solució que proposem és fer un agrupament de les fibres en funció de la part del cervell que connectin. D'aquesta manera separarem les extensions de fibra que siguin diferents anatòmicament i ens quedaran agrupades en bandes, la qual cosa permetrà la valoració de connectivitat anatòmica entre diferents regions cerebrals. A la *figura 12* es veu la diferència entre el tracking i el clustering que es fa a partir d'aquest.

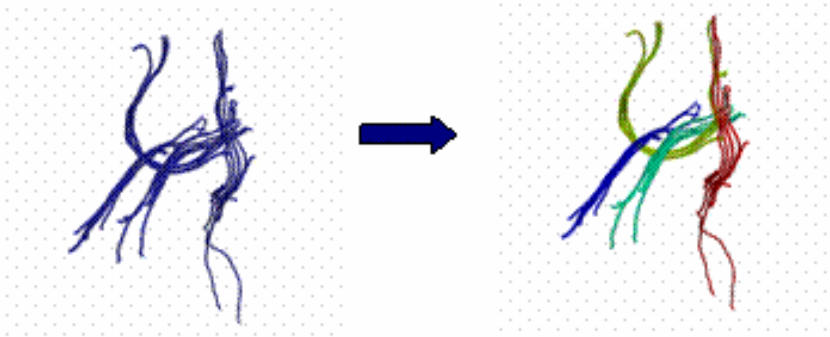


FIGURA 12. *Tracking de fibres (esquerra) i el clustering de fibres (dreta).*

2.4.- Descripció general d'algorisme de clustering.

Un algorisme de clustering es defineix com la classificació d'objectes en diferents grups no definits a priori o, més exactament, com la divisió d'un conjunt de dades d'entrada en subconjunts, de manera que les dades de cada subconjunt comparteixen alguna característica comuna. Normalment la característica que s'utilitza per a fer l'agrupament és la proximitat entre els objectes segons alguna mesura de distància definida, però també es pot agafar com a característica un paràmetre que permeti diferenciar els objectes d'alguna manera. Per exemple, una baralla de cartes espanyola es podria dividir de diferents maneres: en quatre clústers si agaféssim el pal com a criteri, en vuit clústers si agaféssim el pal i a més diferenciéssim les figures del números. Per tant, tot depèn de que considerem similar.

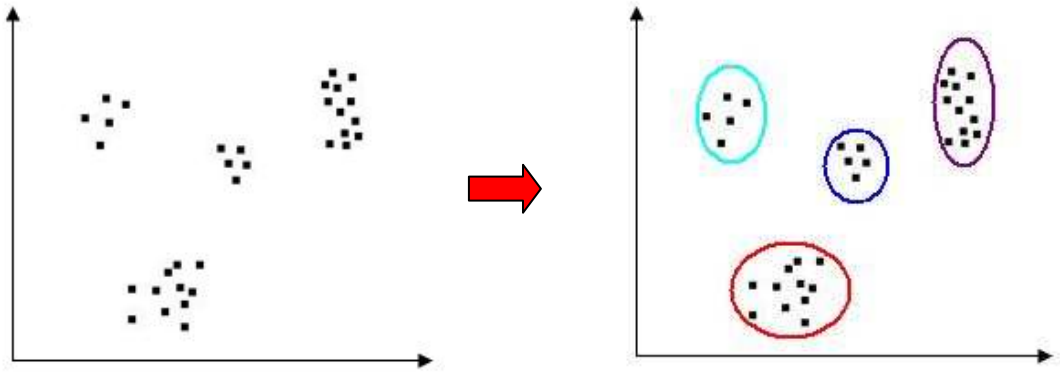


FIGURA 13. *Funcionament d'una tècnica de clustering.*

A la *figura 13* tenim el funcionament d'una tècnica de clustering. A simple vista podem identificar 4 grups en els quals es poden dividir les dades. El criteri de similitud és la distància.

Sota aquest nom es pot englobar qualsevol procediment que tingui un conjunt mostral com a informació d'entrada i un cert nombre de subconjunts del mateix com a sortida.

La idea d'aquests algorismes és que els objectes d'un mateix clúster tinguin una alta similitud entre ells i una similitud baixa amb objectes d'altres clústers.

A continuació veiem una llista d'objectius que es volen assolir amb la tècnica de clustering:

- Funcionament tot i disposar de poques dades.

- Capacitat de manipular diferents tipus d'atributs: numèrics (els més habituals), binaris, nominals, ordinals, etc.
- Les formes dels clústers resultants siguin arbitràries. Per exemple, els mètodes basats en distàncies numèriques tendeixen a trobar clústers esfèrics.
- Possibilitat d'especificar els paràmetres que convinguin, com podria ser el nombre de clústers.
- Manipulació del soroll. Hi ha moltes tècniques que són sensibles a dades errònies.
- Independència de l'ordre de les dades.
- Poder funcionar eficientment amb una dimensionalitat alta.
- Capacitat d'afegir restriccions.
- Aconseguir que els clústers siguin interpretables i utilitzables.

Com es pot suposar no estem parlant d'una tècnica perfecta, sinó que a part dels avantatges també comporta una sèrie de problemes:

- Operar amb un elevat nombre de dimensions i dades d'entrada pot causar un problema pel que fa al temps.
- L'efectivitat del mètode depèn de la mesura de distància que s'esculli.
- Si no existeix una mesura de distància obvia, podrem definir-la, la qual cosa no sempre és fàcil, especialment si estem tractant espais multidimensionals.
- El resultat d'un algorisme de clustering pot ser interpretat de diferents maneres.

El clustering es fa servir habitualment per a l'anàlisi de dades estadístiques i és utilitzat en molts camps com podrien ser la mineria de dades, el reconeixement de models, l'anàlisi d'imatges i la bioinformàtica.

Com es pot suposar existeixen moltes variants de tècniques que compleixen aquesta condició. Quan es parla d'aquestes tècniques s'ha de tenir present que hi ha dos punts claus:

- Com fer l'agrupament dels objectes, que en el nostra cas seran les fibres. Aquest punt el resolen els diferents algorismes de clustering que definirem en el següent apartat.

- La mesura de distància que s'utilitzarà per a fer l'agrupament del que estem parlant. En l'apartat 2.6, descrivim amb detall una sèrie de mesures de distància que són les que s'implementaran en aquest projecte.

2.5.- Classificació dels algorismes de clustering

Els algorismes d'agrupament de fibres (o algorismes de clustering) imposen un tipus de classificació sobre les dades d'entrada. Aquesta classificació pot prendre diverses formes, depenen del tipus d'algorisme que escollim. Podem classificar els diferents tipus d'algorismes de clustering de la següent manera:

- Algorismes exclusius
- Algorismes de solapament
- Algorismes jeràrquics
- Algorismes probabilístics

En els següents apartats explicarem breument en que consisteixen cadascun d'aquests algorismes per a poder apreciar les diferències que existeixen entre ells.

Independentment de l'algorisme que se seleccioni hem de tenir en compte que necessitarem algun mètode per a decidir a quin clúster pertany una fibra. Ens referim als mètodes de distàncies entre fibres, que s'explica en el punt 2.6.

2.5.1.- Algorismes de clustering exclusius

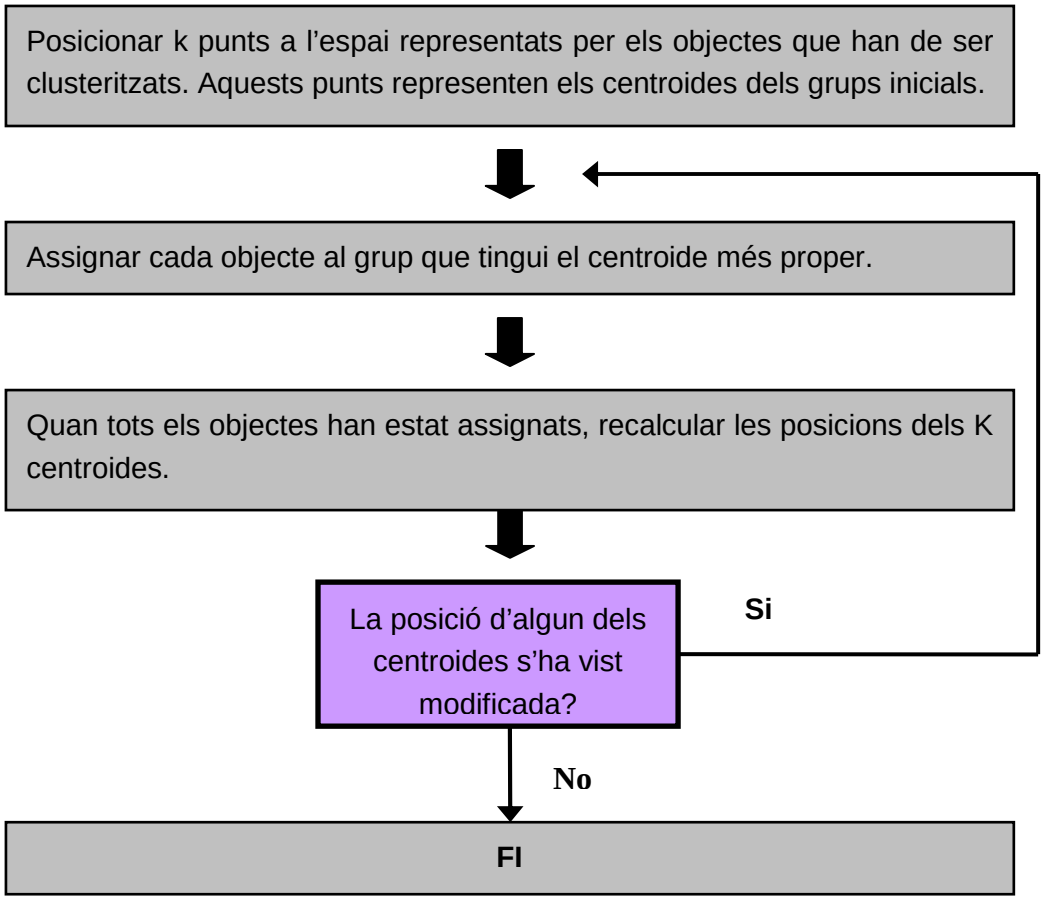
En aquest tipus d'algorismes les dades s'agrupen de forma exclusiva, de manera que si una dada concreta pertany a un clúster definit llavors aquesta no pot ser inclosa en un altre clúster.
En anglès també són coneguts com a hard clustering, degut a que formen part d'un i només d'un grup.
Com a exemple d'aquest tipus proposem estudiar el mètode k-means.

2.5.1.1.- L'algorisme k-means

L'algorisme k-means resol el conegut problema de clustering.
El procediment segueix una forma simple i fàcil per a classificar el conjunt de dades donades a través d'un número de clústers fixat a priori. Anomenarem k a aquest nombre. La idea principal és definir tants centroides com clústers, és a dir, *k*.

Aquests centroides haurien de ser col·locats d'una manera apropiada ja que, el resultat varia en funció de la col·locació inicial. Així, la millor opció és col·locar-los el més lluny possible els uns dels altres.
El pròxim pas és agafar cada punt pertanyent al conjunt de dades i associar-los al centroide més pròxim. Quan ja no queden més punts per a analitzar, el primer pas ha finalitzat i ja disposem d'un primer agrupament. En aquest punt necessitem recalcular *k* nous centroides com a baricentres dels clústers resultants del pas previ. Després d'aconseguir aquests k nous centroides, un nou agrupament ha de ser donat entre els mateixos punts del conjunt de dades i el nou centroide més proper. El que fem és entrar en un bucle fins que no hi hagi més canvis, és a dir, fins que els centroides no es moguin.

Esquemàticament l'algorisme quedaria com es mostra a l'*esquema 1*.



ESQUEMA 1. *L'algorisme k-means*

Cal esmentar que tot i que el k-means hagi finalitzat no significa que hagi trobat la configuració òptima ja que es fortament dependent de la mesura de distància escollida. A més, l'algorisme és especialment sensible a la selecció aleatòria dels centres dels clústers inicials. Per a reduir aquest efecte es pot pensar en executar l'algorisme múltiples vegades.

2.5.2.- Algorismes de clustering de solapament

D'altra banda els algorismes de solapament utilitzen conjunts fuzzy per a agrupar les dades, de manera que cada punt pot pertànyer a dos o més clústers amb diferents nivells de pertinència. En aquest cas, les dades s'associaran a un valor de pertinència apropiat, tenint en compte que aquests indiquen la força de l'associació entre l'element tractat i el clúster en concret.
Podem definir els algorismes fuzzy com un procés d'assignació d'aquests nivells de pertinença que un cop trobats s'usen per assignar els objectes a un o més clústers.
Com a exemple d'aquest tipus proposem estudiar el mètode fuzzy c-means.

2.5.2.1.- L'algorisme Fuzzy c-means

El fuzzy c-means (FCM) és un mètode de clustering que permet a un objecte pertànyer a dos o més clústers. Aquest mètode és freqüentment utilitzat en el reconeixement de patrons.
El seu funcionament es basa en assignar a cadascuna de les dades o objectes d'entrada un grau de pertinença als clústers, com en la lògica fuzzy, en comptes de fer que cadascun pertanyi a un únic clúster com passaria en algorismes de tipus exclusiu. D'aquesta manera els punts situats al límit d'un clúster hi pertanyen amb un grau menor que els punts que es troben situats al centre del clúster. Per a cada punt x, obtenim un coeficient a partir del grau de pertinença al clúster concret. Normalment, la suma d'aquests coeficients és 1.

$$\forall x \quad \sum_{k=1}^{\text{num. clusters}} u_k(x) = 1 \qquad \text{center}_k = \frac{\sum_x u_k(x)^m x}{\sum_x u_k(x)^m}$$

$$u_k(x) = \frac{1}{d(\text{center}_k, x)} \qquad u_k(x) = \frac{1}{\sum_j \left(\frac{d(\text{center}_k, x)}{d(\text{center}_j, x)}\right)^{2/(m-1)}}$$

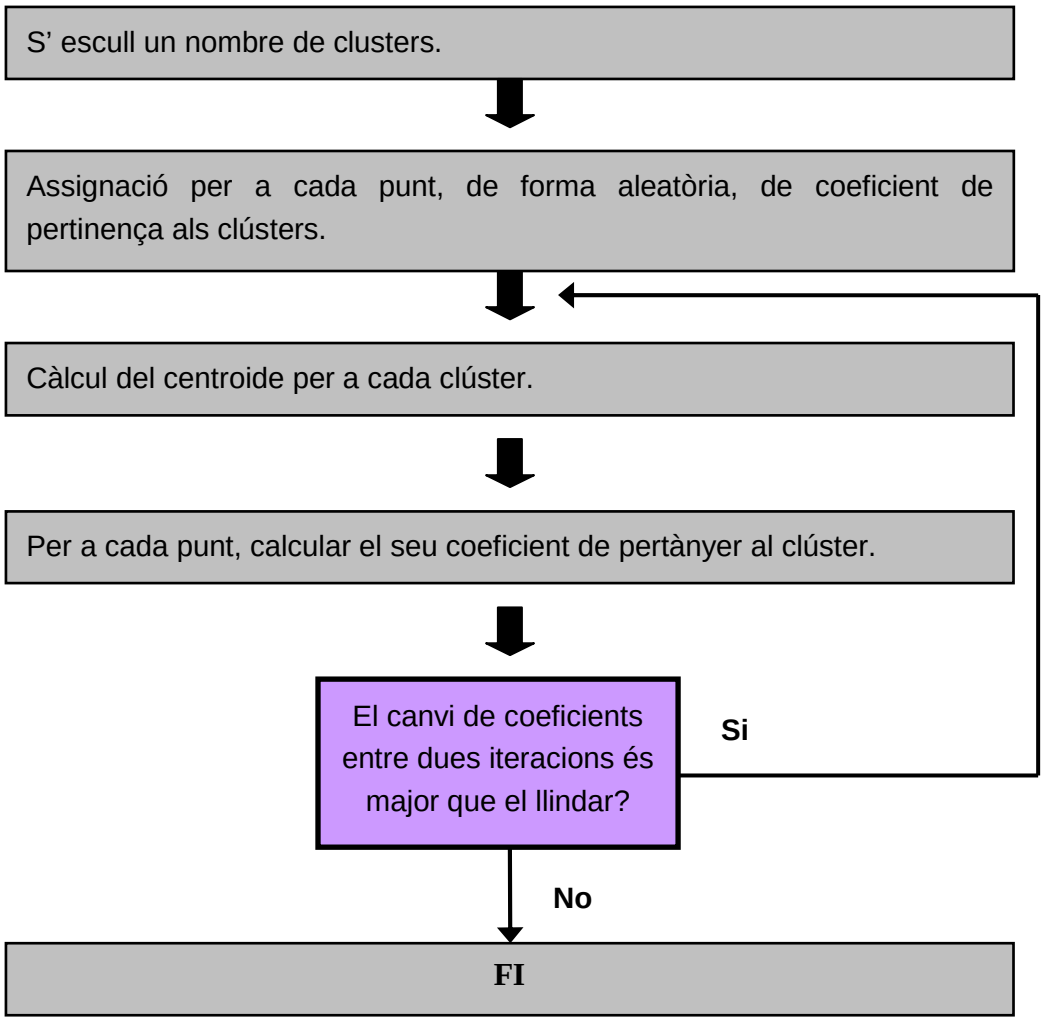
FIGURA 14. *Fórmules necessàries per a l'algorisme fuzzy c-means.*

Les fórmules que veiem a la *figura 14* ens indiquen el següent:

- La suma dels coeficients és igual a 1.
- El centroide del clúster és la mitjana de tots els punts, ponderats pels seus graus de pertinença al clúster.

- El grau de pertinença és la inversa de la distància al clúster.
- Llavors els coeficients són normalitzats, essent el paràmetre m un real més gran que 1. D'aquesta manera la suma de coeficients serà 1.

Esquemàticament l'algorisme quedaria com es mostra a l'*esquema 2*:



ESQUEMA 2. *L'algorisme fuzzy c-means*

L'algorisme minimitza la variància intra-cluster però té el mateix problema que el k-mean, el mínim és un mínim local, i els resultats depenen de l'elecció de pesos inicial.

2.5.3.- Algorismes de clustering jeràrquic

Donat un conjunt d'entrada S, l'objectiu del clustering jeràrquic és produir una jerarquia tal i com el seu nom indica. El resultat es pot donar en forma d'arbre en el qual els nodes representen subconjunts de S. Es formarà una estructura d'arbre on:

- El node arrel fa referència a tot el conjunt
- Els nodes terminals (fulles) són els elements individuals d'*S*.
- Els nodes interns són creats per la unió dels seus fills.

Dins dels mètodes jeràrquics en trobem de dos tipus: els aglomeratius i els divisius.
L'aglomeratiu consisteix en fer una classificació de baix cap a dalt (bottom-up) i segueix el següent patró:

1. Posa cada ítem en un clúster individual.
2. Uneix els dos clústers més semblants.
3. Repeteix el pas 2 fins que només quedi un sol clúster.

El divisiu fa el procés invers a l'aglomeratiu, és a dir, de dalt cap a baix (top-down) i procedeix començant amb un únic clúster on es posen tots els ítems i anar dividint fins que hi hagi tants clústers com ítems.
En qualsevol cas, d'ambdós mètodes se'n deriva un dendograma, que és un gràfic que il·lustra com es van formant les subdivisions o els agrupaments, etapa a etapa.

Esquemàticament l'algorisme quedaria com es mostra a l'*esquema 3*.

Es poden considerar diferents variacions d'aquestes dues tècniques en funció del mètode d'unió (linkage method) que triem. A continuació fem una breu descripció dels més importants:

- En el <u>single-link</u> la distància entre dos clústers és la distància entre el parell d'ítems més propers tal i com es mostra a la *figura 15*. Aquest mètode funciona bé per a clústers allargats i ben separats i pot trobar clústers de diferents mides i formes complexes. Per altra banda, no funciona correctament quan tenim dades amb soroll.

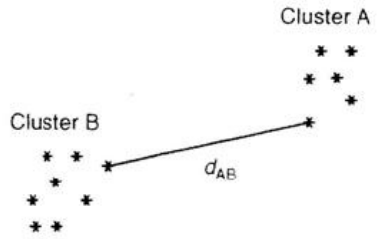


FIGURA 15. *Distància single-link*

– En el complete-link la distància entre dos clústers es defineix com la màxima distància entre parell d'ítems tal i com es mostra a la *figura 16*.

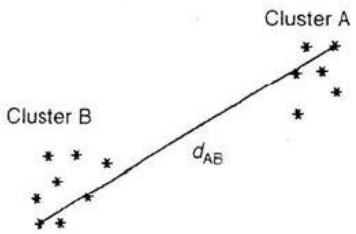


FIGURA 16. *Distància complete-link*

– Existeix també un terme mig entre el single-link i el complet-link, que s'anomena weightet-average algorithm en el qual la distància entre clústers es defineix com la mitjana de les distàncies mínimes i màximes entre parells d'ítems tal i com es mostra a la *figura 17*.

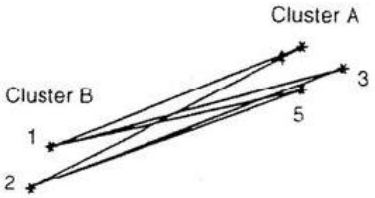
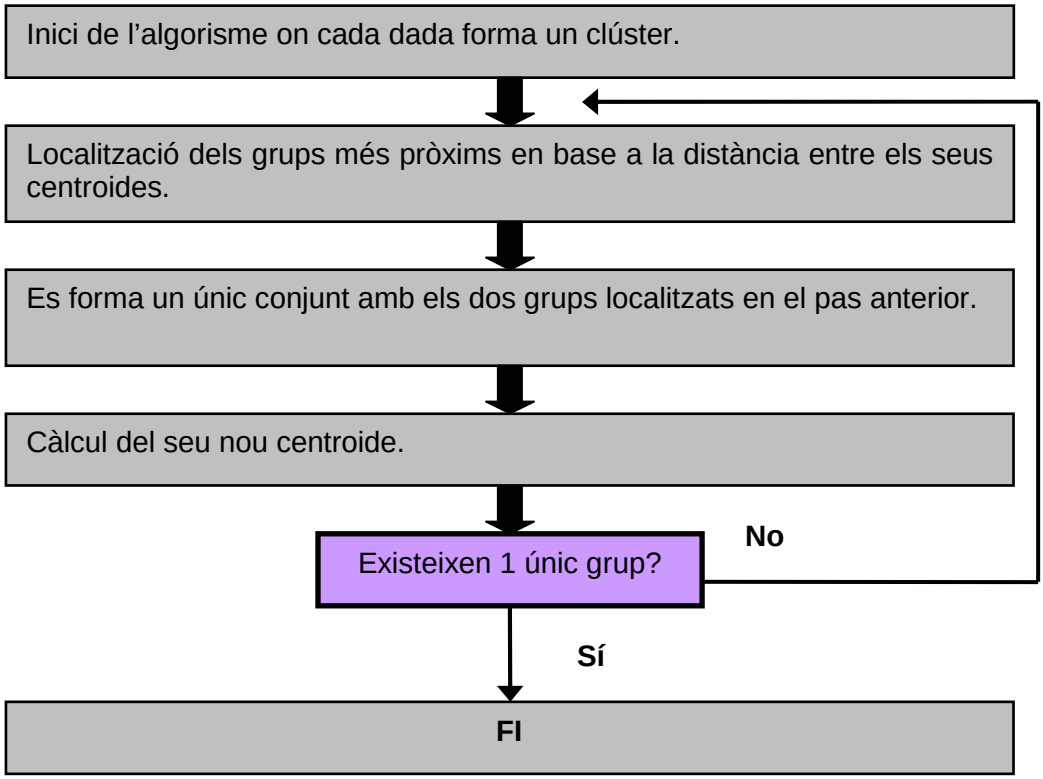


FIGURA 17. *Distància weightet-average*



ESQUEMA 3. *L'algorisme jeràrquic*

2.5.4.- Algorismes de clustering probabilístic

Aquest tipus de mètodes utilitzen una aproximació totalment probabilística, tal i com el seu nom indica.

Els algorismes explicats abans presenten certs defectes, entre els que destaquen la dependència que té el resultat de l'ordre dels exemples i la tendència d'aquests algorismes al sobreajustament. Una aproximació estadística al problema del clustering resol aquests problemes. Des d'aquest punt de vista, el que es busca és el grup de clústers més probables un cop donades les dades. Ara, els exemples tenen certes probabilitats de pertànyer a un clúster. La base d'aquest tipus de clustering es troba en un model estadístic anomenat barreja de distribucions (finite mixtures), on cada distribució representa la probabilitat de que un objecte tingui un conjunt particular de parells atribut-valor, si se sabés que és membre d'aquell clúster. Es tenen k distribucions de probabilitat que representen els k clústers. La barreja més senzilla es té quan els atributs són numèrics amb distribucions gaussianes. Cadascuna de les distribucions (normals) es caracteritza per dos paràmetres: la mitjana (μ) i la variància ($\sigma 2$). A més, cada distribució tindrà certa probabilitat d'aparició p, que serà determinada per la proporció d'exemples que pertanyen al clúster tractat respecte del número total d'exemples. En aquest cas, si hi ha k clústers, s'haurà de calcular un total de 3k-1 paràmetres: les k mitjanes, k variàncies i k-1 probabilitats de la distribució ja que la suma de les probabilitats ha de ser 1, de manera que conegudes k-1 es podrà determinar la k-enèssima.
Si es conegués el clúster al que pertany, en un principi, cadascun dels exemple d'entrenament seria molt senzill obtenir els 3k-1 paràmetres necessaris per a definir totalment les distribucions d'aquests clústers, ja que simplement s'aplicarien equacions de la mitjana i la variància per a cadascun dels clústers. A més, per a calcular la probabilitat de cadascuna de les distribucions únicament es dividiria el número d'exemples d'entrenament que pertanyen al clúster en qüestió entre el número total d'exemples d'entrenament. Un cop obtinguts aquests paràmetres, si es desitgés calcular la probabilitat de pertinença d'un determinat exemple de test a cada clúster, simplement s'aplicaria el teorema de Bayes a cada problema concret i quedaria la següent equació:

$$P(A \mid x) = \frac{P(x \mid A)P(A)}{P(x)} = \frac{f(x;\mu_A,\sigma_A)p_A}{P(x)}$$

En aquesta equació A és un clúster del sistema, x l'exemple de test, p_A la probabilitat del clúster A i $f(x;\mu A,\sigma A)$ la funció de la distribució normal del clúster A, que s'expressa amb l'equació que es mostra a continuació.

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Tot i això, el problema es que no se sap de quina distribució ve cada dada i es desconeixen els paràmetres de les distribucions. És per això que s'adopta el procediment utilitzat per l'algorisme de clustering k-means (explicat a l'apartat 2.5.1) i s'itera.

2.6.- Mètodes de distàncies entre fibres

Els mètodes de clustering agrupen coses que són similars en alguna de les seves característiques i, per tant, aquest necessiten una forma de mesurar la similitud entre els objectes.

No hi ha una manera estàndard de calcular la proximitat entre un parell de fibres. Calcular la proximitat entre un parell de punts és una tasca fàcil si ens ajudem de la distància euclídea. Però hem de pensar que cada fibra pot estar composta per un nombre molt elevat de punts.

A continuació veiem una sèrie de distàncies entre fibres.

2.6.1.- Distància dels punts més propers

Aquest mètode ens dóna una informació molt poc detallada sobre la similitud entre dues fibres.
Es basa en determinar quina és la distància mínima entre els punts de les dues fibres que estem analitzant.
Segueix la següent fórmula:

$$d_c(F_i, F_j) = \min_{p_k \in F_i, p_l \in F_j} \| p_k - p_l \| .$$

La *figura 18* mostra el comportament del mètode de la distància dels punts més propers de forma gràfica. Per cada punts de la fibra A es calcula la distància a tots els punts de la fibra B. S'agafa la distància mínima entre totes les obtingudes. En el dibuix la distància mínima està marcada en vermell.

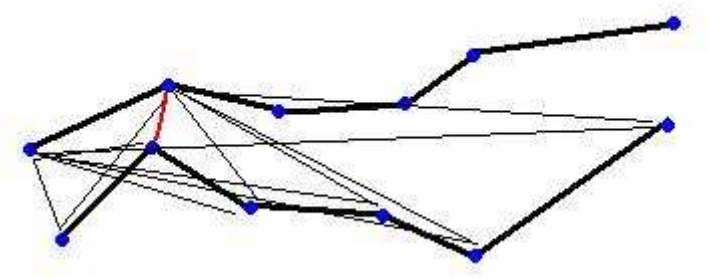


FIGURA 18.
Mètode de la distància dels punts més propers.

Aquest mètode no és vàlid per a fibres de diferent llargada.

2.6.2.- Mitjana de les distàncies de punts més propers

Cada punt de la fibra és mapejat amb el punt més proper de l'altre fibra, formant així, parells de punts.
La distància de la fibra es defineix com la mitjana de les distàncies dels parells de punts més propers.
Segueix la següent fórmula:

$$d_M(F_i, F_j) = \text{mean}(d_m(F_i, F_j), d_m(F_j, F_i))$$

amb

$$d_m(F_i, F_j) = \underset{p_l \in F_i}{\text{mean}} \ \underset{p_k \in F_j}{\min} \ \| \ p_k - p_l \ \| \ .$$

La *figura 19* mostra el comportament del mètode de la mitjana de les distàncies de punts més propers de forma gràfica. S'agafa un punt de la fibra A i es calcula la distància d'aquest punt a tots els punts de la fibra B. S'agafa la distància al punt més proper (marcada en vermell). Es repeteix el pas per a la resta de punts de la fibra A. Tindrem tantes distàncies mínimes com punts tingui A. Llavors s'agafa com a distància la mitjana de totes les distàncies mínimes.

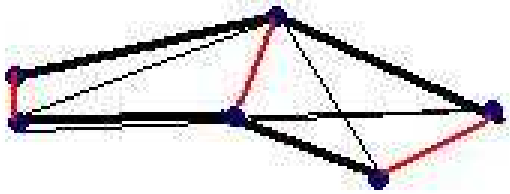


FIGURA 19. *Mètode de la mitjana de les distàncies de punts més propers.*

Aquesta distància serà més aproximada a la realitat que no pas la anterior, però també té el problema de no saber acceptar una fibra que pertany a una mateixa banda que una altra si la llargada d'aquestes és diferent.

2.6.3.- Distància de Hausdorff

Aquesta distància és molt conservativa ja que dues fibres es consideren de la mateixa banda només si totes les distàncies entre parells de punts més propers són petites.
La distància de Hausdorff és la màxima distància entre els dos parells de punts més propers.
Segueix la següent fórmula:

$$d_H(F_i, F_j) = \max(d_h(F_i, F_j), d_h(F_j, F_i))$$

amb

$$d_h(F_i, F_j) = \max_{p_k \in F_i} \min_{p_l \in F_j} \| p_k - p_l \| .$$

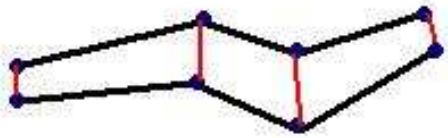


FIGURA 20. *Mètode de la distància de Hausdorff. Exemple 1.*

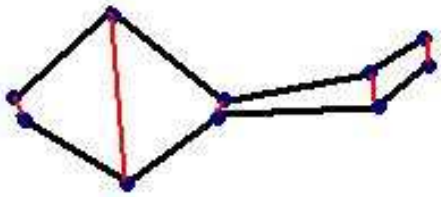


FIGURA 21. *Mètode de la distància de Hausdorff. Exemple 2.*

La *figura 20* i la *figura 21* mostren el comportament del mètode de la distància de Hausdorff de forma gràfica.
A la *figura 20*, les fibres es considerarien del mateix clúster ja que totes les distàncies entre punts més propers són petites.
Per contra, a la *figura 21*, determinaríem que les fibres pertanyen a clústers diferents perquè hi ha una distància entre punts més propers que és força major que les altres.

Aquesta mesura tendeix a sobreestimar la distància entre fibres.

2.6.4.- Distància dels punts finals

Es considera que dues fibres pertanyen a la mateixa banda si els punts finals d'aquestes són propers, ja que es pensa que les fibres d'una mateixa estructura anatomical connecten les mateixes àrees del cervell.
Segueix la següent fórmula:

$$d_1 = \| f_{i,1} - f_{j,1} \| + \| f_{i,end} - f_{j,end} \|$$
$$d_2 = \| f_{i,1} - f_{j,end} \| + \| f_{i,end} - f_{j,1} \|$$
$$D_E(i,j) = \min(d_1, d_2).$$

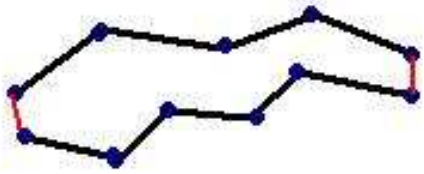


FIGURA 22.
Mètode de la distància dels punts finals.

La figura 22 mostra el comportament del mètode de la distància dels punts finals de forma gràfica. *S'agafa la distància entre el punt inicial de la fibra A i el punt inicial de la fibra B i la distància entre el punt final de la fibra A i el punt final de la fibra B i se sumen.*
Podríem assegurar que pertanyen al mateix clúster, ja que la distància que ens dóna es veu que és petita.

Podem trobar problemes utilitzant aquesta distància quan les fibres que estem tractant estan fetes malbé o quan fibres de diferents bandes comencen i acaben aproximadament a la mateixa regió.

2.6.5.- Distància sobre el llindar

Amb aquest mètode es defineix la dissimilitud entre dues fibres amb la següent fórmula:

$$D_t = \frac{\int_s \max(\mathrm{dist}(s) - T_t, 0)\mathrm{d}s}{\int_s \max\left(\frac{\mathrm{dist}(s)-T_t}{|\mathrm{dist}(s)-T_t|}, 0\right)\mathrm{d}s}.$$

En aquesta equació, la *s* parametritza la llargada de l'arc de la fibra més curta i *dist(s)* és la menor distància euclidea de la localització s de la fibra més curta i la fibra més llarga. T_t és un llindar i només les fibres que estiguin per sobre d'aquest llindar contribueixen a la distància.

En aquest cas les fibres no tenen perquè ser de la mateixa llargada per a que es consideri que pertanyen a la mateixa banda. Així, aquí sí ens funcionaria en cas de tenir fibres que estiguin fetes malbé.

El problema que té és que si dues fibres de diferents llargades realment sí pertanyen a bandes diferents pot arribar a considerar-les com si fossin de la mateixa.

2.7.- Visualització de clústers

Ja hem explicat com es fa el traçat de fibres (tècniques 3D a *l'apartat 2.2*) i com fer l'agrupament d'aquestes (*apartat 2.5*) fent ús d'alguna mesura de distància (*apartat 2.6*). Arribat aquest punt, només falta explicar com es fa la visualització dels *clústers*. Explicarem dues maneres de fer aquesta visualització:

- Pintant cadascuna de les fibres d'un color segons el *clúster* al que pertanyen. A la *figura 23* cadascun dels colors identifica una funció de les fibres segons la zona que connecten.

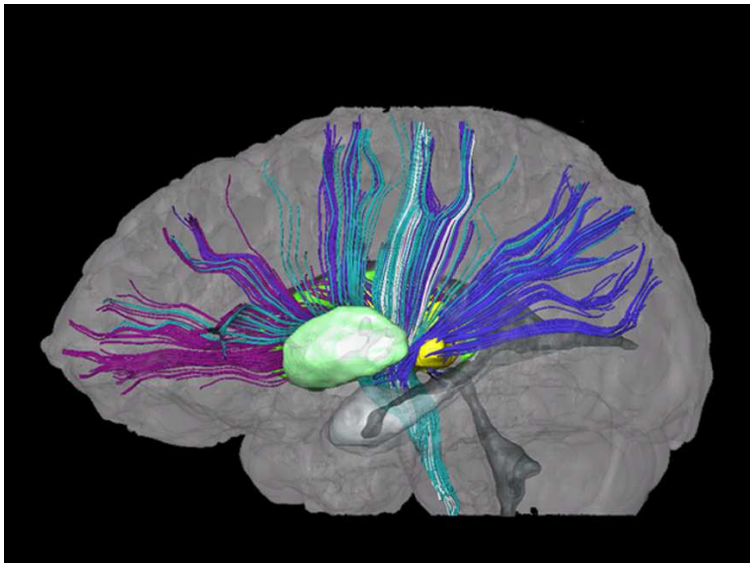


FIGURA 23. *Visualització de clústers.*

- Agrupant totes les fibres d'un mateix *clúster* i visualitzar-les com una peça única; com un tub. En aquest cas cada tub serà pintat d'un color diferent, tal com passa en el cas de pintar les fibres una a una.
  Per a aquesta tècnica podríem utilitzar els *streamtubes* (Veure *figura 24*) dels que hem parlat anteriorment, que els podem utilitzar per a agrupar en un tub un conjunt de *streamlines*, que serien les fibres en sí.

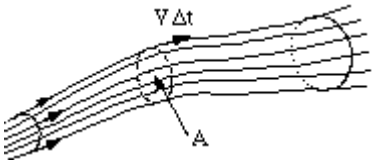


FIGURA 24.
Concepte d'streamtube.

Tot i complir amb el requeriment que busquem, s'ha de tenir en compte que és molt important estalviar en temps computacional, i és per aquest motiu, que el que utilitzarem seran *streamsurfaces* enlloc de *streamtubs*.

El motiu d'utilitzar *streamsurfaces* radica en la idea de per què hem de calcular tot el tub si en realitat només en veurem una cara? Com el seu nom indica els *streamsurfaces* pinten una superfície. Llavors per a fer que sembli un tub, s'afegiran unes el·lipses als extrems i es farà un degradat del color per a donar-li sensació de profunditat.

A la *figura 25* es pot veure com les zones no estan representades ni per línies ni per tubs, sinó que es pinta tota una superfície. Són les anomenades streamsurfaces i és el que volem aconseguir per a visualitzar les bandes.

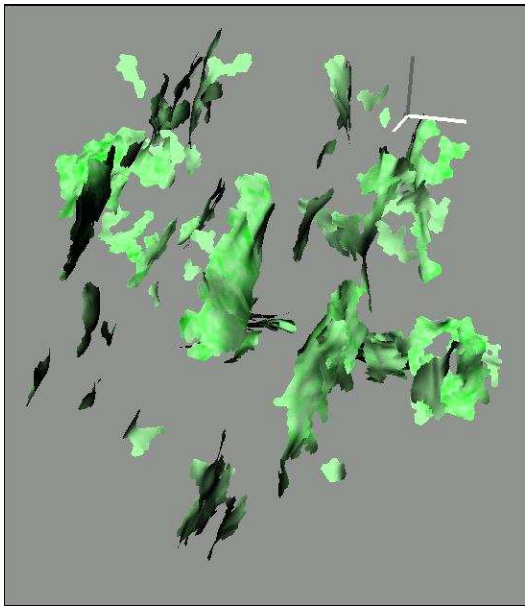


FIGURA 25.
Streamsurfaces.

3.- ESTUDI DE LA PLATAFORMA

En aquest apartat farem un estudi de l'estructura de la plataforma, identificant les diferents capes.
També coneixerem el llenguatge de programació que haurem de fer servir per a fer la implementació.

3.1. Estructura de la plataforma DTIWeb

L'aplicació DTIWeb segueix un patró de disseny per capes. Com es habitual en la majoria de projectes, aquestes capes són: la capa de dades, la capa d'aplicació i la capa de presentació.
A la *figura 26* podem veure el diagrama general de aplicació.

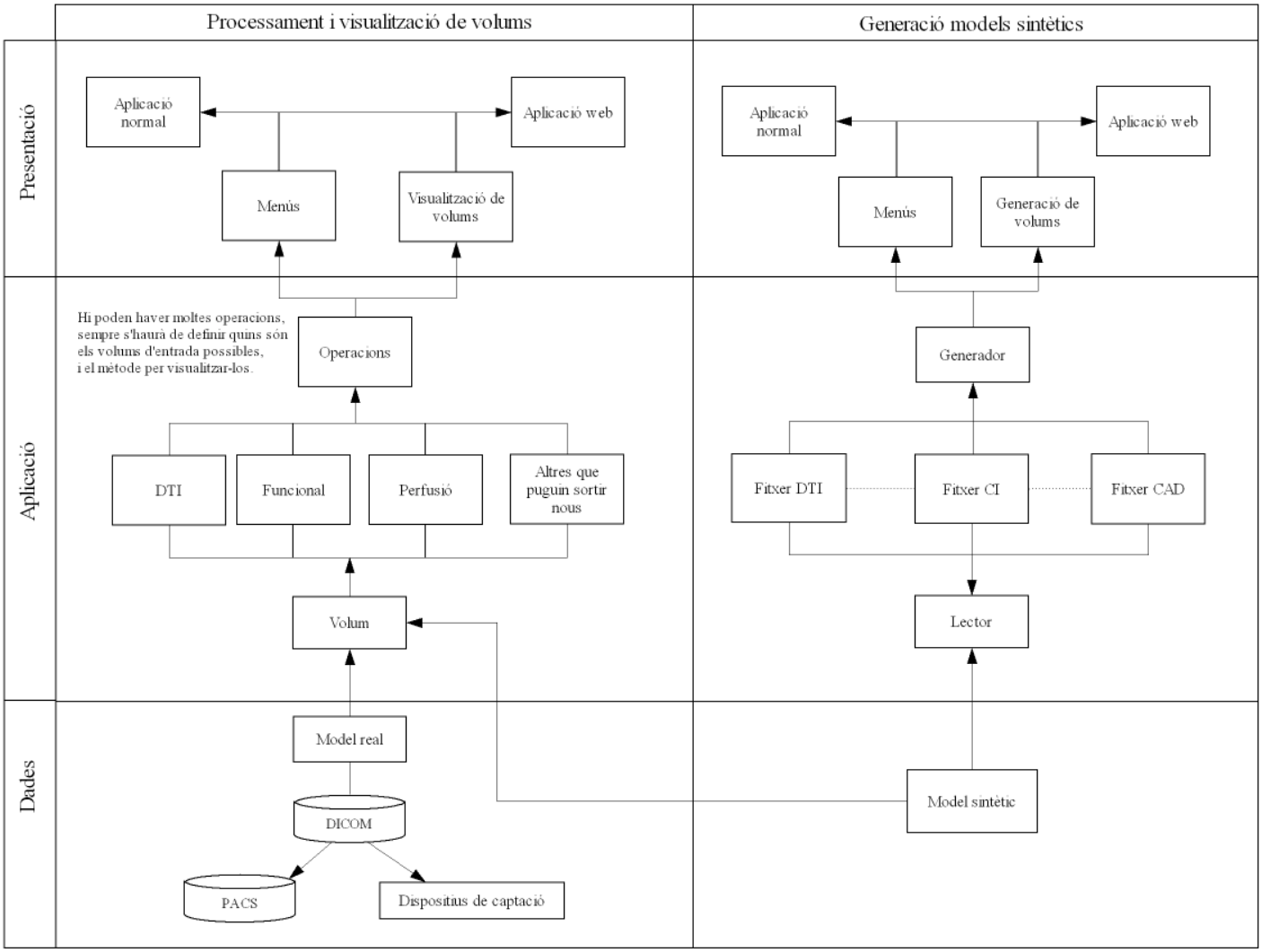


FIGURA 26. *Diagrama general de l'aplicació*

3.1.1.- Capa de dades

La capa de dades conté tots els mòduls relacionats amb la lectura i interpretació dels fitxers d'entrada. Aquests fitxers d'entrada estan formats per les dades que proporcionen els aparells de captació. També hi ha desenvolupades les funcionalitats que permeten guardar els models.

3.1.2.- Capa d'aplicació

La capa d'aplicació fa referència a la creació del model de vòxels (que s'ha explicat en l'apartat de l'estudi teòric), a les operacions que es poden realitzar sobre aquest model i a generar models sintètics.

Serà en aquesta capa on implementarem els mètodes de clustering i de distància.

3.1.3.- Capa de presentació

La capa de presentació correspon a tota la part que permet la visualització dels volums i a la part de la interfície gràfica.

Serà en aquesta capa on implementarem les funcionalitats de visualització dels clústers.

3.2. Llenguatge de programació

El llenguatge de programació que s'ha utilitzat per a implementar aquesta aplicació és el Java.

Java és un llenguatge de programació orientat a objectes desenvolupat per Sun Microsystems a principis dels anys 90.
Una de les característiques d'aquest llenguatge és que les seves aplicacions es compilen, normalment, en bytecode, tot i que la compilació en codi màquina també és possible. En temps d'execució, el bytecode és interpretat i compilat a codi natiu per a l'execució, tot i que l'execució directa per hardware del bytecode per un processador Java també és possible.

El model d'objectes és més simple que en llenguatges també molt importants com C i C++ i, a més, elimina eines de baix nivell com són els punters, que solen portar mal de caps als programadors.

Per a la part dels gràfics, s'han utilitzat les llibreries de Java 3D, que és una interfície de programació d'aplicacions 3D basades en escenes de grafs per a la plataforma Java.

4.- IMPLEMENTACIÓ

En aquest 4t apartat es realitza tota la part d'implementació.

En el primer subapartat d'anàlisis i requeriments generals s'estudia com s'introduiran les noves funcionalitats dins la plataforma i s'expliquen detalladament les noves funcionalitats mitjançant diagrames i fitxes de casos d'ús.

Un cop fetes aquestes consideracions, s'implementen els diferents mètodes de distància a partir d'un model amb dues fibres per veure resultats gràfics.
Es fa el mateix amb els algorismes de clustering. Aquests són testats a partir de models sintètics i models reals.

Després de realitzat proves amb els models, sorgeix un apartat d'optimització a partir de la reducció de les fibres.

Finalment, es comparen els algorismes de clustering i mètodes de distància triant diferents combinacions d'aquests.

4.1.- Anàlisis i requeriments generals

A partir dels objectius que ens hem proposat, explicats en la fase de l'estudi teòric, podem definir quines seran les funcions principals de l'usuari de l'aplicació.

A la *figura 27* es presenta el diagrama de casos d'ús de l'aplicació. Les noves funcionalitats s'hauran d'integrar a l'aplicació web ja existent. Podem veure en el diagrama les noves funcionalitats que incorporarà aquest projecte que estan marcades de color blau.

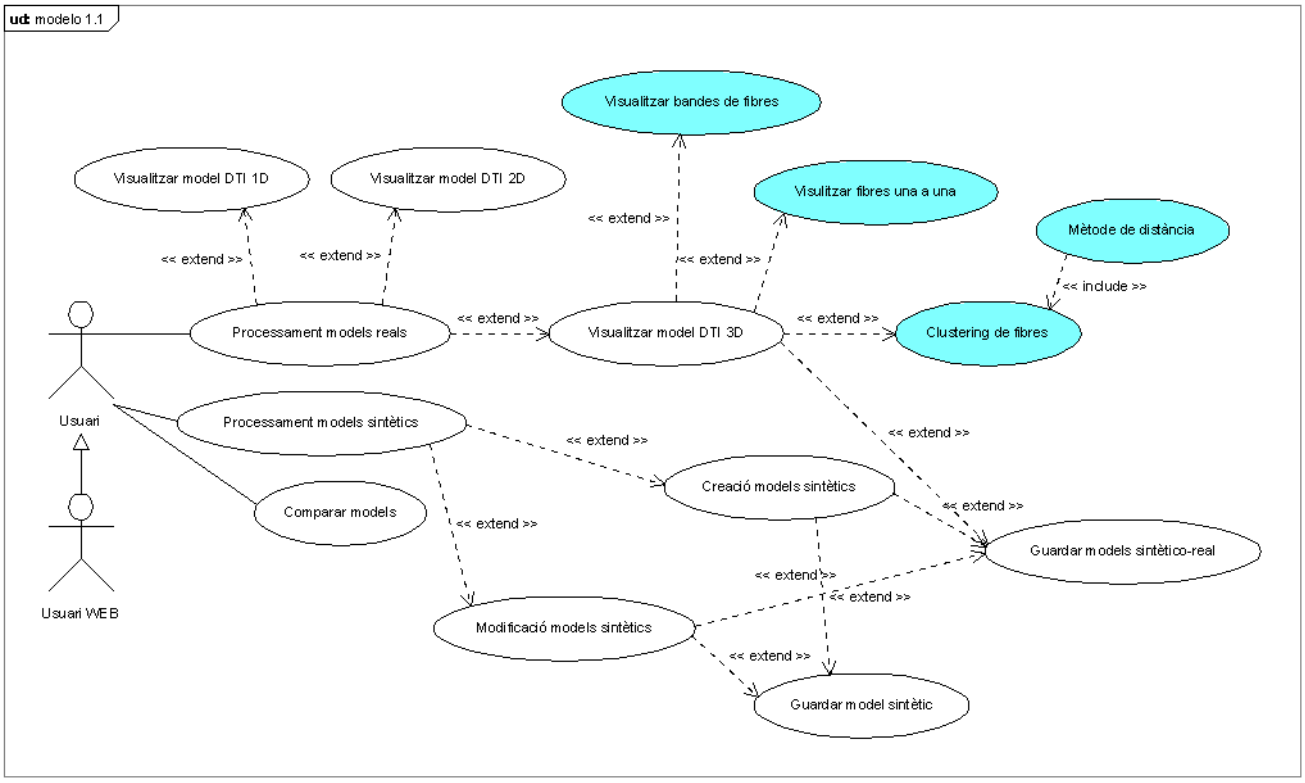


FIGURA 27. *Diagrama de casos d'ús general de l'aplicació*

Les funcionalitats de visualitzar bandes de fibres i visualitzar fibres una a una, fan referència a la part de visualització dels clústers.
Per altra banda les funcionalitats de clustering de fibres i distància de mesura, s'usaran per a fer l'agrupament de les fibres en les visualitzacions 3D sobre models DTI. Aquest dues darreres funcionalitats es presenten en el *diagrama de casos d'ús 1* i *diagrama de casos d'ús 2,* de forma més detallada, respectivament.

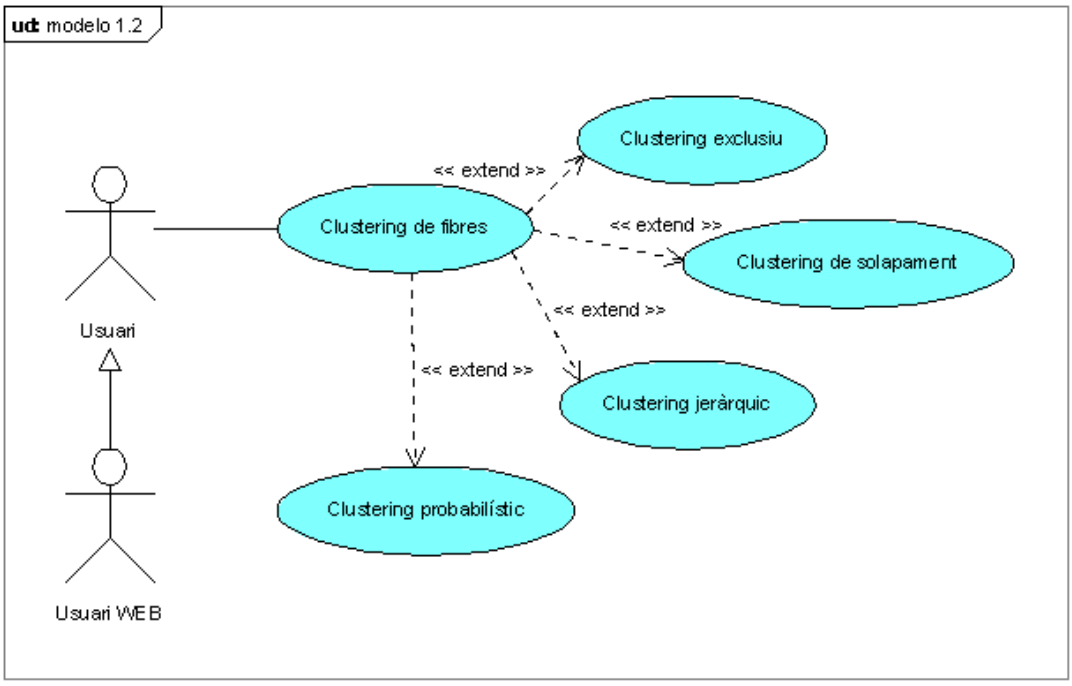


Diagrama de casos d'ús 1. *Funcionalitat 'clustering de fibres'*

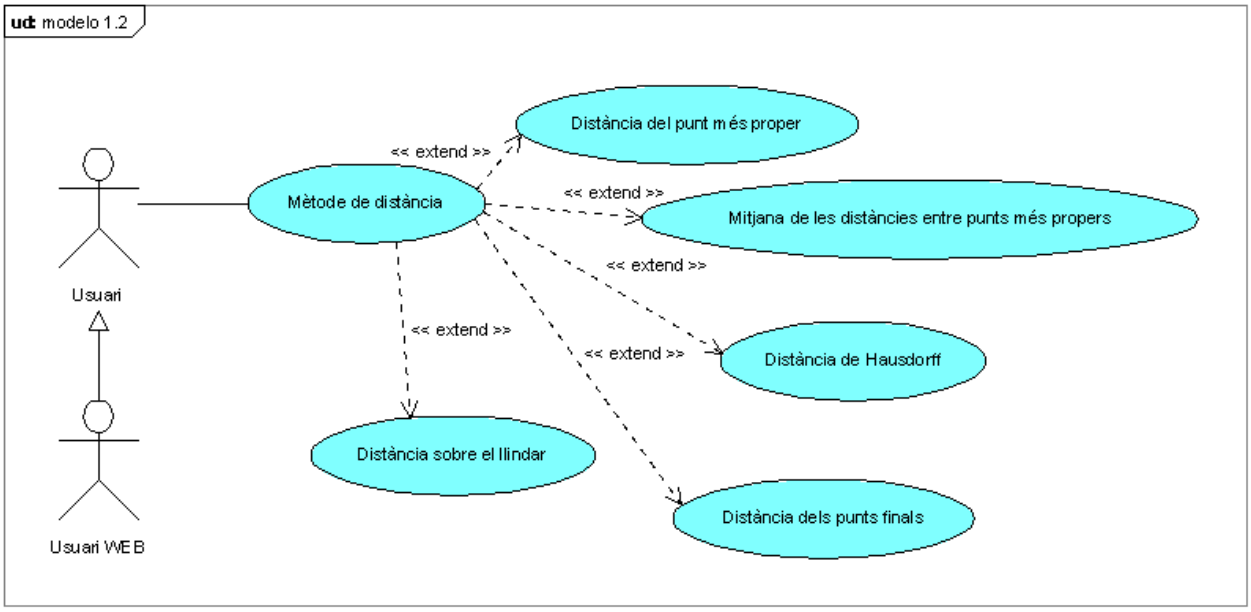


Diagrama de casos d'ús 2. *Funcionalitat 'mètodes de distància'*

Hem separat aquests dos diagrames del diagrama general de casos d'ús per a que es pogués fer una millor interpretació. Com es pot veure en ambdós diagrames, la funcionalitat general compta amb una sèrie de funcionalitats que n'estenen. Aquestes són les que es triarà a l'aplicació. Les funcionalitats 'clustering de fibres' i 'mètode de distància' serveixen únicament per a englobar-les.

A continuació es presenten les fitxes de cas d'ús corresponents a cadascuna de les funcionalitats:

CAS D'ÚS	Clustering de fibres
Descripció	Dóna opció a elegir el tipus de clustering que es vol fer
Actor	Usuari
Precondició	Es té un tracking de fibres fet
Flux principal	1.- Es mostren els diferents algorismes de clustering
	2.- L'usuari escull el mètode que més li convé
Fluxos alternatius	No n'hi ha
Postcondició	El mètode de clustering ha estat escollit

CAS D'ÚS	Mètode de distància
Descripció	Dóna opció a elegir el tipus de distància per a fer el clustering
Actor	Usuari
Precondició	Es té un mètode de clustering seleccionat
Flux principal	1.- Es mostren els diferents mètodes de distància
	2.- L'usuari escull el mètode que més li convé
Fluxos alternatius	No n'hi ha
Postcondició	El mètode de distància ha estat escollit

CAS D'ÚS	Clustering exclusiu
Descripció	Genera un agrupament de les fibres
Actor	Usuari
Precondició	Es té un tracking de fibres fet
Flux principal	1.- Posicionar k centroides
	Repetir mentre la posició d'algun centroide varii
	2.- Assignar cada fibra al grup que tingui el centroide més proper
	3.- Recalcular les posicions dels k centroides
	4.- Es mostra el resultat per pantalla
Fluxos alternatius	No n'hi ha
Postcondició	S'ha fet l'agrupament de fibres

CAS D'ÚS	Clustering de solapament
Descripció	Genera un agrupament de les fibres
Actor	Usuari
Precondició	Es té un tracking de fibres fet
Flux principal	1.- S'escull un nombre de clústers i un llindar
	Repetir mentre el canvi de coeficients entre dues iteracions sigui major que el llindar
	2.- Assignar de forma aleatòria, un coeficient de pertinença als clústers per a cada punt
	3.- Càlcul del centroide per a cada clúster
	4.- Per a cada fibra, calcular el seu coeficient de pertànyer al clúster
	5.- Es mostra el resultat per pantalla
Fluxos alternatius	No n'hi ha
Postcondició	S'ha fet l'agrupament de fibres

CAS D'ÚS	Clustering jeràrquic
Descripció	Genera un agrupament de les fibres
Actor	Usuari
Precondició	Es té un tracking de fibres fet
Flux principal	1.- S'escull N
	2.- Cada dada forma un clúster
	Repetir mentre hi hagi menys d'N grups
	3.- Es localitzen els grups més pròxims en relació a la distància entre centroides.
	4.- S'ajunten els dos grups localitzats en el punt anterior
	5.- Es calcula el nou centroide
	6.- Es mostra el resultat per pantalla
Fluxos alternatius	No n'hi ha
Postcondició	S'ha fet l'agrupament de fibres

CAS D'ÚS	Distància del punt més proper
Descripció	Calcula la distància entre els punts més propers de dues fibres
Actor	Usuari
Precondició	Es té un tracking de fibres fet
Flux principal	Per tants punts com tingui la primera fibra
	Per tants punts com tingui la segona fibra
	1.- Càlcul de la distància entre Ai i Bj
	2.- S'agafa la distància mínima entre les calculades
Fluxos alternatius	No n'hi ha
Postcondició	S'obté la distància

CAS D'ÚS	Mitjana de les distàncies entre punts més propers
Descripció	Calcula la mitjana de les distàncies entre punts més propers de dues fibres
Actor	Usuari
Precondició	Es té un tracking de fibres fet
Flux principal	Per tants punts com tingui la primera fibra
	Per tants punts com tingui la segona fibra
	1.- Càlcul de la distància entre Ai i Bj
	2.- Es guarda la distància al punt més proper
	3.- S'agafa com a distància la mitjana de totes les distàncies mínimes
Fluxos alternatius	No n'hi ha
Postcondició	S'obté la distància

CAS D'ÚS	Distància de Hausdorff
Descripció	Calcula la distància de Hausdorff entre dues fibres
Actor	Usuari
Precondició	Es té un tracking de fibres fet
Flux principal	Mentre hi hagi punts per tractar a la primera fibra i cap distància superi el llindar
	Per tants punts com tingui la segona fibra
	1.- Càlcul de la distància entre Ai i Bj
	2.- Es guarda la distància al punt més proper
	3.- Es compara la distància amb un llindar
	4.- Les fibres es consideren del mateix grup si cap distància supera el llindar.
Fluxos alternatius	No n'hi ha
Postcondició	S'obté la distància

CAS D'ÚS	Distància dels punts finals
Descripció	Calcula la distància dels punts finals entre dues fibres
Actor	Usuari
Precondició	Es té un tracking de fibres fet
Flux principal	1.- Càlcul de la distància entre el punt inicial de la primera fibra i el punt inicial de la segona fibra
	2.- Càlcul de la distància entre el punt final de la primera fibra i el punt final de la segona fibra.
	3.- La distància és la suma de les dues distàncies calculades en els punts 1 i 2.
Fluxos alternatius	No n'hi ha
Postcondició	S'obté la distància

CAS D'ÚS	Distància sobre el llindar
Descripció	Calcula la distància a partir d'un llindar
Actor	Usuari
Precondició	Es té un tracking de fibres fet
Flux principal	Determinació d'un llindar
	Per tants punts com tingui la primera fibra
	Per tants punts com tingui la segona fibra
	1.- Calcular la distància entre els dos punts
	2.- Si la distància és més petita que la distància mínima la distància mínima serà la distància actual
	3.- distancia 1 = distancia 1 + max(distancia mínima - llindar, 0)
	4.- distancia 2 = distancia 2 + max((distancia mínima - llindar) / (abs(distancia mínima - llindar)), 0)
	5.- S'agafa com a distància la divisió de la distancia1 per la distancia2
Fluxos alternatius	No n'hi ha
Postcondició	S'obté la distància

Diagrama de classes general

En el *diagrama de classes 1* es mostren les classes que conté l'aplicació. Les classes pintades de blau són les que s'incorporen en aquest projecte i la resta són les classes de les que ja disposava abans l'aplicació.

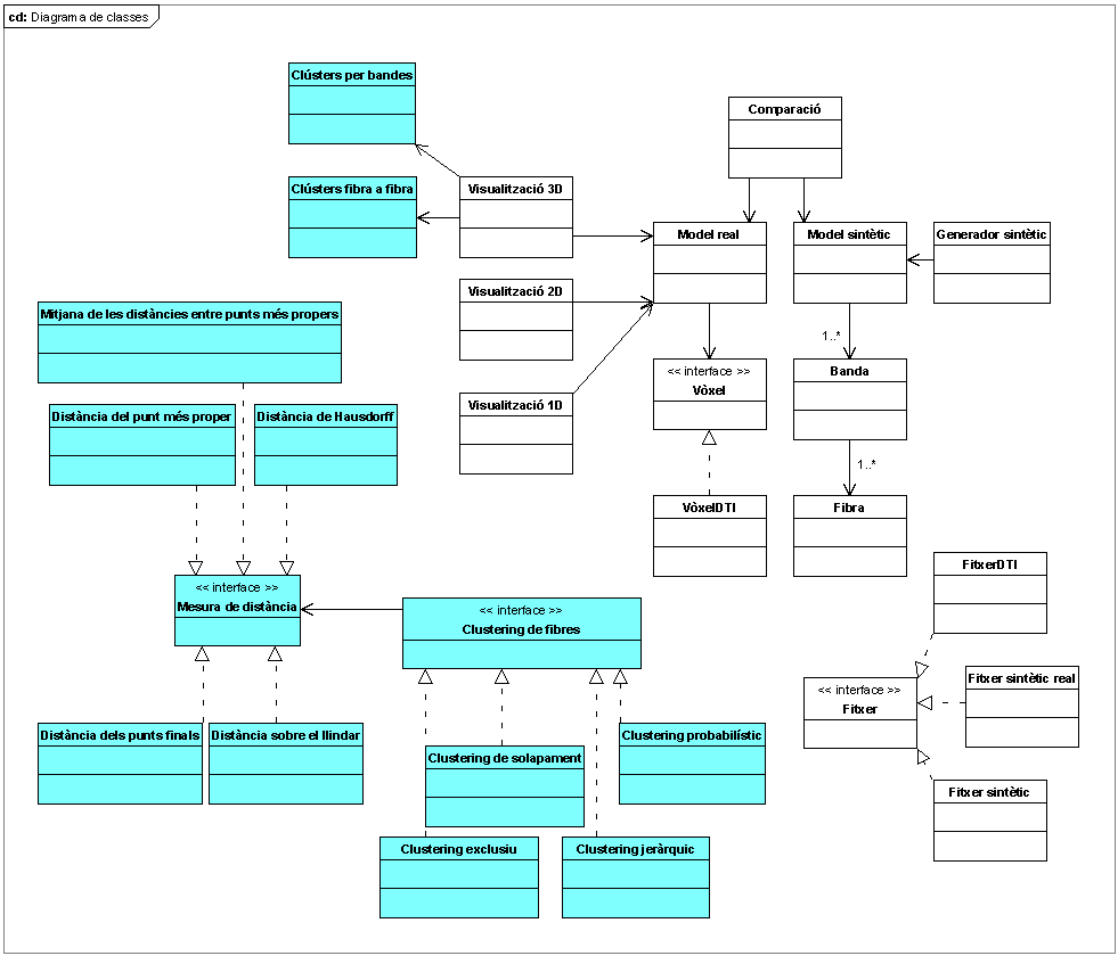


Diagrama de classes 1. *Aplicació*

A continuació es fa una breu descripció de cadascuna de les noves classes que s'incorporen:

CLASSE	EXPLICACIÓ
<< interface >> **Mesura de distància**	Classe abstracta de la que hereden els diferents mètodes de distància.

Distància del punt més proper	Classe que implementa el mètode trobarDistancia() buscant els punts més propers.
Mitjana de les distàncies entre punts més propers	Classe que implementa el mètode trobarDistancia() buscant la mitjana de les distàncies més petites entre punts.
Distància de Hausdorff	Classe que implementa el mètode trobarDistancia() buscant la major distància entre les mínimes.
Distància dels punts finals	Classe que implementa el mètode trobarDistància() fent la suma de les distàncies dels punts extrems.
Distància sobre el llindar	Classe que implementa el mètode trobarDistancia() utilitzant un llindar.
<< interface >> **Clustering de fibres**	Classe abstracta de la que hereden els diferents algorismes de clustering.
Clustering exclusiu	Classe que implementa el mètode trobarFibres() a partir de l'algorisme k-means.
Clustering de solapament	Classe que implementa el mètode trobarFibres() a partir de l'algorisme fuzzy c-means.

Clustering jeràrquic	Classe que implementa el mètode trobarFibres() a partir de l'algorisme jeràrquic aglomeratiu.
Clustering probabilístic	Classe que implementa el mètode trobarFibres() a partir de l'algorisme EM.
Clústers fibra a fibra	Classe que implementa el mètode construirEscena() a partir de crear una línia per cada fibra i assignant-li un color.
Clústers per bandes	Classe que implementa el mètode construirEscena() creant per a cada clúster una superfície i donant-li sensació de profunditat.

4.2.- Implementació dels mètodes de distància

En aquest apartat mostrarem com s'han implementat els mètodes de distància entre fibres i valorarem els resultats. Per a fer la valoració de resultats hem utilitzat un mateix model de dues fibres que presentem en l'apartat següent.

4.2.1.- Model utilitzat per realitzar les proves

Un cop finalitzada la implementació i amb la finalitat de provar els mètodes de distància hem utilitzat un mateix model.
El model consta de dues úniques fibres per a trobar les distàncies corresponents.

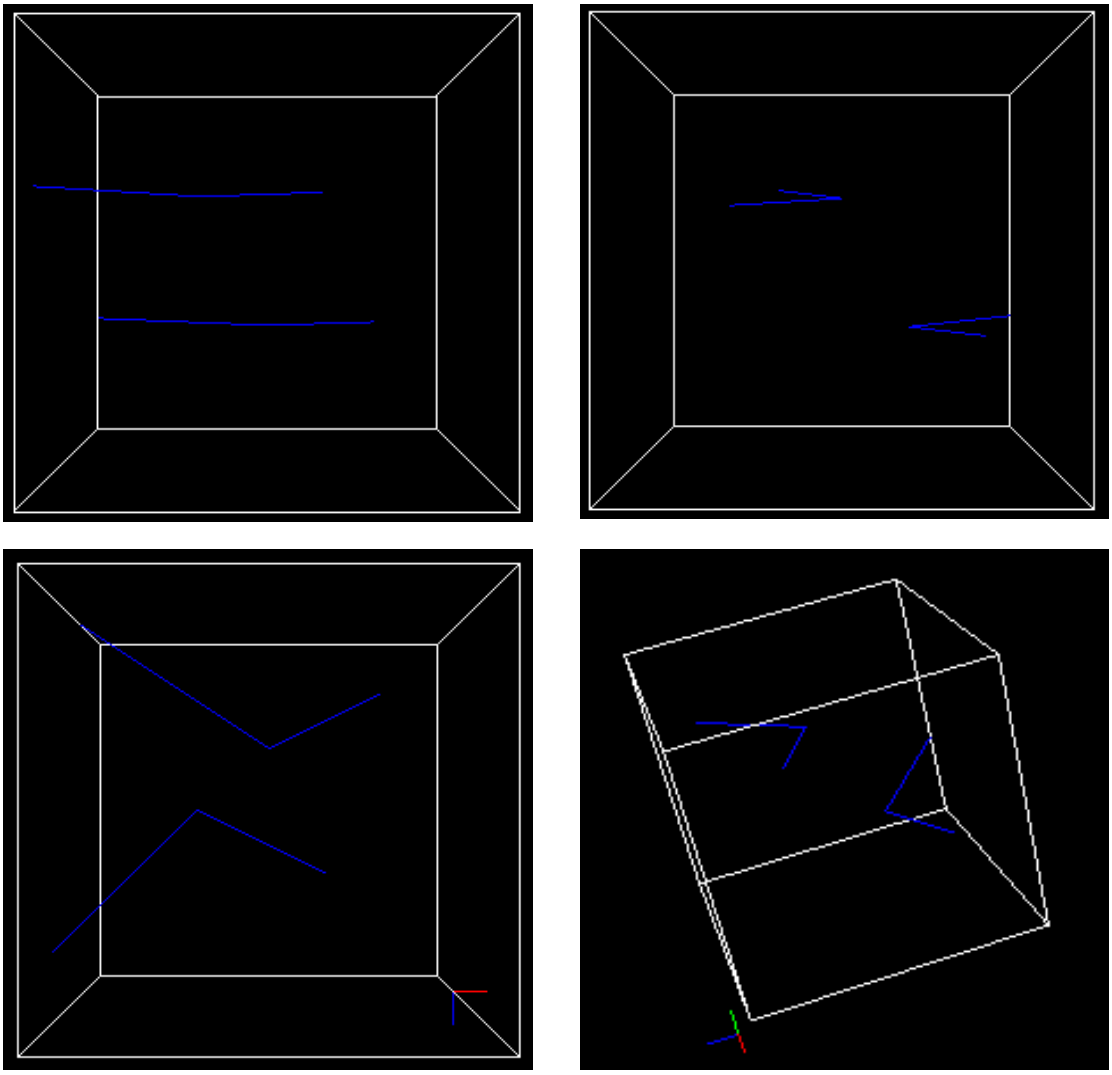


***Figura 28**. Model sintètic que conté dues fibres.*

La figura 28 mostra les diferents vistes d'un model sintètic creat per al testeig dels diferents mètodes de distància. Les vistes de la figura corresponen als plans següents (d'esquerra a dreta i de dalt a baix): pla XY, pla YZ, pla XZ i vista en 3 dimensions.

Podem veure com els punts de les fibres es mouen sobre el pla XZ, mantenint-se la component y estàtica. Tot i això, s'ha de tenir en compte que les fibres estan en diferents punts de l'eix de les y.

4.2.2.- Distància dels punts més propers

Aquest mètode comença buscant la fibra més curta en cas que no siguin de la mateixa mida. Si són de la mateixa mida serà indiferent. Després busca els dos punts més propers de les dues fibres que estem analitzant i retorna la distància entre aquests. *(Veure l'apartat 2.6.1)*

Podem veure més clara la idea amb el *diagrama de seqüència 1*:

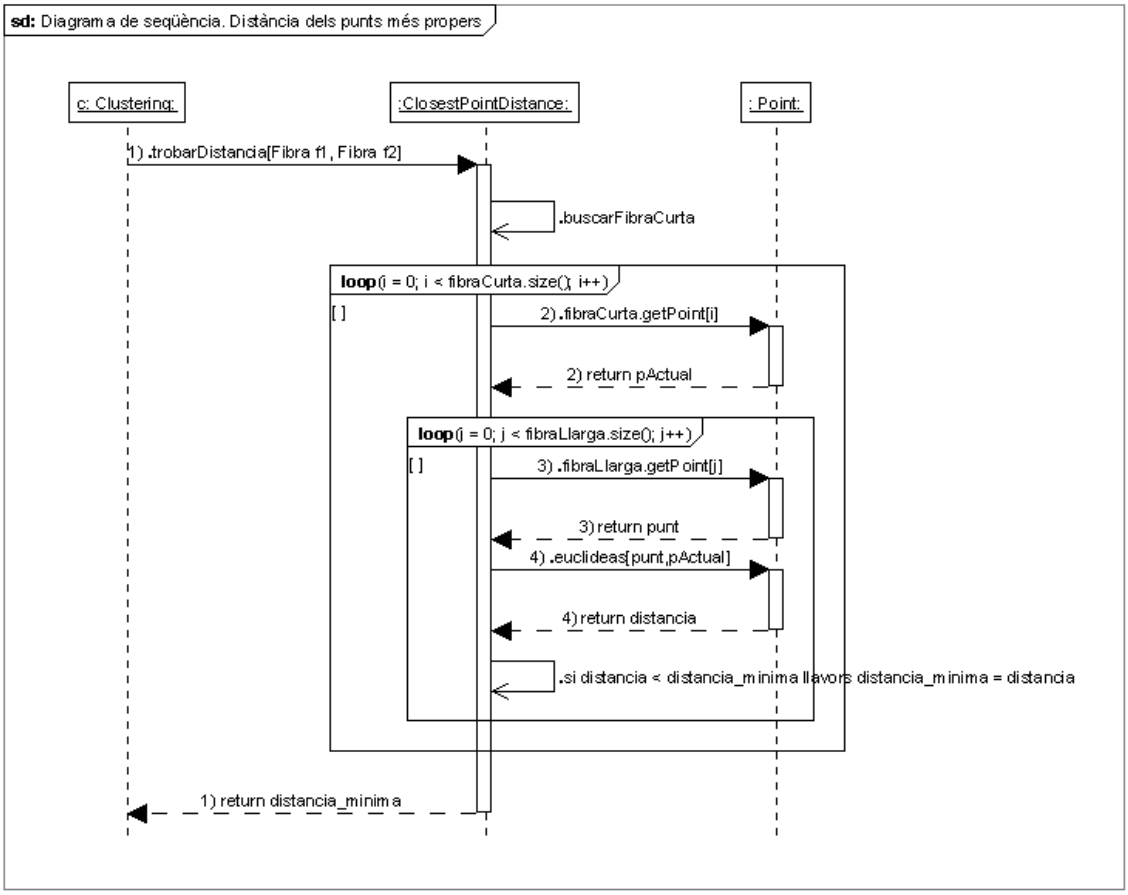


Diagrama de seqüència 1. *Mètode de distància dels punts més propers.*

A continuació mostrem una captura de pantalla resultat d'aplicar aquest mètode sobre dues fibres de dos clústers diferents.

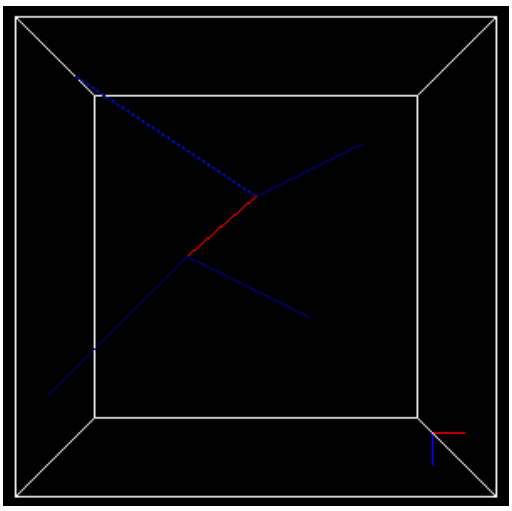


***Figura 29**. Distància entre els punts més propers. Pla XZ.*

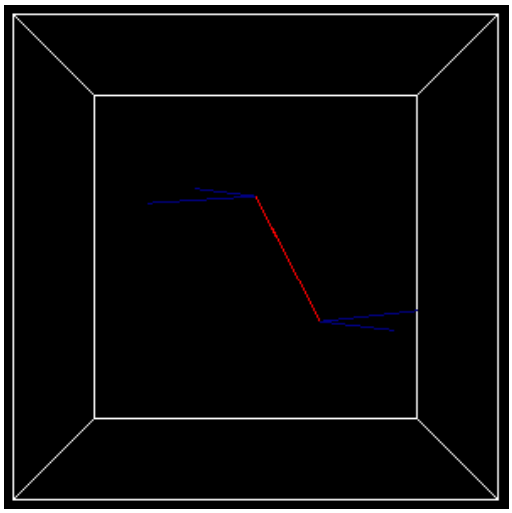


***Figura 30**. Distància entre els punts més propers. Pla YZ.*

En la *figura 29* veiem una vista del pla XZ del model que hem explicat en l'apartat anterior i en la *figura 30* una vista del pla YZ. Amb aquestes dues vistes ens fem una idea del traçat entre els punts més propers.

Hem fet que l'aplicació ens pintés la distància mínima entre les fibres de color vermell.

El valor d'aquesta distància és de 2.44 i es pot apreciar a simple vista que és la menor entre els punts de les dues fibres.

4.2.3.- Mitjana de distàncies dels punts més propers

El mètode de la mitjana de les distàncies entre punts més propers comença buscant la fibra més curta. Un cop se sap quina és la fibra més curta i quina la més llarga, es busca per cada punt de la fibra més curta, el punt més proper de la fibra més llarga. Un cop s'han tractat tots els punts de la fibra curta, es tenen tantes distàncies com punts tingui la fibra més curta. I amb aquests valors es fa la mitjana i és el valor que retorna i el que serà utilitzat pel mètode de clustering.
(Veure l'apartat 2.6.2)

Veiem a continuació el *diagrama de seqüència 2* que explica el funcionament del mètode:

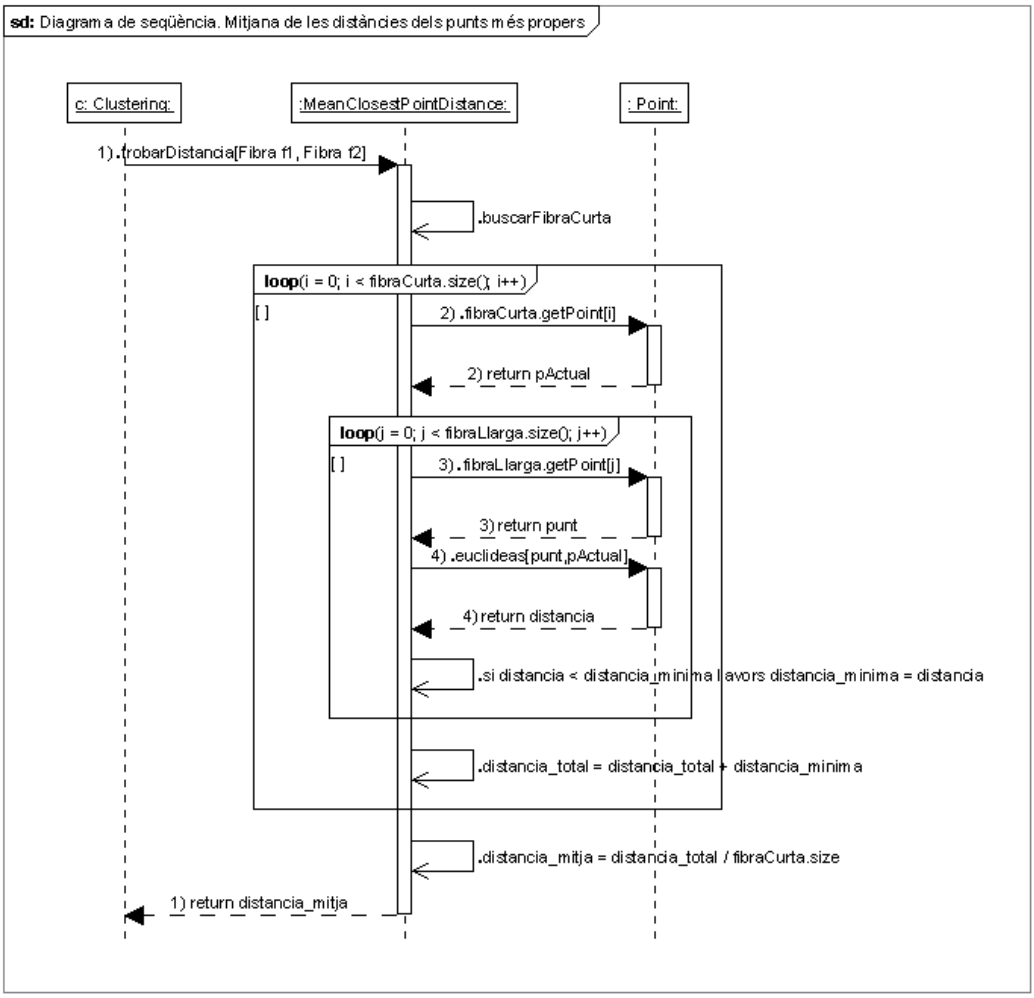


Diagrama de seqüència 2. *Mètode de la mitjana de les distàncies dels punts més propers.*

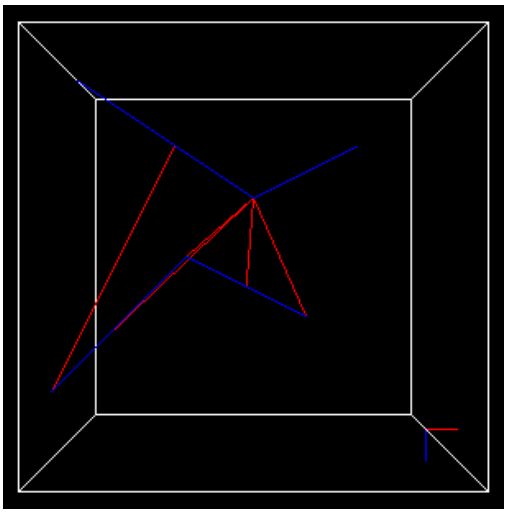


Figura 31. *Mitjana de les distàncies dels punts més propers. Pla XZ.*

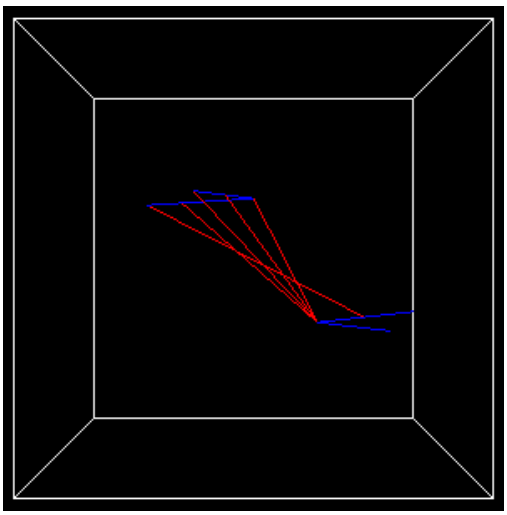


Figura 32. *Mitjana de les distàncies dels punts més propers. Pla YZ.*

A la *figura 31* podem veure una vista del pla XZ i a la *figura 32* una de la vista YZ de la mateixa escena de l'exemple anterior. Ara veiem totes les distàncies mínimes (marcades en vermell) de cada punt de la fibra més curta a la fibra més llarga.

La *taula 1* mostra els resultats numèrics del mètode de la mitjana de les distàncies entre punts més propers. Ens informa quin és el punt més pròxim de la fibra curta a la llarga i quin és el valor d'aquesta distància.
També quina és la suma de les distàncies i la mitjana d'aquestes, que serà el valor que retornarà el mètode.

Punt de la fibra curta	Distància mínima	Punt de la fibra llarga
0	4,68	2
1	3,81	3
2	2,45	3
3	2,47	3
4	2,87	3
Suma de les distàncies mínimes	16,28	
Mitjana de les distàncies entre punts propers	**3,256**	

Taula 1.
Mitjana de les distàncies entre punts més propers.

4.2.4.- Distància de Hausdorff

El mètode de la distància de Hausdorff comença buscant la fibra més curta i, llavors, busca per cada punt de la fibra més curta el punt més proper de la fibra més llarga. Finalment, el que retorna és la distància més gran d'entre totes les distàncies mínimes que haguem trobat. *(Veure l'apartat 2.6.3)*

El *diagrama de seqüència 3* explica el funcionament del mètode.

A la *figura 33* i la *figura 34* veiem el resultat d'aplicar el mètode sobre les dues fibres (marcades en blau). La distància resultant està marcada en vermell i el seu valor és 4.68.

Si mirem el resultat del mètode anterior podem veure que aquesta distància coincideix amb una de les mínimes que s'havien trobat i que visualment ja s'apreciava com la més gran.

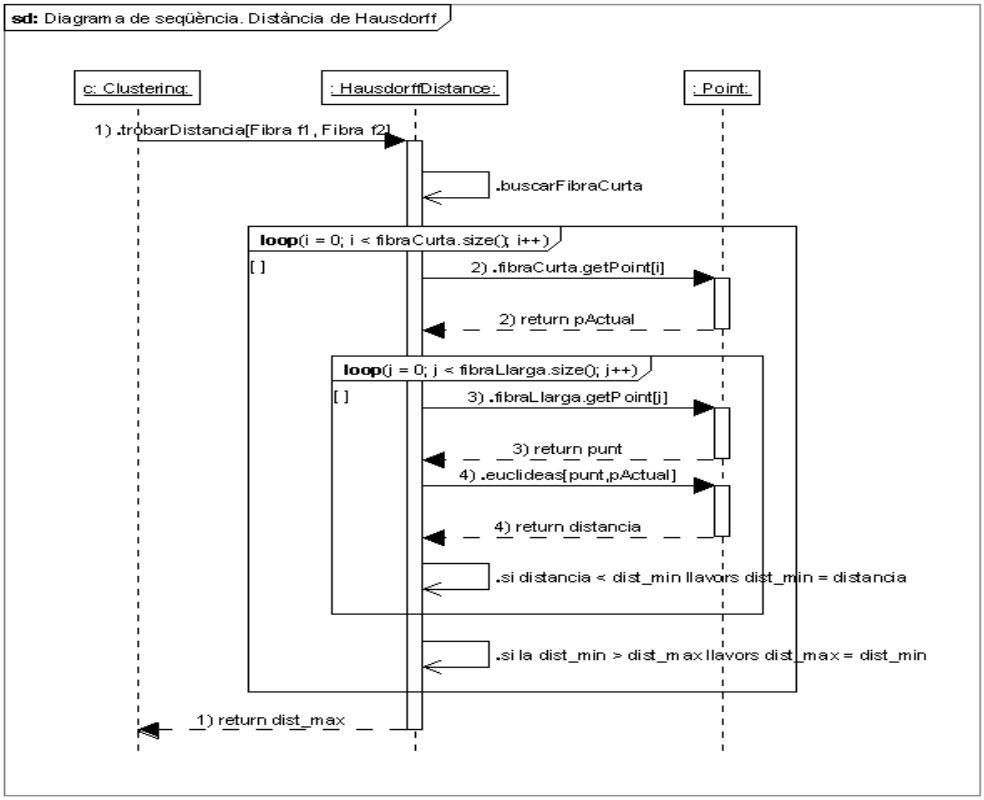


Diagrama de seqüència 3. *Mètode de la distància de Hausdorff.*

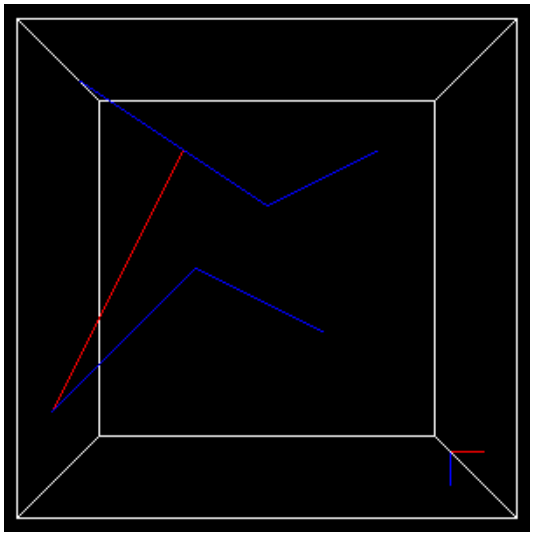


Figura 33.
Distància de Hausdorff. Pla XZ

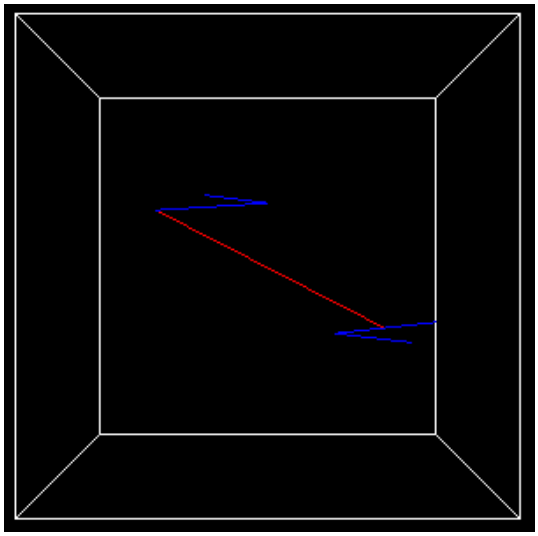


Figura 34. *Distància de Hausdorff. Pla YZ*

4.2.5.- Distància dels punts finals

El mètode de la distància dels punts finals comença obtenint 4 punts: l'inicial i el final de la primera fibra i l'inicial i el final de la segona fibra. El següent pas és calcular dues distàncies.
La primera distància és la suma de la distància dels punts inicials de les dues fibres amb la distància dels punts finals de les dues fibres.
La segona distància és la suma de la distància entre el punt inicial de la primera fibra i el punt final de la segona, amb la distància entre el punt final de la primera fibra i el punt inicial de la segona.
Llavors s'agafa la distància més petita.

Ens podem trobar amb el cas de que el punt inicial d'una fibra estigui en un extrem i el punt inicial de l'altra fibra estigui en l'altre extrem. Aquest és el sentit de calcular les dues distàncies. *(Veure l'apartat 2.6.4)*

Veiem el *diagrama de seqüència 4* corresponent a aquest mètode de distància:

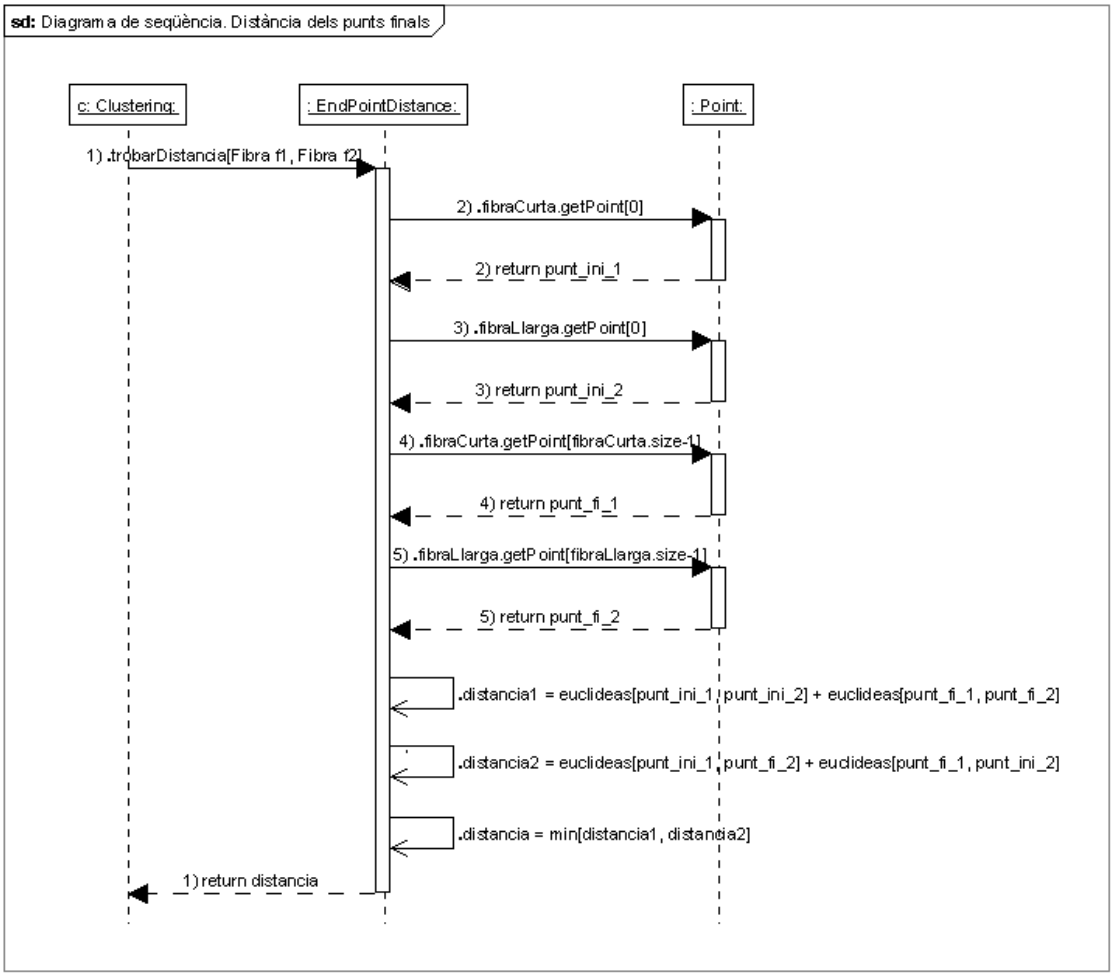


Diagrama de seqüència 4. *Mètode de la distància dels punts finals.*

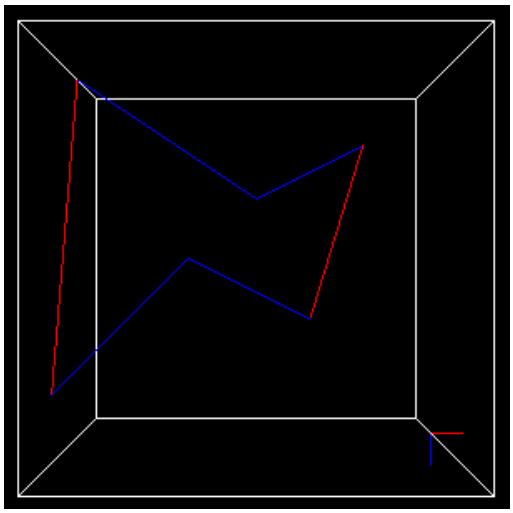


Figura 35.
Distància dels punts finals. Pla XZ.

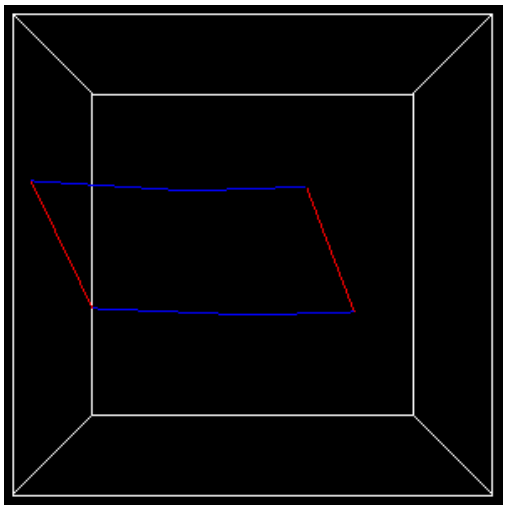


Figura 36.
Distància dels punts finals. Pla XY.

El resultat d'aquest mètode el podem veure a *les figures 35 i 36*, on s'han pintat en vermell les dues distàncies entre els punts extrems de la fibra. Quan cridem aquest mètode per a fer el clustering, el valor que retornarà serà la suma d'aquestes dues distàncies.
En aquest exemple la suma és de 8.96.

4.2.6.- Distància sobre el llindar

(Veure l'apartat 2.6.5)

Al *diagrama de seqüència 5* es mostra el funcionament del mètode.

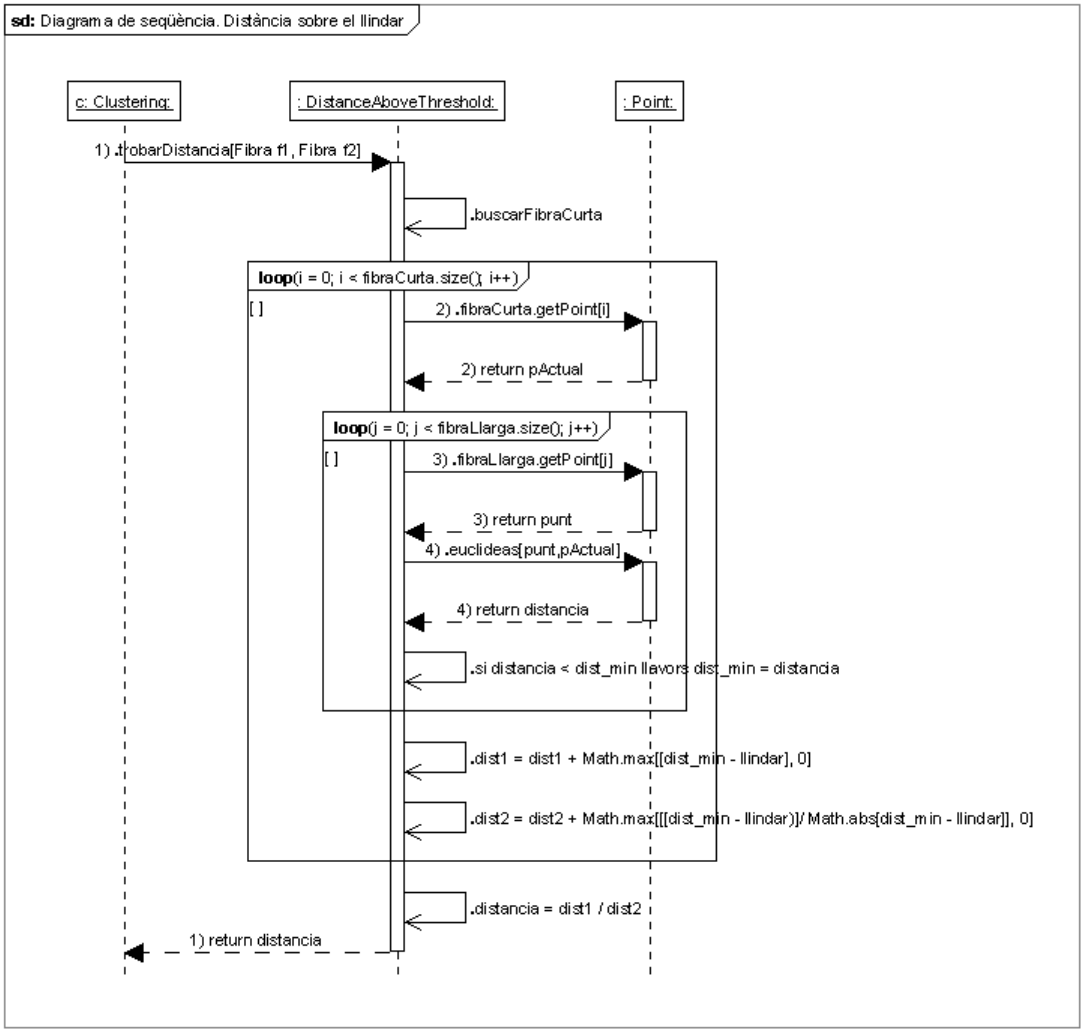


Diagrama de seqüència 5. *Mètode de la distància sobre el llindar.*

4.3.- Implementació dels mètodes de clustering

En aquesta part s'explica la implementació dels 4 algorismes de clustering explicats teòricament a l'apartat 2.5. L'explicació de cada algorisme d'agrupament consta de la part aplicada sobre models sintètics i la part aplicada sobre models reals.

L'aplicació sobre models sintètics té com a funció validar la correctesa de l'algorisme, ja que, des d'un principi sabem quin ha de ser el resultat i podem veure si coincideix.

Un cop tenim la seguretat de que hem fet una bona implementació de l'algorisme, ja podem procedir a veure els resultats sobre models reals.

4.3.1.- Models utilitzats per a realitzar les proves

A l'igual que en l'apartat de les distàncies, en aquest apartat d'agrupament de les fibres del model, hem utilitzat un conjunt de models de proves per a fer les validacions dels mètodes.

Partint de la idea de l'Extreme Programming, s'han anat testejant els mètodes a mida que s'implementaven. Un cop obteníem resultats, hem anat refinant el mètode fins a arribar a la solució òptima.

A continuació presentem els models utilitzats per als testejos dividits en models sintètics i en models reals.

4.3.1.1.- Models sintètics

Hem creat diferents complexitats de models sintètics, per fer proves començant amb els més senzills i acabant amb els més complexos. D'aquesta manera podrem estar més segurs de que la implementació de l'algorisme de clustering sigui correcta.

Model ms_01

El primer model sintètic creat el podem veure a la *figura 37*. Hi ha quatre imatge que corresponen a les vistes dels plans XY, YZ, XZ i 3D ordenades d'esquerra a dreta i de dalt a baix.

Aquest model disposa de 29 fibres repartides en 3 grups diferents, el clúster vermell, el cian i el lila.

Els grups estan separats; no hi ha cap encreuament de les fibres en cap cas.

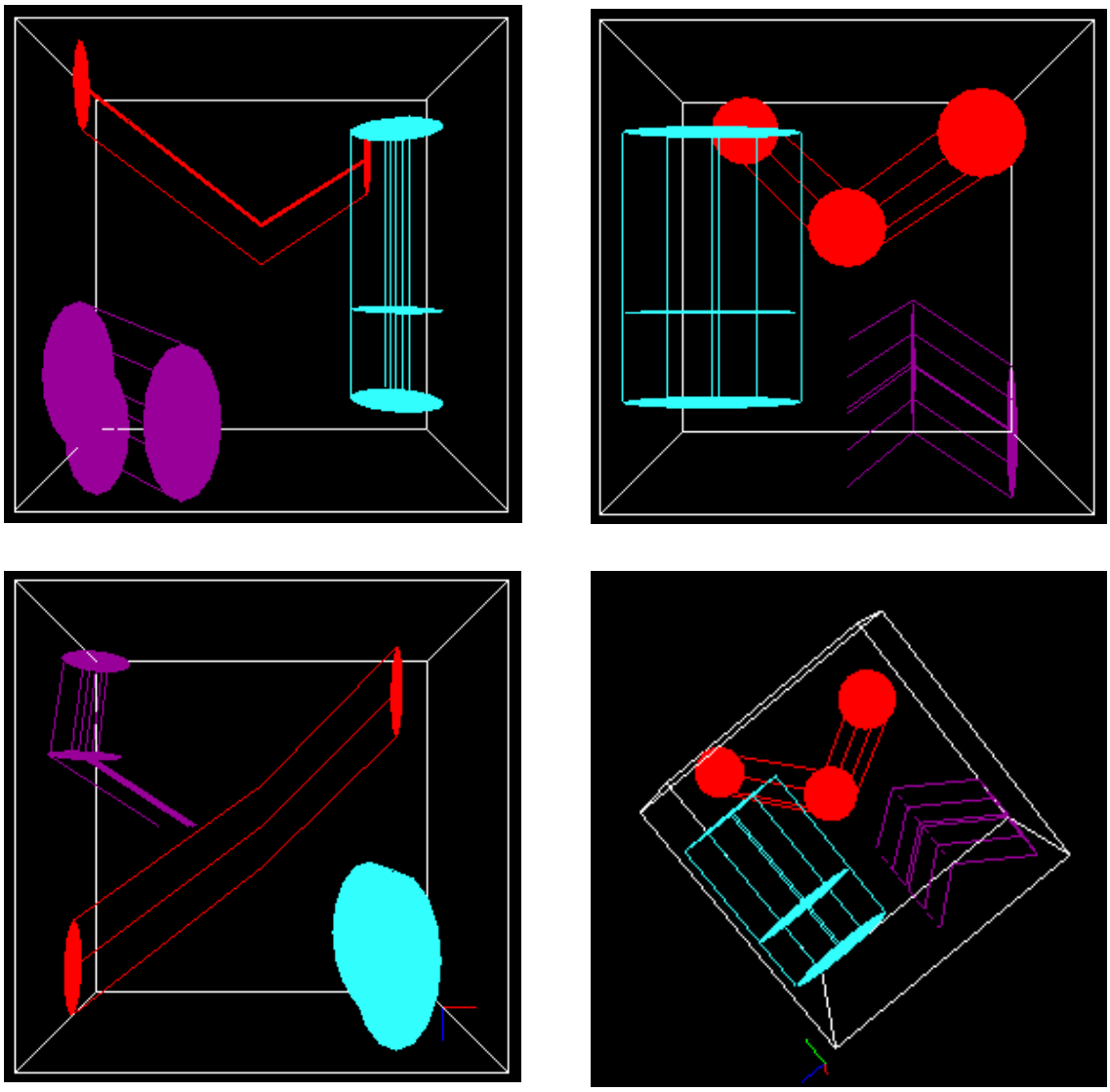


Figura 37. *Vistes del model sintètic ms_01.*

Model ms_02

El model sintètic ms_02 el podem veure a la *figura 38*. Hi ha quatre imatge que corresponen a les vistes dels plans XY, YZ, XZ i 3D ordenades d'esquerra a dreta i de dalt a baix.

Aquest model és el que menys fibres conté de tots els que hem fet servir. Conté 14 fibres repartides en 2 grups diferents, el clúster verd i el lila.

Els grups s'encreuen en la part centrals d'aquests. Aquesta és la característica destacable d'aquest model, ja que, ens serà útil per a evitar possibles errors.

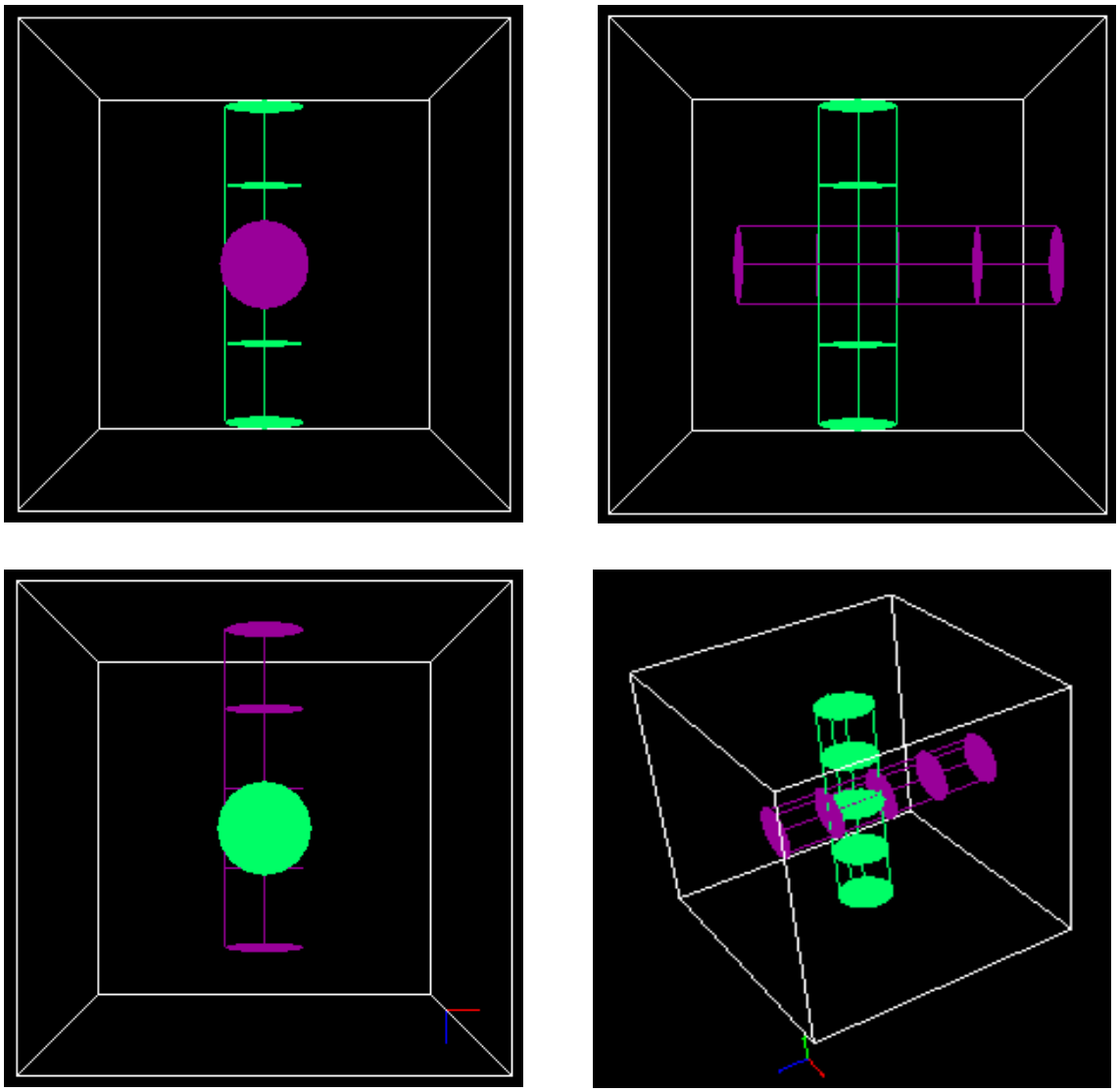


Figura 38. *Vistes del model sintètic ms_02.*

Model ms_03

El model sintètic ms_03 el podem veure a la *figura 39* on hi ha les 4 vistes al igual que en els models anteriors.

Aquest model té la mateixa estructura que l'anterior, dos clústers que es creuen a la meitat. L'única diferència és que aquest té 200 fibres més, en comptes de 14, 214.

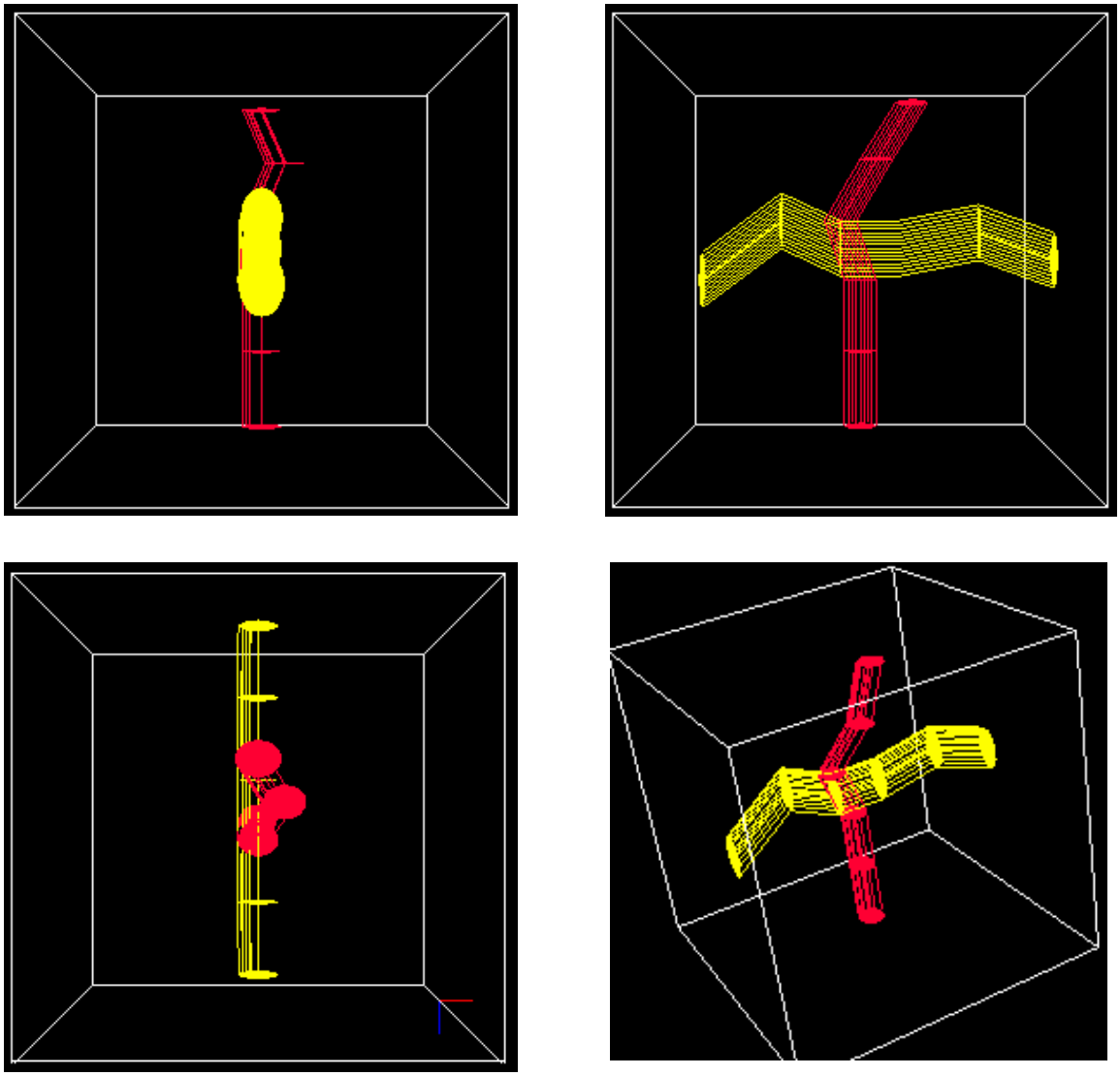


Figura 39. *Vistes del model sintètic ms_03.*

Model ms_04

El model sintètic ms_04 el podem veure a la *figura 40* on hi ha les 4 vistes al igual que en els models anteriors.

Aquest model disposa de 5 clústers de diferents mides, el blau, el groc, el vermell, el verd i el lila. Estan disposats per tot l'escenari.

El nombre de fibres ja és més elevat. Exactament conté 780 fibres.

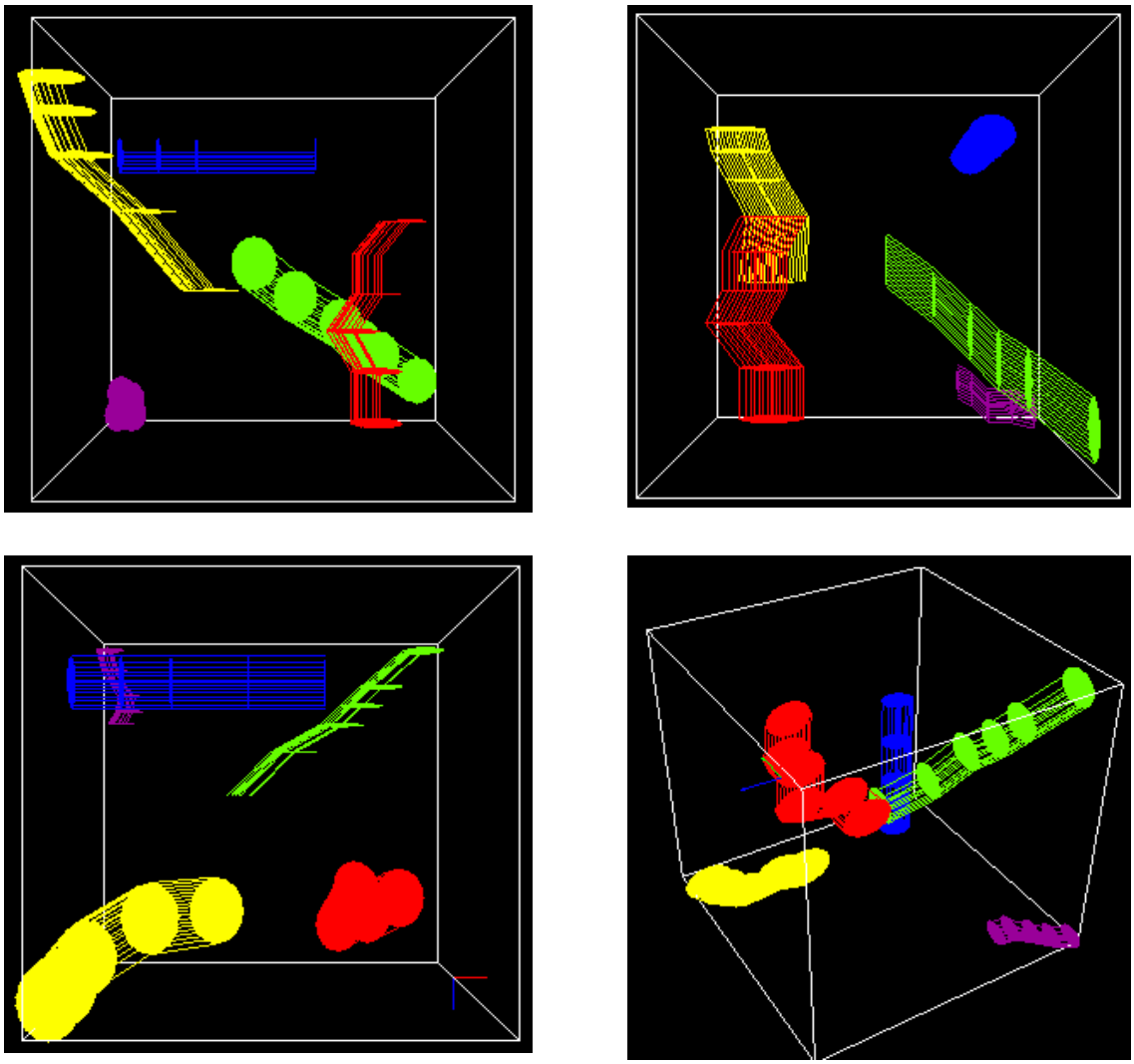


Figura 40. *Vistes del model sintètic ms_04*

Altres models sintètics

A part dels 4 models exposats a dalt, s'han fet proves esporàdiques amb models amb estructures semblants però amb un nombre de fibres que supera les 1000 unitats, per comprovar temps d'execució.

4.3.1.2.- Models reals

Disposem de dos models reals per a provar els resultats.

Voluntari 1

El model pertanyent al voluntari 1 té 1160 fibres i a la *figura 41* es mostren les diferents vistes (pla xy, pla yz, pla xz i vista 3D).

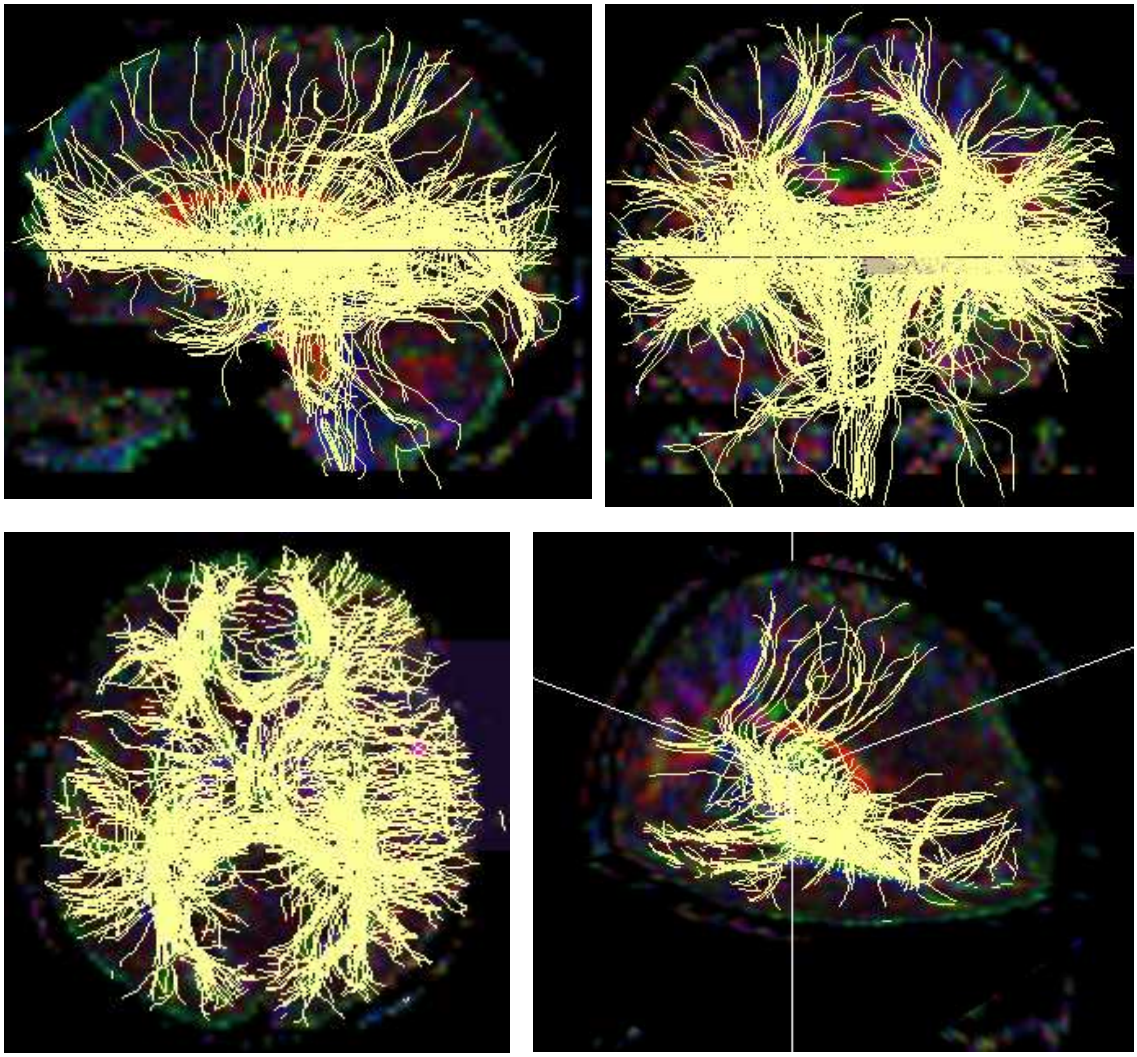


Figura 41. *Vistes del model real del voluntari 1*

Voluntari 2

El model pertanyent al voluntari 2 té 1311, 150 més que les del voluntari 1. Si mirem la *figura 42*, veure'm però, que la estructura és la mateixa.

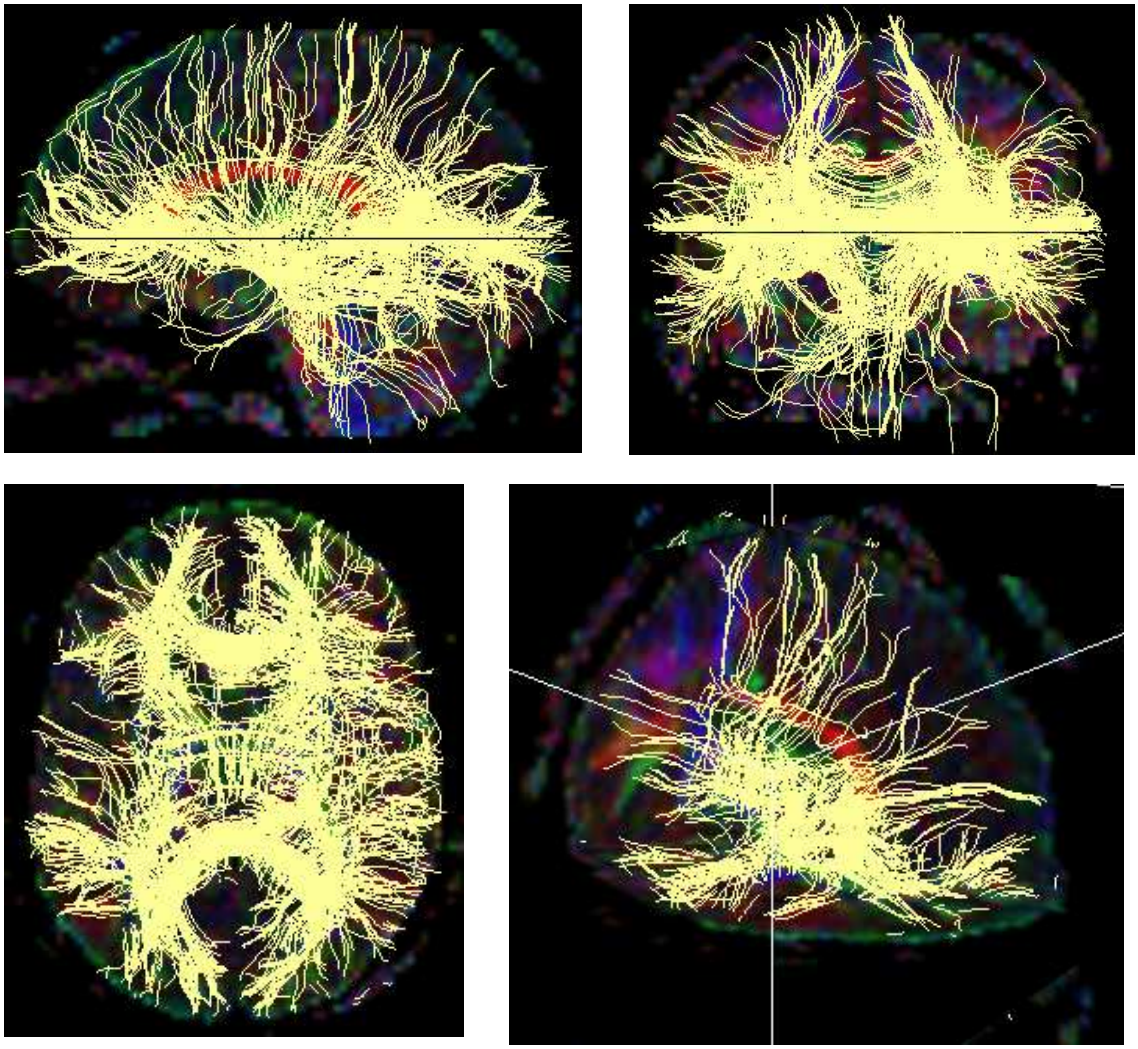


Figura 42. *Vistes del model real del voluntari 2.*

4.3.2.- L'algorisme de clustering jeràrquic

4.3.2.1.- Implementació sobre models sintètics

Per a fer la implementació del clustering jeràrquic necessitem un llindar. Aquest llindar farà referència a la màxima distància que hi pot haver entre dos grups per a que es considerin del mateix grup o de grups diferents. El llindar el pot triar l'usuari, però també podem optar per establir un llindar estàndard, ja que, tot i tractar diferents models reals, tots segueixen la mateixa estructura. En el cas dels models sintètics el llindar que establirem anirà variant perquè farem proves sobre diferents graus de complexitat, la qual cosa significarà que les fibres d'una mateixa banda estaran més o menys separades.

4.3.2.1.1.- 1a implementació

Posem cada fibra en un clúster, per tant, tenim tants clústers com fibres disposi el model. Calculem totes les distàncies entre tots els clústers i anem actualitzant la distància mínima i els clústers que la formen, de manera que quan estan totes calculades, ens quedem amb la més petita. A continuació fusionem els dos clústers que haguem trobat sempre i quan aquesta distància sigui menor que el llindar.
La fusió implicarà afegir totes les fibres del segon clúster al primer i esborrar el segon clúster (o viceversa).
El procés s'anirà repetint, sempre amb un grup menys que la vegada anterior, mentre la menor distància entre grups no superi el llindar.

```
Inicialització de variables

PER totes les fibres del model FER
      Crear clúster c
      Fibra f = agafar fibra
      Afegir fibra f al clúster c
      Afegir clúster c al vector de clústers
FPER
MENTRE continuar Fer
      PER tots els clústers menys l'últim FER
            PER tots els clústers menys el primer FER
                  distancia = trobarDistancia(fibra mitjana del
                        clúster 1, fibra mitjana del clúster 2)
                  SI la distància < distància mínima LLAVORS
                        Distancia mínima = distància
                  FSI
            FPER
      FPER
      SI la distància mínima < llindar LLAVORS
            Fusionar els dos clústers que feien la distància mínima
      ALTRAMENT
            No continuem iterant (continuar = fals)
      FSI
FMENTRE
```

PSEUDOCODI 1. *Clustering jeràrquic. 1a implementació.*

A continuació podem veure el *diagrama de seqüència 6* que ens mostra aquesta primera implementació i els resultats obtinguts.

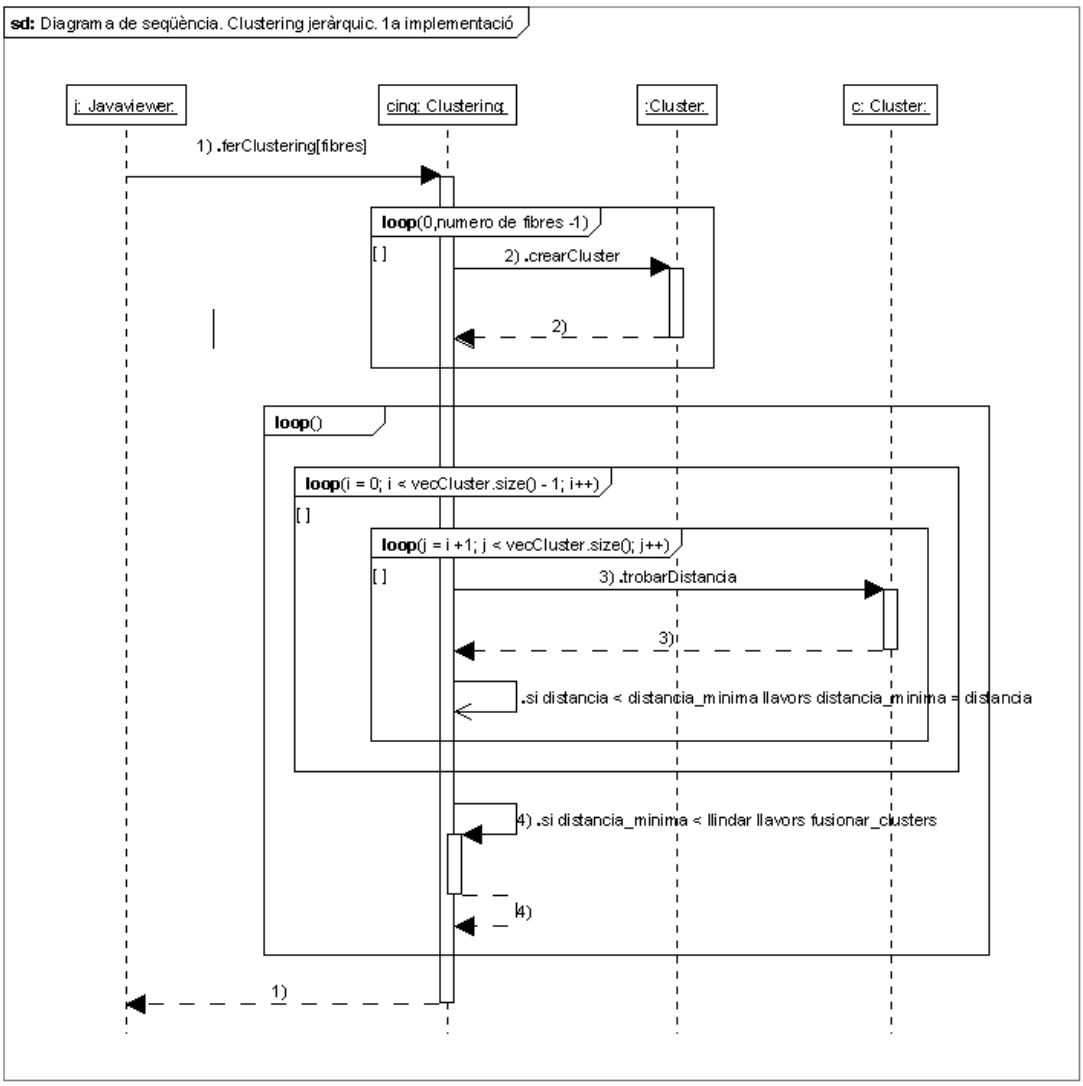


Diagrama de seqüència 6. *Primera implementació de l'algorisme jeràrquic.*

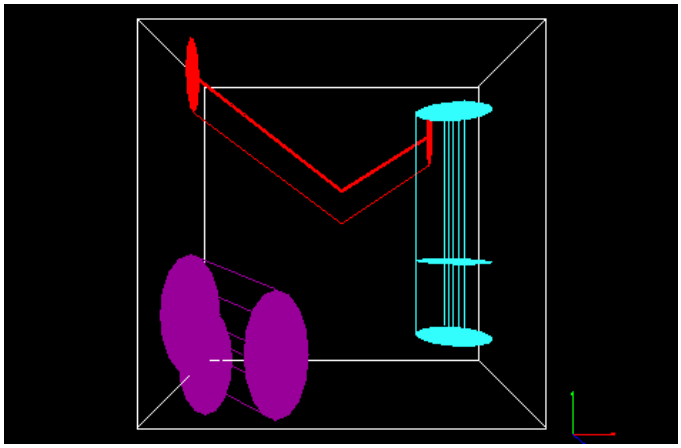


Figura 43. *MODEL ms_01.*

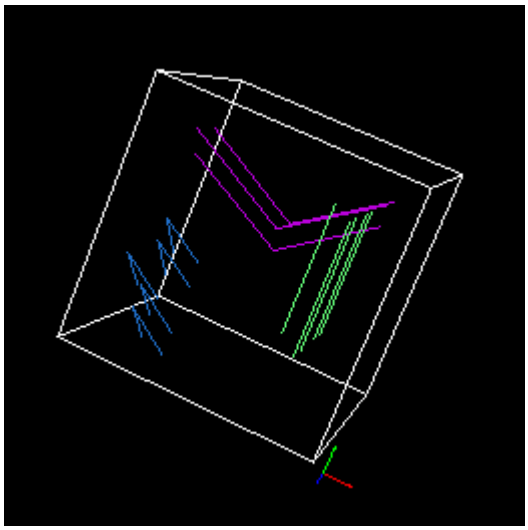


Figura 44. *Resultat d'aplicar l'algorisme de clustering jeràrquic sobre el model de la figura 43.*

Podem veure com en la figura X com l'algorisme aplicat sobre el model de la figura X s'ha comportat tal i com s'esperava, ja que, ha agrupat les fibres d'entrada en 3 grups diferents.

NOTA: L'algorisme de clustering diferencia els diferents clústers trobats en diferents colors, que no tenen a veure amb els del model sintètic.

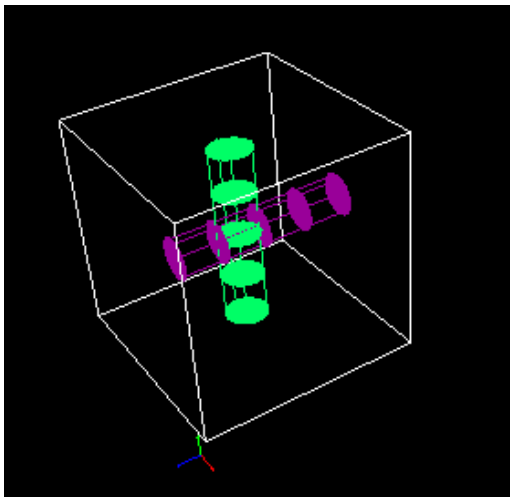


FIGURA 45. *Model ms_02.*

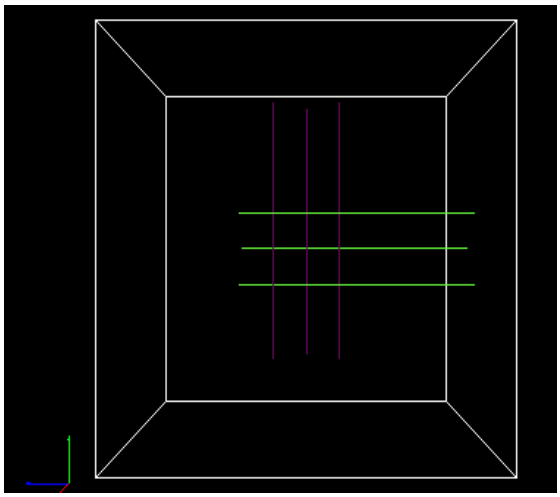


FIGURA 46. *Resultat d'aplicar l'algorisme de clustering jeràrquic sobre el model de la figura 45.*

El model de la figura X és un altre model senzill format únicament per 2 clústers i 14 fibres, però els clústers es creuen, cosa que pot portar problemes segons la distància que s'agafi. En aquest cas, hem executat l'algorisme utilitzant la mitjana dels punts més propers. Podem veure en la figura X com l'algorisme també ha donat un resultat correcte.

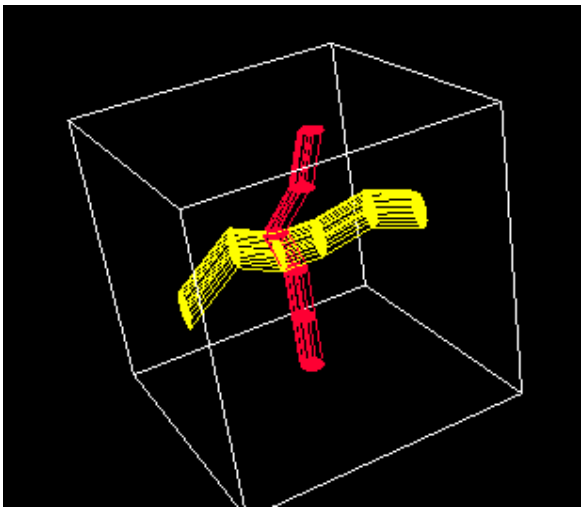


FIGURA 47. *Model ms_03.*

En aquest exemple, topem amb el gran problema d'aquest tipus de clustering. El model a avaluar ha estat el de la *figura 47* que té dos grups de fibres que es creuen com en l'exemple anterior. La diferència està en la complexitat, entenen per complexitat el nombre de fibres. Mentre el model anterior tenia un total de 14 fibres, aquest en té 214. Aquestes 200 fibres de més fan que l'algorisme passi d'instantani a trigar més de 30 minuts en trobar el resultat.
Aquest alentiment tan exagerat és degut a dos factors:

- la distància escollida. La distància entre fibres resultant de la mitjana de punts més propers entre dues fibres, també té un cost elevat, ja que ha de comparar tots els punts de la primera fibra amb els de la segona.
- el càlcul de les distàncies. Amb el model de 14 fibres, el nombre de distàncies a calcular era de 91, i amb el model de 214 fibres, aquest augmenta fins a 22.791. Cada cop que s'ajunten dos clústers es tornen a calcular totes les distàncies, per tant, es repeteixen càlculs.

El primer punt conflictiu es resol utilitzant un altre mètode de distància no tan complex, com pot ser la distància dels punts finals. El resultat serà correcte i el temps d'execució acceptable. Però, aquest tipus de mètode tot i anar bé en aquest cas, podem trobar casos en que els punts finals de dues bandes caiguin per la mateixa zona.

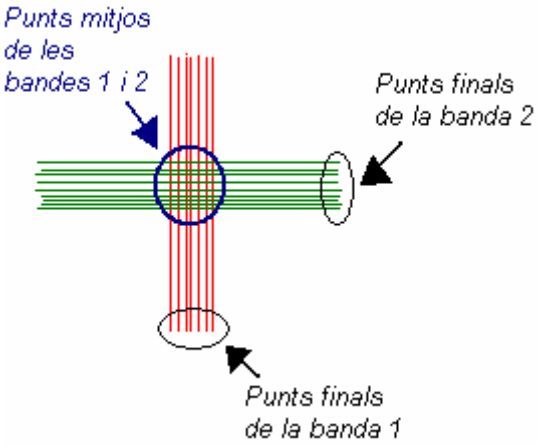


FIGURA 48.

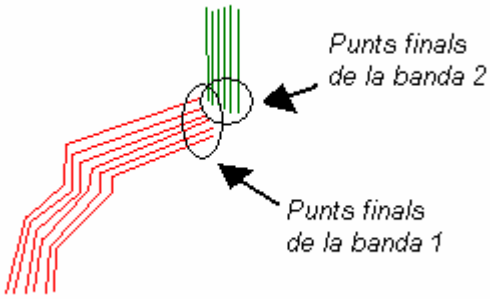


FIGURA 49.

En els dos dibuixos de les *figures 48 i 49* veiem dos models on es veu com el fet de canviar la mesura de distància no és una bona opció per a arreglar el problema. A la *figura 48* veiem el perquè sí funciona l'algorisme agafant els punts finals, ja que aquests estan clarament separats. Però, si mirem el dibuix de la *figura 49*, els punts finals de les fibres d'ambdues bandes cauen a la mateixa zona i l'algorisme les consideraria dins un mateix conjunt.

El mateix passaria si optéssim per agafar els punts centrals de les fibres. En el segon dibuix funcionaria, però en el primer no, ja que al creuar-se les bandes, els punts del mig són allà mateix.
Aquests mètodes de distància poden ser útils en models senzills i conjunts perfectament separats, però, en situacions més complicades, la mitjanes dels punts més propers, és més exacta.
Per a entendre el segon punt conflictiu, explicarem, primer, com podem saber la quantitat de distàncies que s'han de calcular.
Per a calcular les distàncies necessitem agrupar els clústers tenint en compte que:

- no influeix l'ordre. Si la distància entre el clúster 1 i el 5 és 10.4, la distància entre el clúster 5 i l'1 serà també 10.4.
- no es repeteixen els clústers. No tindria sentit calcular la distància entre un clúster i ell mateix.

Un cop fetes aquestes consideracions podem afirmar, que estem parlant de combinacions sense repetició, que podem expressar amb el símbol C_n^p. Les combinacions sense repetició d'*n* elements agafats de *p* en *p* es defineixen com les diferents agrupacions formades amb *p* elements diferents, escollint-los d'entre els *n* elements dels quals disposem, considerant una variació diferent a una altra només si es difereixen en algun element (no influeix l'ordre de col·locació dels seus elements).
El número de combinacions que es poden construir es pot calcular fent ús de la següent fórmula:

$$C_n^p = \binom{n}{p} = \frac{n!}{(n-p)!\,p!}$$

on *n* és el número d'elements i *p* el número d'elements que volem que tingui cada agrupació.
Així si disposem de 5 fibres (veure *figura 50*), començarem amb 5 clústers. Per tant, la nostra *n* serà 5 i, com volem buscar les distàncies 2 a 2, la *p* valdrà 2. Aplicant la fórmula de dalt obtindrem un total de 10 distàncies (5! /((5-2)! * 2!).

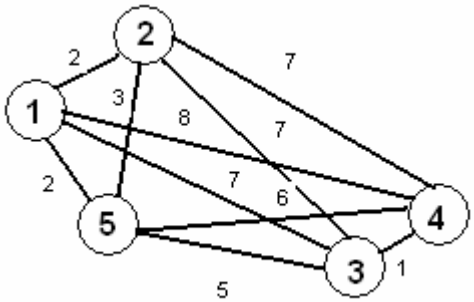


FIGURA 50.
Model format per 5 clústers amb les seves corresponents distàncies.

Sabent això, intentarem fer l'algorisme més eficient amb una segona implementació.

4.3.2.1.2.- 2a implementació

En aquesta segona implementació es disminueixen els càlculs de distàncies, evitant fer càlculs innecessaris. Les diferències respecte a la implementació anterior són els següents:

- Creem un vector per a guardar els centroides. Abans els calculàvem cada vegada. No era el principal motiu d'alentiment però també ajudarà a agilitzar l'algorisme.
- Evitem calcular distàncies que ja coneguem. Calculem totes les distàncies inicials, però cada cop que ajuntem dos clústers, només calcularem les distàncies que s'hagin vist afectades per aquest canvi.

El centroide de cada clúster correspondrà a la fibra mitjana d'aquest, que en un principi serà la única fibra de la que consta el clúster.
Llavors es calculen totes les distàncies entre tots els parells de clústers. En comptes de guardar únicament el valor numèric, guardem un objecte Distància que contindrà el valor de la distància i els números dels dos clústers que la formen.
D'aquesta manera podrem ordenar els objectes de distància segons el valor d'aquesta i disposar dels números de clúster als que fa referència. Caldrà per a això redefinir el mètode compareTo. La ordenació serà creixent, de manera que agafant el primer objecte obtindrem la menor distància entre dos clústers.
Mirarem a quins dos clústers es refereix la distància i procedirem a fusionar-los.
Llavors haurem d'actualitzar la fibra mitjana del clúster modificat. Pel que fa a les distàncies les haurem d'esborrar, re-indexar o actualitzar la distància segons convingui.

L'algorisme continuarà mentre hi hagi clústers a unir. La condició d'unió és que la distància no superi el llindar que hem escollit. Com a cas excepcional també podria passar que l'algorisme acabés perquè només quedés un únic clúster. En cas que això succeís, podríem suposar que em posat un llindar massa alt o tolerant.

```
Inicialització de variables

PER totes les fibres del model FER
      Crear clúster c
      Fibra f = agafar fibra
      Afegir fibra f al clúster c
      Afegir clúster c al vector de clústers
      Afegir fibra f al vector de centroides
FER
```

```
PER tots els clústers menys l'últim FER
      PER tots els clústers menys el primer FER
            distancia = trobarDistancia(fibra 1 del vector de
                  centroides, fibra 2 del vector de centroides)
            Crear Distancia amb els números de clúster i la
                  distància
            Afegir la distància al vector de distàncies
      FPER
FPER
Ordenar el vector de distàncies segons el valor de la distància
MENTRE continuar FER
      SI la primera distància del vector < llindar LLAVORS
            Fusionar els dos clústers que feien la distància mínima
            Eliminar el clúster 2 del vector de centroides
            Actualitzar centroide del clúster 1
            Actualitzar distàncies
            Ordenar el vector de distàncies
      ALTRAMENT
            No continuem iterant (continuar = fals)
      FSI
FMENTRE
```

PSEUDOCODI 2. *Clustering jeràrquic. 2a implementació.*

```
a = Obtenim el clúster 1 del vector de clústers
b = Obtenim el clúster 2 del vector de clústers
PER totes les fibres que tingui el 2n clúster FER
      Afegir fibra al clúster 1
FPER
Actualitzem el clúster 1 dins el vector de clústers
Eliminem el clúster 2 del vector de clústers.

Retorna el vector de clústers
```

PSEUDOCODI 3. *Fusionar clústers.*

```
PER tots els objectes distància FER
      SI el clúster 1 o el 2 de l'objecte distància coincideix amb
            el clúster 2 de la distància actual LLAVORS
            Esborrar distància del vector de distàncies
      ALTRAMENT
            SI el clúster 1 de l'objecte distància > clúster 2 de
                  la distància actual LLAVORS
                  Decrementar el valor del clúster 1 de l'objecte
                  distància
            FSI
            SI el clúster 2 de l'objecte distància > clúster 2 de
                  la distància actual LLAVORS
                  Decrementar el valor del clúster 2 de l'objecte
                  distància
            FSI
            SI qualsevol dels clústers de l'objecte distància =
                  clúster 1 de la distància actual LLAVORS
                  Buscar nova distància i actualitzar en el vector
            FSI
FPER
```

PSEUDOCODI 4. *Actualitzar distàncies.*

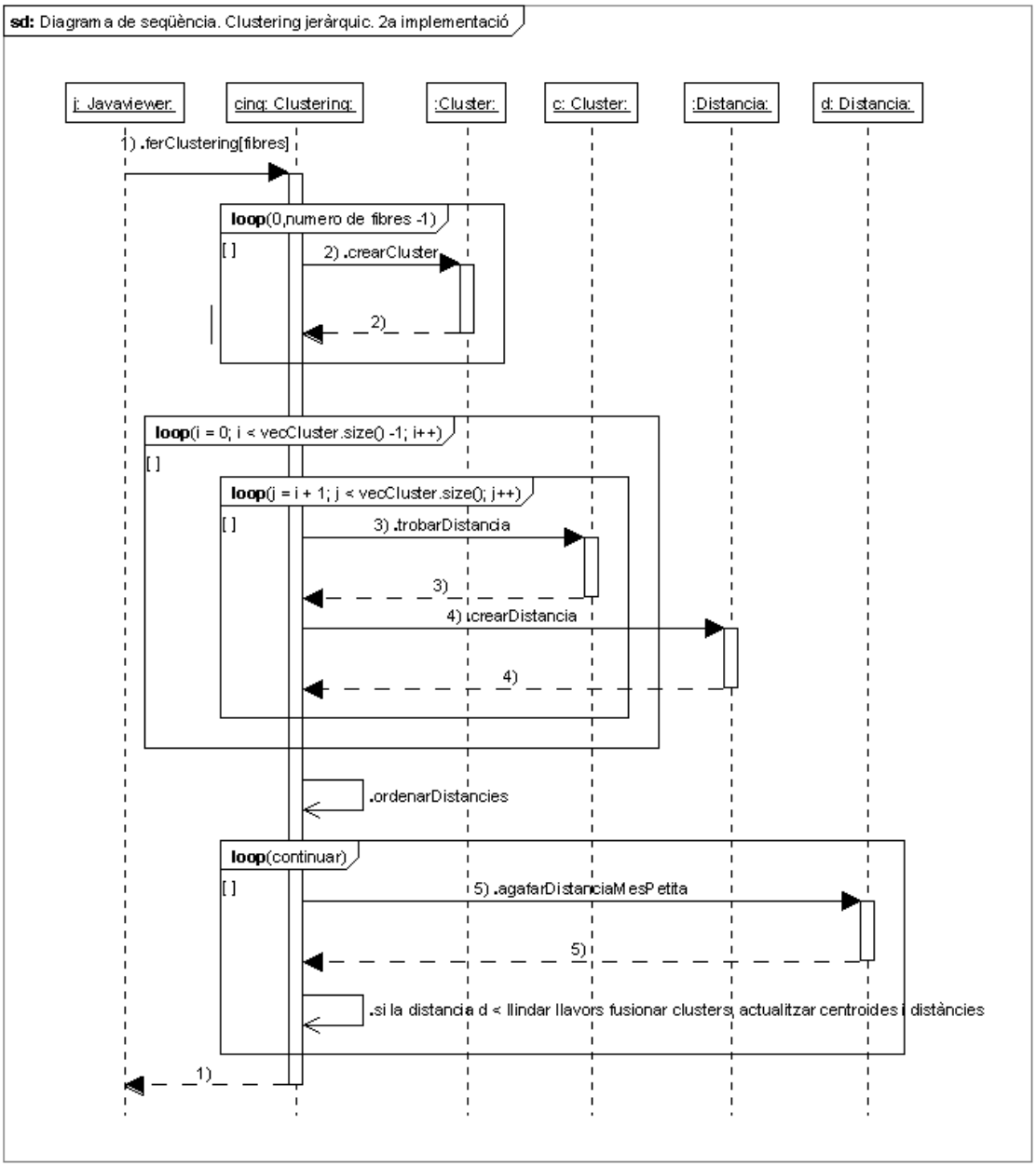


Diagrama de seqüència 7. *Segona implementació de l'algorisme jeràrquic.*

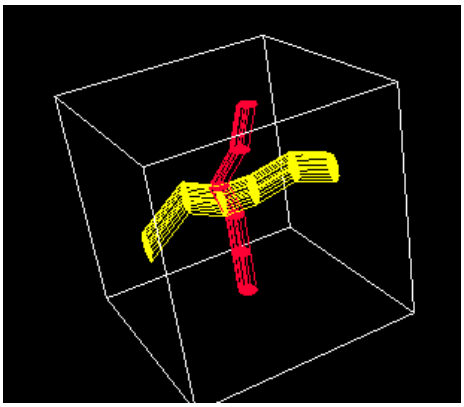
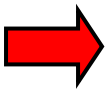


Figura 51. *Model ms_03.*



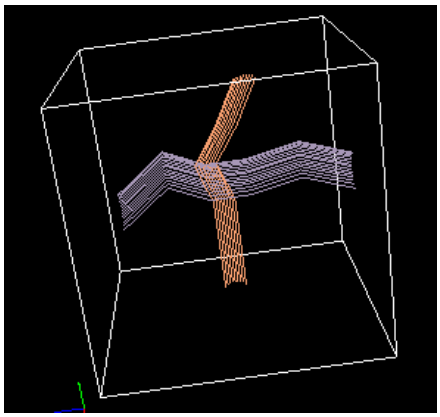


FIGURA 52. *Resultat d'aplicar l'algorisme de clustering jeràrquic sobre el model de l'esquerra*

Hem aconseguit poder visualitzar el clustering sobre el model de la figura X que té 214 fibres, però no ha estat instantani.
Proposem, doncs, intentar disminuir encara més el cost per el segon punt, el càlcul de les distàncies.
La idea ha estat disminuir el nombre de fibres a la meitat i agafar-les de forma aleatòria, ja que la meitat és un nombre prou considerable com per a que quedin ben repartides. Continuant amb l'exemple en el que hem comprovat que el temps d'execució era massa elevat, ara a l'utilitzar només la meitat de les fibres, disposarem de 107 fibres, que signifiquen un total de 5.671 distàncies, una quarta part. Hem executat l'algorisme amb un temps acceptable donant per sortida la imatge de la *figura 53*.

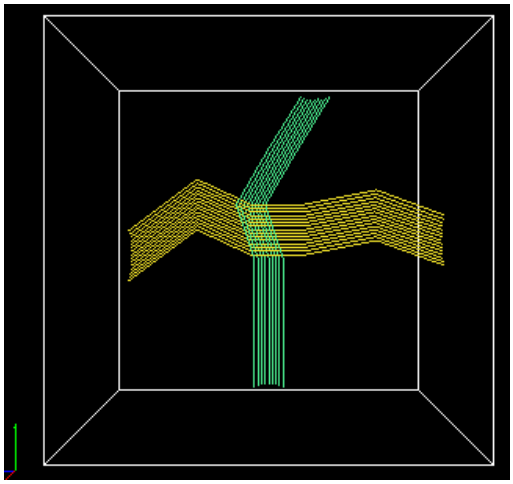


FIGURA 53.
Resultat d'aplicar l'algorisme de clustering jeràrquic sobre el model ms_03, reduint el nombre de fibres a la meitat.

Observem que el resultat és el que buscàvem, tenim els dos clústers separats correctament i el nombre de fibres d'un grup i altre no es veuen desproporcionades respecte a l'original. En aquest cas la generació de nombres aleatoris ha estat la esperada, però també, podria passar que n'agafés moltes d'una banda i poques de l'altra.

Aquesta manera de procedir no serà, doncs, massa científica. A més, cal saber de quantes fibres disposa un model real. Agafant el model del voluntari 2, un dels models reals dels quals disposem, obtenim un total de 1391 fibres. Si procedíssim de la mateixa manera, la meitat seria 695. Tenint en compte que amb el model sintètic de 214 fibres trigava massa, amb la meitat de les fibres del model real el cost de càlcul computacional és inacceptable. I si pensem en agafar menys fibres de la meitat podem arribar a un resultat que no s'ajusti a la realitat per la falta d'informació.
La solució que aquí es proposa es modificar lleugerament l'algorisme sense que això afecti massa a la idea general del mateix.

4.3.2.1.3.- 3a implementació

Aquesta implementació seguirà la següent idea:

- Agafar la primera distància que sigui inferior al llindar.

Canviem doncs una mica la filosofia de l'algorisme que diu que s'han de calcular totes les distàncies i agafar la menor. Amb la nostra tercera implementació agafarem la primera distància que no superi el llindar i unirem els dos clústers. Un cop units i calculat el nou centroide, tornarem a considerar tots els clústers sense descartar els que abans havíem mirat i no s'havien acceptat. Això ho farem així, perquè al unir dos clústers, el seu nou centroide serà diferent i podria ser que un dels clústers que abans no era acceptable, ara sí ho sigui.
En aquesta implementació ja no guardarem les distàncies com a objectes, sinó com a valors numèrics, perquè ara ja no es tractarà d'un recorregut de totes les distàncies, sinó d'una cerca, motiu pel qual es redueix significativament el cost. Com parlem d'una cerca, el fet de guardar les distàncies no reduiria el cost, ja que aquestes s'haurien d'actualitzar, la qual cosa ens obligaria a recórrer-les totes per saber quines s'han de modificar.

```
Inicialització de variables

PER totes les fibres del model FER
      Crear clúster c
      Fibra f = agafar fibra
      Afegir fibra f al clúster c
      Afegir clúster c al vector de clústers
      Afegir fibra f al vector de centroides
FPER
MENTRE continuar FER
      continuar = fals
      MENTRE quedin no arribem al penúltim clúster i no trobat FER
            MENTRE no arribem a l'últim clúster i no trobat FER
                  trobarDistancia entre les dues fibres mitjanes de
                  cada clúster guardades al vector de centroides
                  SI la distància < llindar LLAVORS
                        trobat = cert
                  FSI
            FMENTRE
      FMENTRE
      SI trobat LLAVORS
            continuar = cert
            Fusionar clústers que tenen la distància < llindar
            Esborrar la fibra mitjana del vector de centroides que
            es correspon amb el clúster 2
            Actualitzar la fibra mitjana del vector de centroides
            que es correspon amb el clúster 1
      FSI
```

PSEUDOCODI 5. *Clustering jeràrquic. Tercera implementació.*

A continuació en el diagrama de seqüència 8 es mostra aquesta tercera implementació.

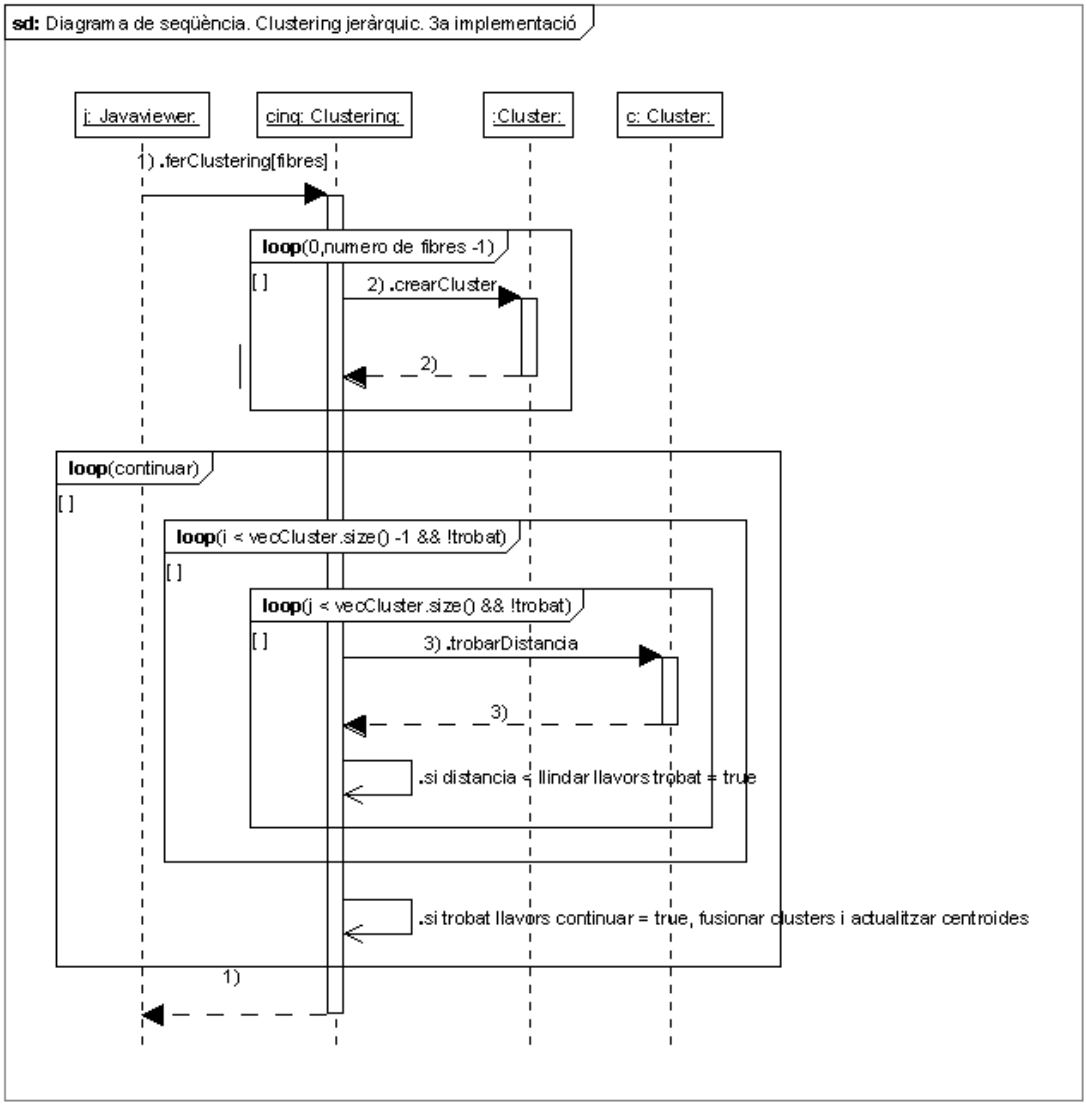


Diagrama de seqüència 8. *Tercera implementació de l'algorisme jeràrquic.*

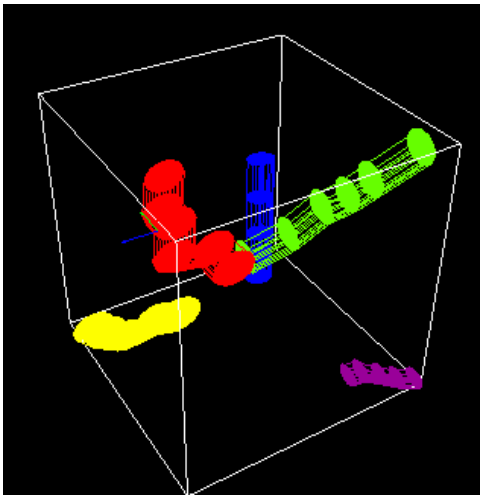


FIGURA 54. *Model ms_04.*

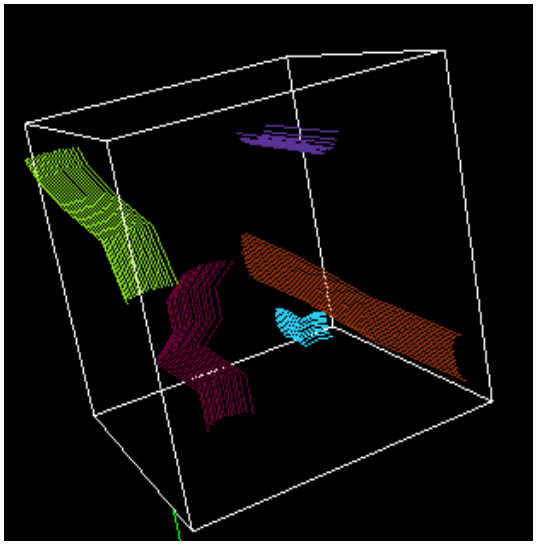


FIGURA 55. *Resultat d'aplicar l'algorisme de clustering jeràrquic sobre el model de l'esquerra*

El model de la figura 54 disposa de 780 fibres repartides en 5 clústers i podem veure a la figura 55 que el resultat de l'algorisme és correcte.

Per últim, provem amb un model que té un nombre de fibres similar al d'un model real. Aquest model de la *figura 56* que veiem a la dreta disposa de 1391 fibres. Veiem, també, com en aquest cas el resultat és vàlid. Tenim els dos clústers que es creuen, separats, un de color blau fosc i l'altre de color blau clar.

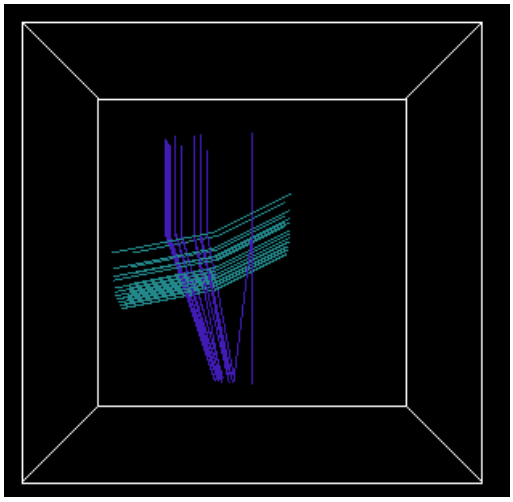


FIGURA 56. *Resultat d'aplicar l'algorisme de clustering jeràrquic.*

Ens ha faltat, però fer una última consideració. I és que, hem de pensar que les fibres en un model sintètic estan ordenades, és a dir, que les fibres de les respectives bandes estaran guardades correlativament en el fitxer. Per aquesta raó, hem desordenat les fibres per a fer la prova més real.

El resultat ha estat que tot i trobar un agrupament correcte de les fibres en bandes, el temps d'execució no ha estat raonable. Ha trigat prop dels 10 minuts.

La rapidesa de l'algorisme està massa condicionada a que les fibres quedin més o menys *desordenades*.

4.3.2.2.- Resultats sobre models reals

Les proves fetes sobre els models sintètics ens donen la seguretat de la correcta implementació de l'algorisme. Per a finalitzar la implementació del clustering jeràrquic provem sobre models reals.

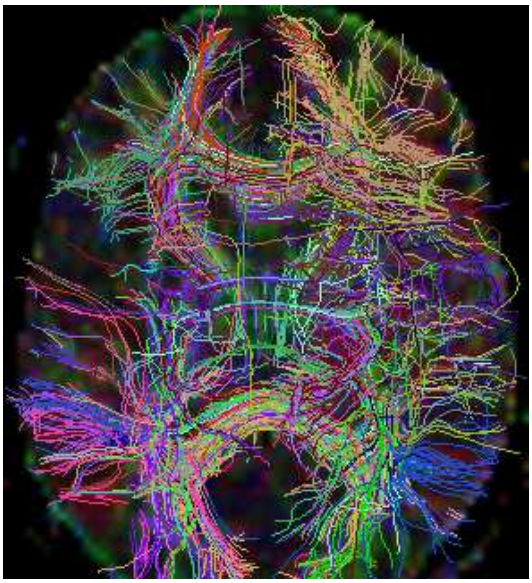


FIGURA 57. Resultat d'aplicar el clustering jeràrquic sobre un model real.

Hem aplicat l'algorisme de clustering jeràrquic sobre el model real del voluntari 2. Hem utilitzat un llindar de 10 i el mètode de la distància dels punts finals.
El resultat ha estat el de la *figura 57*, amb un total de 76 clústers.
Aquest nombre tan elevat de clústers indica que el llindar que hem escollit és massa baix.
Hauríem de triar un llindar més elevat i així hi hauria menys clústers, però no ho hem fet perquè el temps d'execució d'aquest algorisme ha estat de 25 minuts. Tenint en compte que el mètode de les distàncies del punts finals és el més ràpid, no podem acceptar aquest mètode.
Hem provat d'utilitzar el mètode de la mitjana dels punts més propers i tot i esperar més d'una hora no l'hem vist acabar.

En l'apartat anterior al fer proves amb models sintètics amb un nombre similar de fibres, el temps, tot i ser força elevat, no ho era tant com al fer les execucions sobre models reals. Hem vist que la diferència està en la complexitat de les fibres, ja que, les d'un model sintètic poden tenir uns 10 punts, mentre que les dels models reals en tenen centenes. Aquest fet fa que els càlculs de les distàncies es disparin.

Cal remarcar que he estat treballant amb un portàtil *(*)*, cosa que explica en part la lentitud de les execucions. Però, encara que ho hagués fet amb un ordinador de sobretaula, els temps no haguessin estat bons.

(*) Característiques del portàtil: Processador Intel Core 2 Duo 1.6 GHz, 2MB RAM.

Conclusions

L'algorisme de clustering jeràrquic dóna molt bons resultats, sense haver de decidir a priori quin ha de ser el nombre de clústers resultants, la qual cosa és una bona característica, ja que no sempre podrem saber amb certesa els grups en els que es divideix el model.
Per altra banda, el gran desavantatge és que el cost d'execució és molt si partint d'un gran nombre de fibres.

4.3.3.- L'algorisme K-means

4.3.3.1.- Implementació sobre models sintètics

Per a fer la implementació del clustering exclusiu k-means necessitem establir la k. La k serà el nombre de grups en que acabaran agrupades les fibres un cop aquest finalitzi.

4.3.3.1.1.- 1a implementacio

El mètode començarà, doncs, inicialitzant k clústers. Per cada clúster es tindrà un centroide que haurem seleccionat de manera aleatòria entre tot el conjunt d'entrada.
Llavors, el procediment passa per fer un recorregut de totes les fibres i comparar la distància entre la fibra actual i els centroides dels k clústers. D'aquesta manera la fibra actual s'afegirà al clúster que tingui un centroide més proper.
Un cop s'hagin repartit totes les fibres es recalcularà el valor de tots els centroides i es repetirà el procés de mirar les distàncies entre les fibres del conjunt i els centroides dels clústers mentre els centroides varïin entre iteració i iteració. Entre iteració i iteració buidarem els clústers per a tornar a emplenar-los segons les noves distàncies.
Finalment, retornarem el vector de clústers que tindrà una mida k, amb totes les fibres d'entrada repartides entre aquests.

```
Inicialització de variables

PER clústers com haguem seleccionat (k) FER
      Crear clúster c
      Afegir clúster c al vector de clústers
      Escollir fibra a l'atzar i afegir al vector de centroides.
FPER
MENTRE continuar FER
      continuar = fals
      PER totes les fibres del model FER
            Buscar distància entre la fibra i el primer centroide
            PER tots els centroides menys el primer FER
                  Buscar distància entre la fibra i el centroide
                  Guardar el valor i el clúster al que pertany la
                  menor distància entre l'anterior i  l'actual
                  Assignar la distància actual com a anterior
            FPER
            Afegir la fibra al clúster més proper
      FPER
      PER tots els clústers FER
            SI la distància entre el centroide anterior i l'actual
            del clúster ha variat LLAVORS
                  Continuar = cert
            FSI
            Actualitzar el centroide
      FPER
```

***PSEUDOCODI 6**. Clustering k-means.*

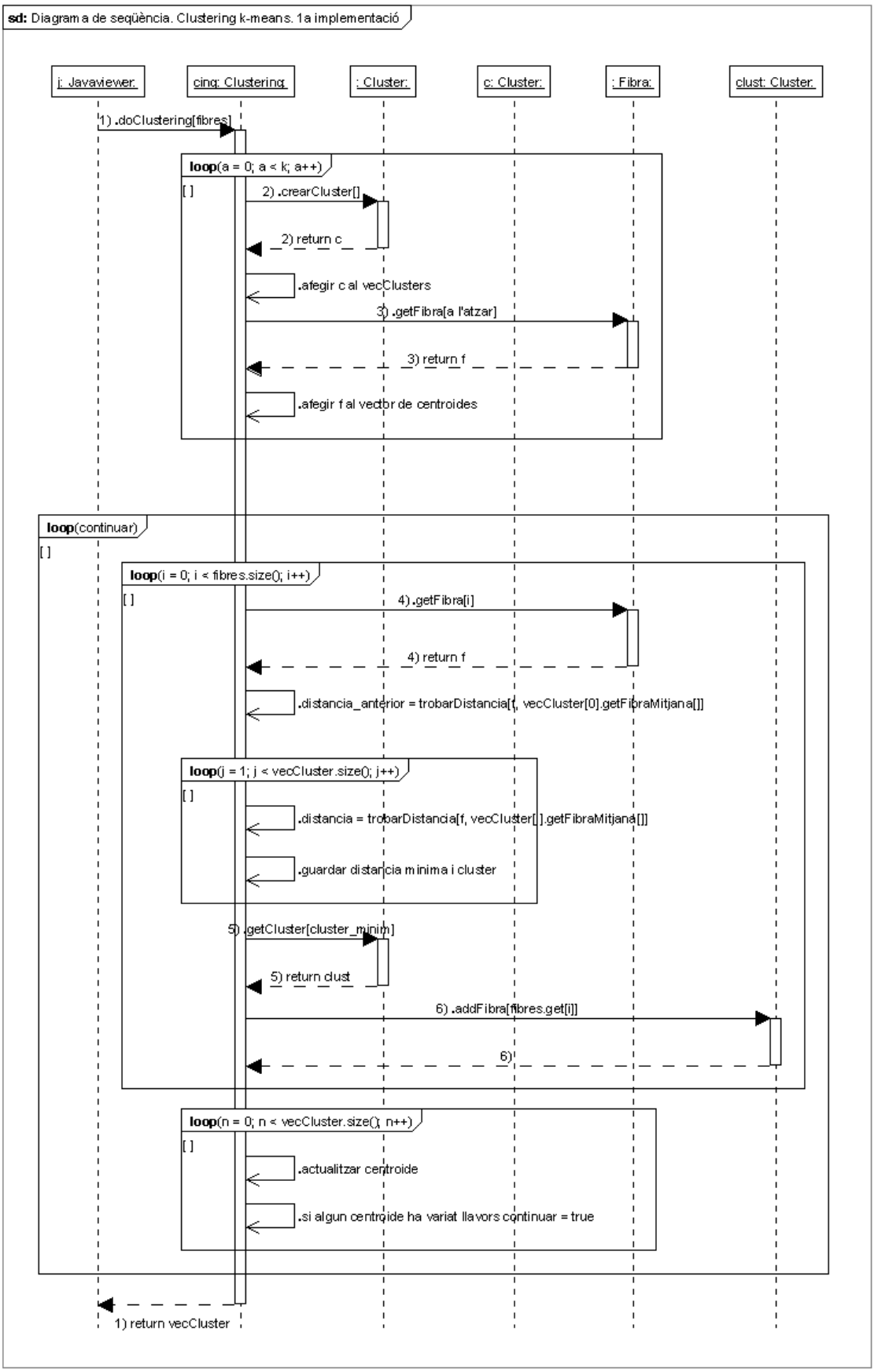


Diagrama de seqüència 9. *Algorisme k-means.*

El *diagrama de seqüència 9* ens mostra el funcionament de l'algorisme k-means.

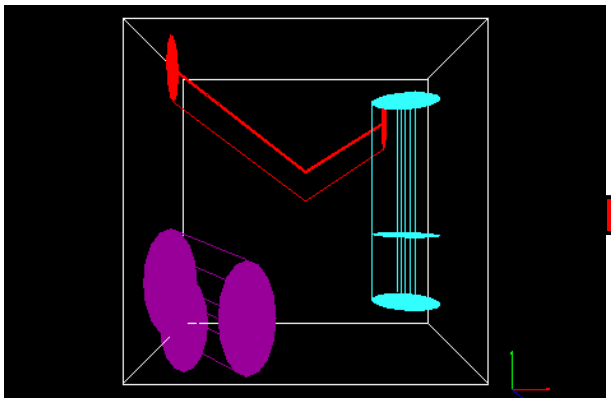


FIGURA 58. *Model ms_01.*

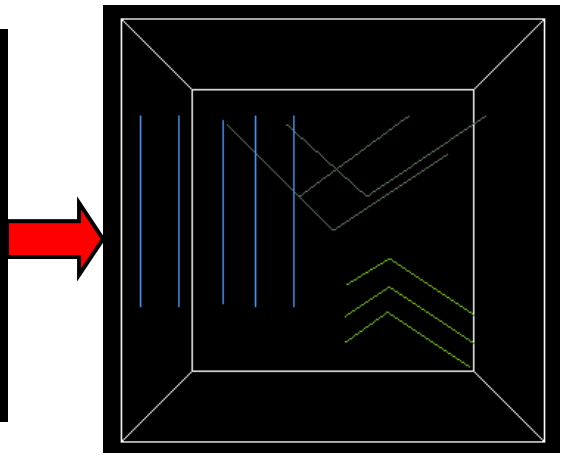


FIGURA 59. *Resultat d'aplicar el clustering k-means sobre el model de la figura 58.*

A l'exemple de dalt (*figura 58* i *figura 59*) veiem com aplicant l'algorisme k-means sobre el model senzill ms_01 funciona correctament. Li hem assignat a la k el valor 3, ja que, es pot apreciar clarament que hi ha d'haver 3 clústers. El mètode de distància utilitzat ha estat la distància dels punts més propers.

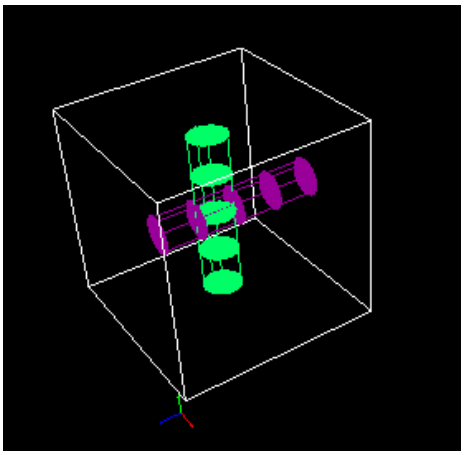


FIGURA 60. *Model ms_02.*

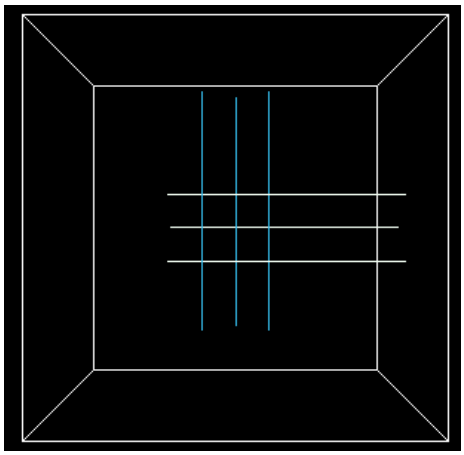


FIGURA 61. *Resultat d'aplicar el clustering k-means sobre el model de la figura 60.*

En aquest segon exemple (*figura 60* i *figura 61*) el valor que li hem assignat a k ha estat 2. Veiem que també troba els dos clústers que esperàvem. En aquest cas hem utilitzat com a mesura de distància la distància dels punts finals, ja que és força més significativa que no pas la que havíem utilitzat en l'exemple anterior que era la distància dels punts més propers. El problema trobat utilitzant aquesta distància és que al creuar-se les fibres, els punts més propers són precisament els que s'encreuen i es crea així un resultat que no és el que ens interessa. També podríem haver escollit qualsevol dels altres mètodes de distància.

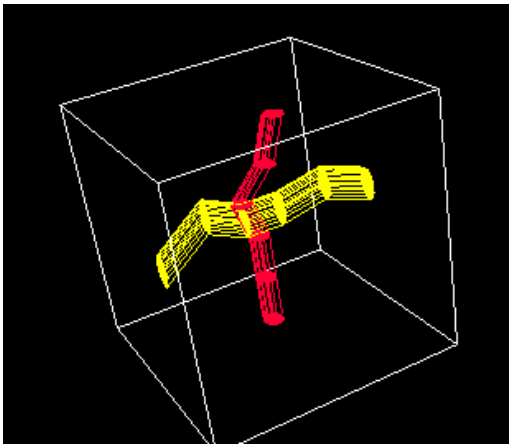


FIGURA 62. *Model ms_03.*

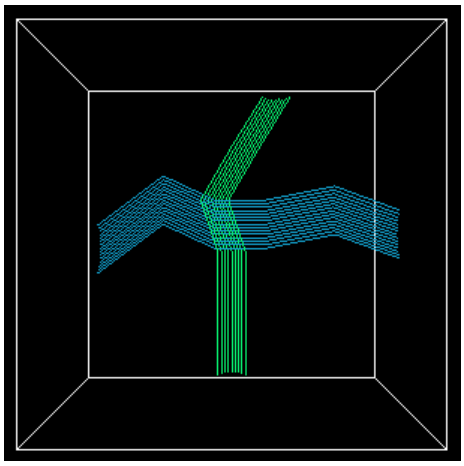


FIGURA 63. *Resultat d'aplicar el clustering k-means sobre el model de la figura 62.*

La tercera prova (*figura 62* i *figura 63*) l'hem fet sobre un model de més fibres per assegurar-nos que aquest no sigui un problema. La sortida ha estat l'esperada i el temps d'execució també. En aquest cas també hem assignat un valor de 2 a la k i ens hem valgut del mètode de la mitjana dels punts més propers per comprovar que no alentís en excés el resultat, ja que és el que té un cost computacional més elevat.

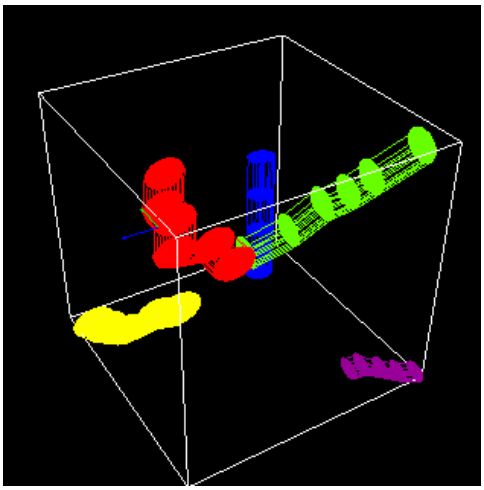


FIGURA 64. *Model ms_04.*

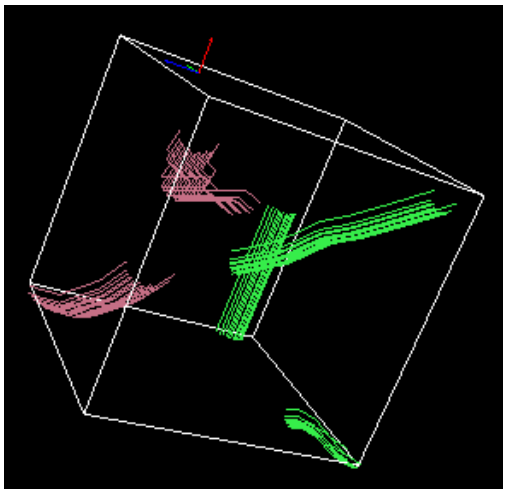


FIGURA 65. *Resultat d'aplicar el clustering k-means sobre el model de la figura 64.*

Fent una prova sobre el model de 780 fibres i 5 clústers (*figura 64* i *figura 65*) també hem aconseguit una execució en un temps acceptable utilitzant la mitjana dels punts més propers. Hem mantingut la k a 2 per mostrar el major problema d'aquest tipus d'algorisme. Com es pot apreciar ha partit el model en dos grups. Els grups inicials en vermell i groc han passat a ser els roses del resultat, i els grups de color blau, verd i lila, són els verds. Podem dir, però, que els dos grups que ha separat són correctes, ja que, ha agafat les fibres d'una manera agrupada, és a dir, que no ha agafat una fibra d'aquí i una d'allà.

La *figura 66* és el resultat de l'algorisme aplicat sobre el mateix model que en l'exemple anterior però posant-li els 5 clústers en els que estan repartides les fibres realment. El resultat ha estat una mala distribució de les fibres.
El motiu és la dependència d'aquest tipus d'algorisme de la selecció aleatòria dels centroides.
A continuació proposem una segona implementació per intentar solucionar aquest problema.

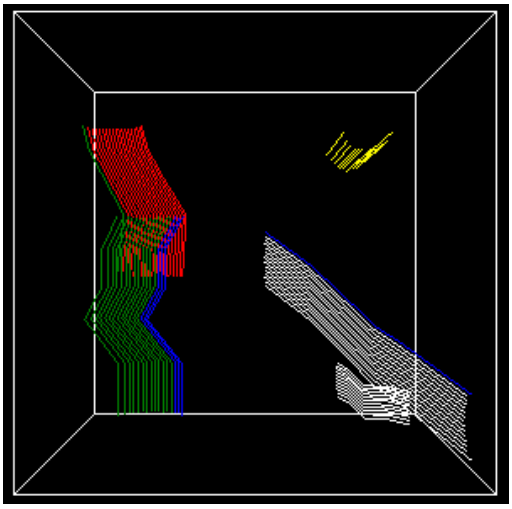


FIGURA 66. *Resultat d'aplicar el clustering k-means.*

4.3.3.1.2.- 2a implementació

El canvi realitzat en aquesta segona implementació es troba en la part de l'elecció dels centroides inicials.
Els continuarem agafant a l'atzar perquè nosaltres a priori no sabem a quins grups pertanyen, però només els acceptarem si la distància entre el proper centroide escollit i la resta de centroides és prou gran. Haurà de superar un valor llindar que li proporcionarem.

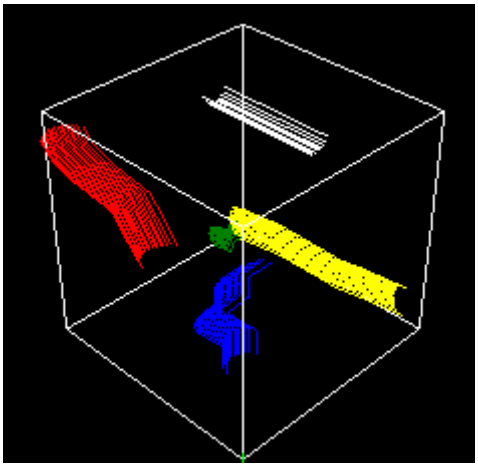


FIGURA 67. *Resultat del k-means.*

Veiem a la *figura 67* com el canvi ha donat bons resultats.

Hem fet varies execucions per descartar que el bon resultat fos casual.
En totes, el resultat ha estat com el de la figura.
El valor llindar que li hem entrat ha estat 8.

4.3.3.2.- Resultats sobre models reals

Provant sobre models reals hem obtingut les següents sortides:

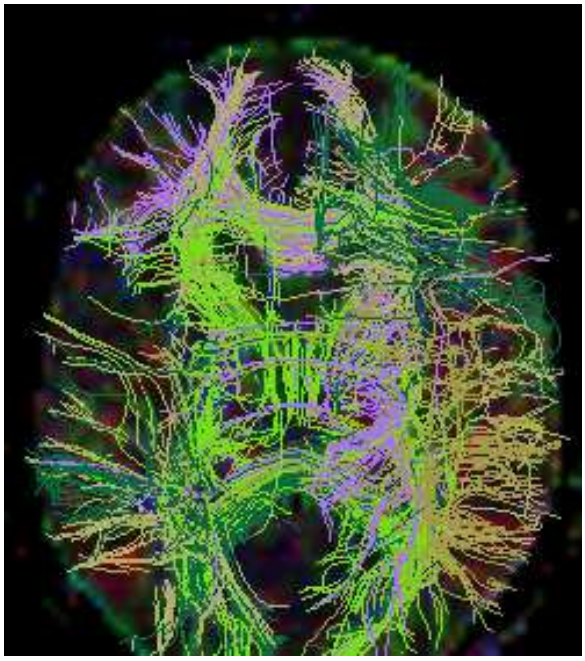


FIGURA 68. Resultat del k-means.

La imatge de la *figura 68* és el resultat d'aplicar l'algorisme amb una k de 5 i el mètode de distància de la mitjana dels punts més propers.

Es pot veure la separació de les zones, tot i que no queden massa ben definides.

A la imatge de la *figura 69* s'han aplicat els mateixos paràmetres que en l'anterior. Però, tot i això, el resultat no ha estat el mateix. Si ens fixem, per exemple, en la zona lila de la imatge superior i la zona lila de la imatge de la dreta, veiem que no hi ha hagut una unió de les mateixes fibres, tot i algunes d'elles coincidir.
Aquesta diferència és deguda a la dependència del mètode dels centroides inicials.
La zona groga de la imatge de la dreta, en canvi, si que sembla coincidir bastant amb la marró de dalt.

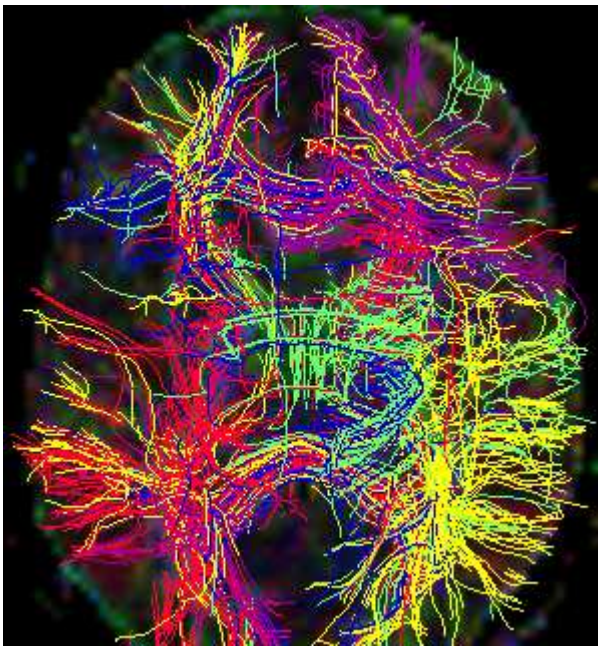


FIGURA 69. Resultat del k-means.

4.3.4.- L'algorisme Fuzzy C-Means

4.3.4.1.- Implementació sobre models sintètics

Per a fer la implementació del clustering de solapament fuzzy c-means necessitem establir la k. De la mateixa manera com en l'algorisme k-means, aquesta k serà el nombre de grups o clústers en que acabaran agrupades les fibres un cop aquest finalitzi.

4.3.4.1.1.- 1a implementació

El mètode començarà, doncs, inicialitzant un vector amb k clústers buits. També farem un nou vector on marcarem totes les fibres com a no tractades per passar a marcar-les com a tractades a mida que les anem utilitzant.
El següent pas es basa en assignar coeficients de forma aleatòria. Aquests coeficients són valors entre 0 i 1 i existeix un coeficient diferent per a cada fibra i número de clúster. També en aquest mateix pas creem un vector de coeficients antics, inicialitzats tots a 1000. Podem veure aquest emmagatzemament de dades en la *figura 70*.

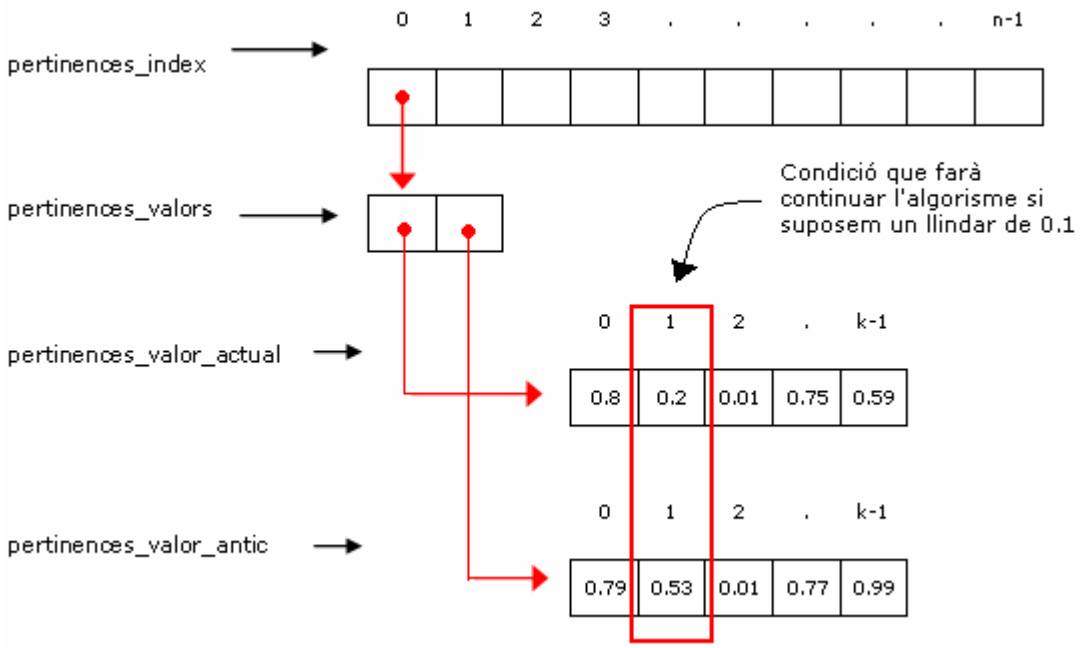


***Figura 70**. Esquema de l'emmagatzemament de pertinences*

Un cop fetes totes les assignacions, busquem la fibra amb un coeficient de pertinença més alt, per a cada clúster. D'aquesta manera haurem començat a inicialitzar el vector de centroides.

Fins ara ens hem dedicat a inicialitzar l'algorisme *(Veure pseudocodi 7)*. A continuació entrem en el desenvolupament de l'algorisme *(Veure pseudocodi 8)*.
Busquem quin és el coeficient més elevat entre els diferents parells de fibra-clúster i actualitzem el centroide que pertany al clúster trobat fent la mitjana entre la fibra que hi ha al centroide i la fibra que ens ha donat el coeficient de pertinença més elevat. Aquesta nova fibra serà el nou centroide per al clúster tractat.
Llavors, calculem els nous coeficients de pertinença entre fibres i clústers. El valor d'aquests coeficients els trobem fent la inversa de la distància entre fibra i clúster. Així al vector de coeficients antics hi posem els actuals, i al vector de coeficients actuals hi posem els que acabem de calcular.
El fet de guardar els coeficients antics i actuals s'explica per ser el motiu de continuïtat de l'algorisme. Si la diferència entre qualsevol d'aquests parells és major que un cert llindar, continuarem iterant. En cas contrari sortirem del bucle, la qual cosa significarà que ja tenim les agrupacions de fibres fetes.
Per acabar, hem de afegir físicament les fibres als clústers per a permetre la posterior visualització *(Veure pseudocodi 9)*. Això és degut a que durant l'algorisme no hem realitzat aquest pas, sinó que anàvem actualitzant coeficients només. El motiu de procedir així és pel fet de que tractem un algorisme de clustering de solapament on una fibra pot pertànyer a qualsevol dels clústers amb major o menor dependència.
Com en els altres algorismes s'acaba retornant el vector de clústers que serà de mida k i tindrà totes les fibres d'entrada repartides entre aquests.

```
Inicialització de variables

PER cada clúster FER
      c = nou clúster()
      vecCluster->afegir(c)
FPER
PER totes les fibres del model FER
      fibres_tractades->afegir(0)
      PER cada clúster FER
            X = nou aleatori(0,1)
            pertinences_actuals->afegir(x)
            pertinences_antigues->afegir(1000)
      FPER
      pertinences_valor->afegir(pertinences_actuals)
      pertinences_valor->afegir(pertinences_antigues)
      pertinences_index->afegir(pertinences_valor)
FPER
PER cada clúster FER
      pertinença_antiga <- -1
      PER totes les fibres f del model FER
            pertinença_actual <- pertinences_actuals.agafar()
            SI pertinença_actual > pertinença_anterior LLAVORS
                  pertinença_anterior = pertinença_actual
                  fibra = f
            FSI
```

***PSEUDOCODI 7**. Clustering fuzzy c-means. Inicialització de l'algorisme*

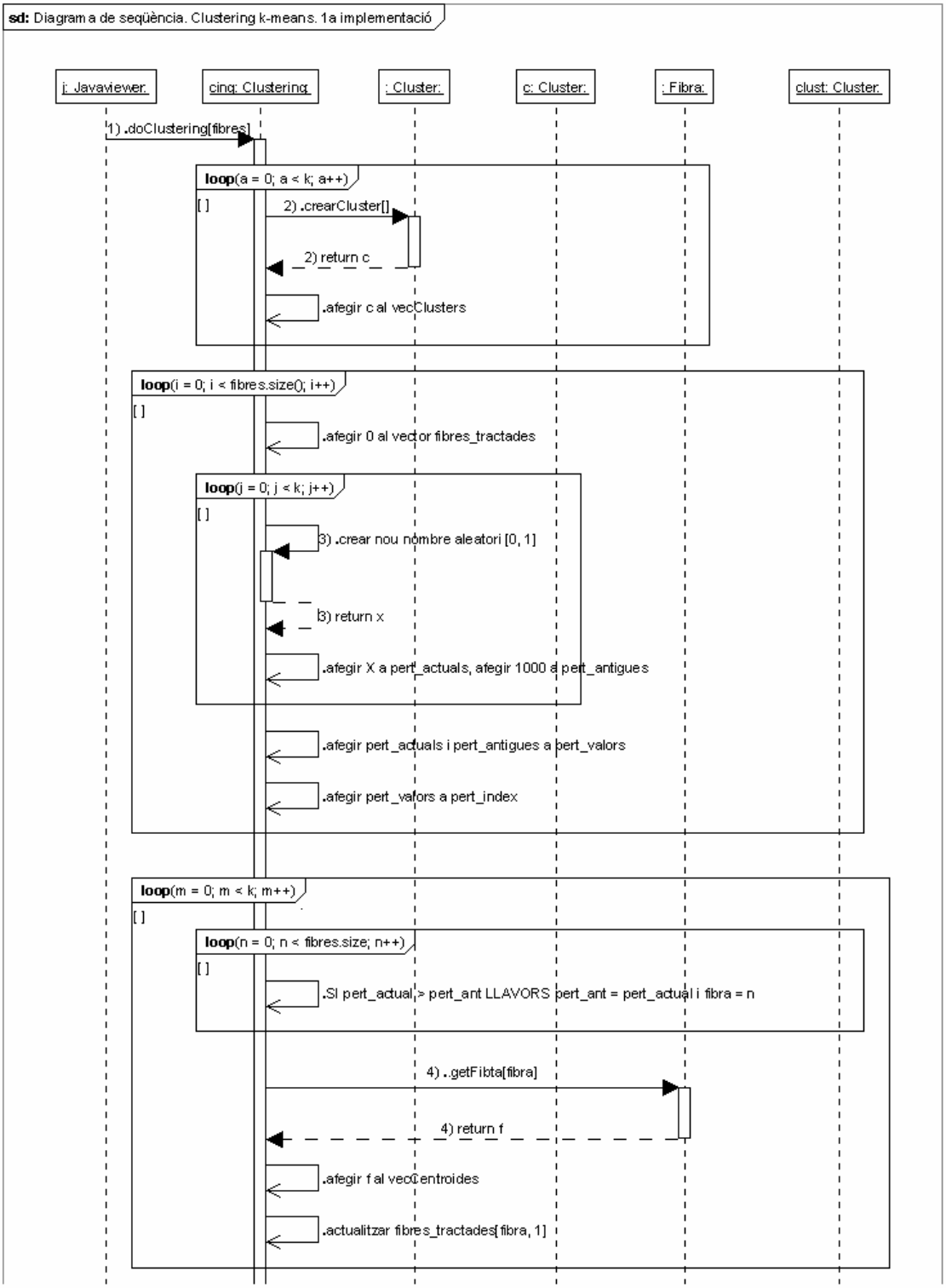
```
MENTRE continuar FER
      continuar = fals; pert_anterior = -1
      PER cada fibra f del model FER
            PER cada clúster c FER
                  pertinença_actual <- pertinences_actuals.agafar()
                  SI pert_actual > pert_anterior LLAVORS
                        pert_anterior = pert_actual
                        fibra = f
                        cluster = c
                  FSI
            FPER
            Afegir la fibra al centroide més proper
      FPER
      fibra_centroide <- vecCentroides->agafar(cluster)
      fibra_mitjana <- calcularFibraMitjana(fibra_centroide, fibra)
      vecCentroides->actualitzar(cluster, fibra_mitjana)
      fibres_tractades->actualitzar(fibra, 1)
      PER cada fibra f del model FER
            PER cada clúster c FER
                  pertinença_anterior = pertinença_actual
                  distancia<-trobarDistancia(f, c)
                  nou_coeficient<-1/distancia
                  SI nou_coef - antic_coef > llindar LLAVORS
                        continuar <- cert
                  FSI
            FPER
      FPER
FMENTRE
```

***PSEUDOCODI 8**. Clustering fuzzy c-means. Cos de l'algorisme*

```
PER cada clúster c FER
      coef_antic = -1
      PER cada fibra f del model FER
            coef_actual = (pert_index->agafar(f))->agafar(c)
            SI coef_actual > coef_antic LLAVORS
                  coef_antic <- coef_actual
                  coef_index <- c
            FSI
      FPER
      (vecCluster->agafar(coef_index))->afegir(f)
FPER

Retorna vecCluster
```

***PSEUDOCODI 9**. Clustering fuzzy c-means. Fi de l'algorisme*


sd: Diagrama de seqüència. Clustering k-means. 1a implementació
i: Javaviewer:
cinq: Clustering:
: Cluster:
c: Cluster:
: Fibra:
clust: Cluster:
1) .doClustering[fibres]
loop(a = 0; a < k; a++)
[ ]
2) .crearCluster[]
2) return c
.afegir c al vecClusters
loop(i = 0; i < fibres.size(); i++)
[ ]
.afegir 0 al vector fibres_tractades
loop(j = 0; j < k; j++)
[ ]
3) .crear nou nombre aleatori [0, 1]
3) return x
.afegir X a pert_actuals, afegir 1000 a pert_antigues
.afegir pert_actuals i pert_antigues a pert_valors
.afegir pert_valors a pert_index
loop(m = 0; m < k; m++)
[ ]
loop(n = 0; n < fibres.size; n++)
[ ]
.SI pert_actual > pert_ant LLAVORS pert_ant = pert_actual i fibra = n
4) ..getFibta[fibra]
4) return f
.afegir f al vecCentroides
.actualitzar fibres_tractades[fibra, 1]

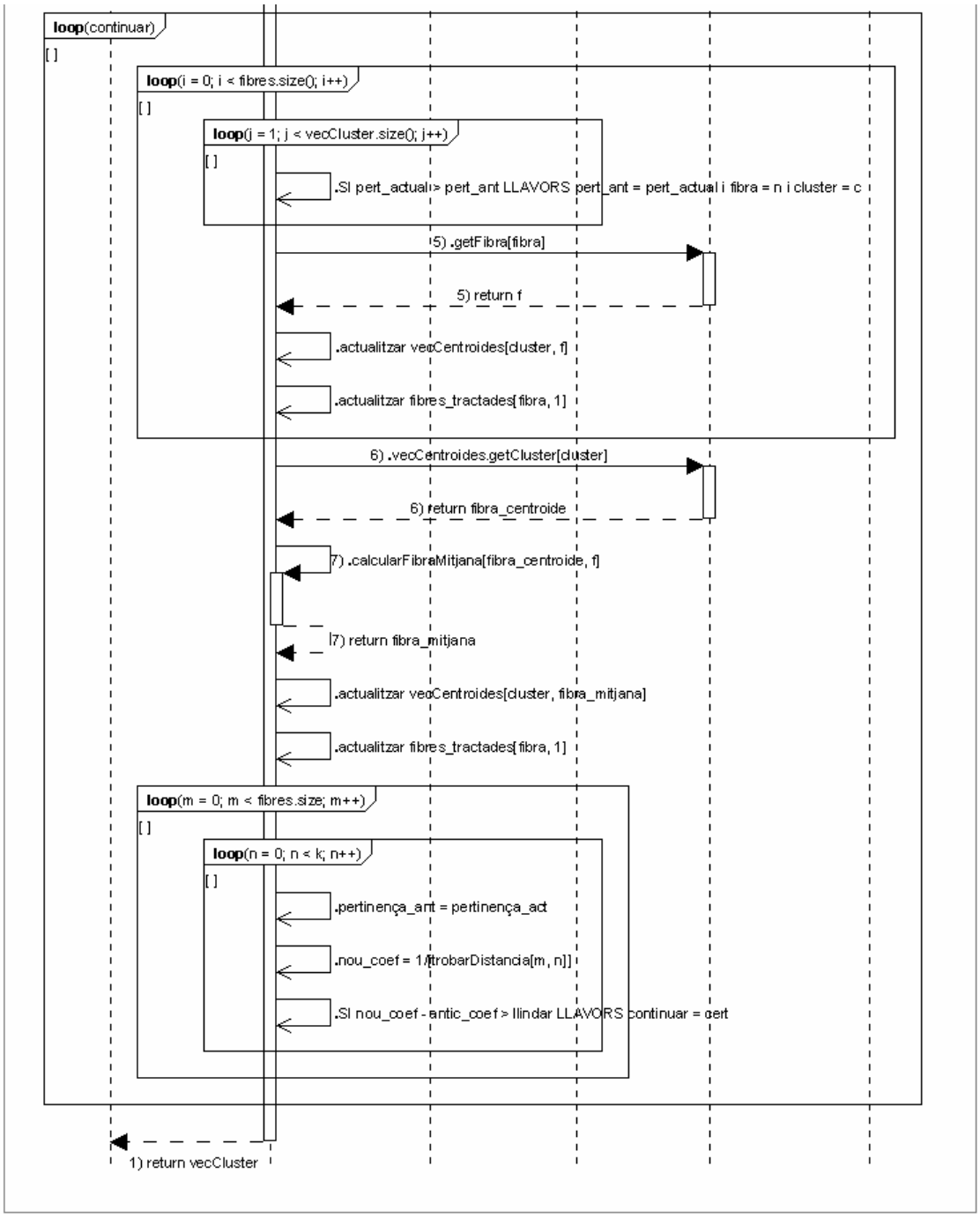


***Diagrama de seqüència 10**. Algorisme Fuzzy C-means*

En el *diagrama de seqüència 10* podem veure el funcionament de l'algorisme fuzzy c-means.

Per a analitzar el funcionament d'aquest algorisme d'agrupament, hem començat per els models més senzills, de la mateixa manera com hem fet amb els altres algorismes. A la *figura 71* podem veure el model sintètic i a la *figura 72* el resultat d'aplicar-hi el clustering. Com es pot apreciar, les fibres s'ha agrupat correctament en els tres clústers que formen el model. Per inicialitzar l'algorisme hem seleccionat el nombre de clústers igual a 3 i un llindar de 0.1. Cal notar una comoditat en aquest llindar respecte el de l'algorisme jeràrquic, per exemple, ja que abans el llindar s'utilitzava per comparar-lo amb una distància. Això vol dir que depenen del model el llindar podria variar segons la disposició de les fibres i caldria trobar-ne un que s'hi adeqüés. Ara, però, no tenim aquest problema, perquè no treballem en distàncies, sinó amb coeficients que normalitzem, prenen aquests, valors entre 0 i 1.

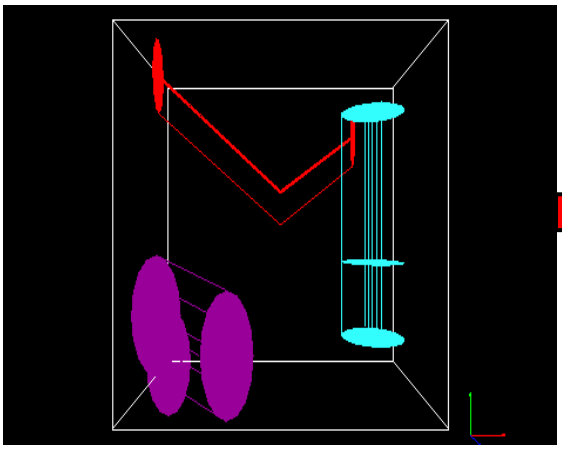


Figura 71. *MODEL ms_01.*

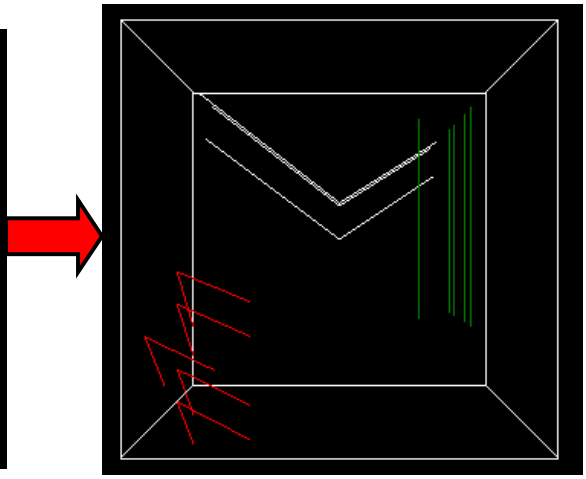


Figura 72. *Resultat d'aplicar l'algorisme fuzzy c-means sobre la figura X. Pla XY*

Després de provar diverses vegades sobre el mateix model, ens adonem que no sempre surt l'agrupament com volem. Ho podem veure a la *figura 73*.

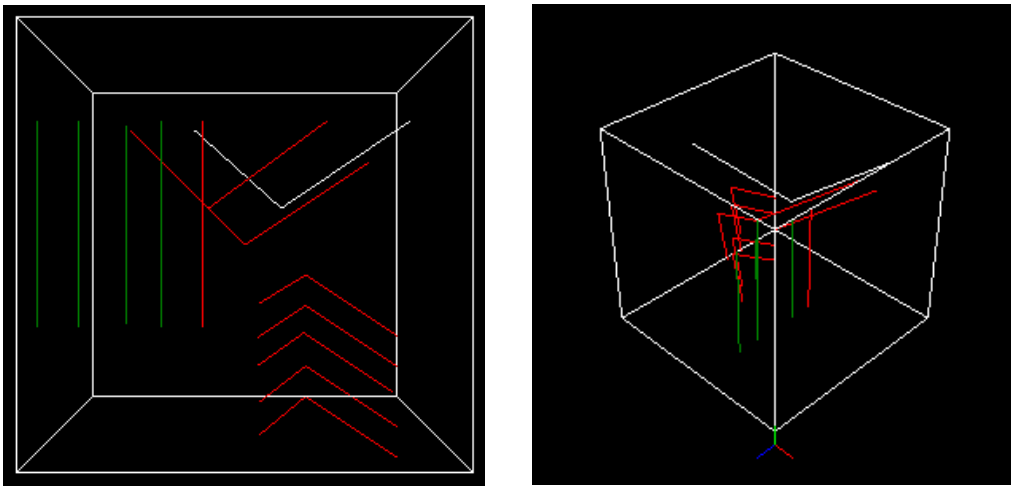


Figura 73. *Resultat d'aplicar l'algorisme fuzzy c-means sobre la figura 71. A l'esquerra: pla YZ, a la dreta: vista 3D*

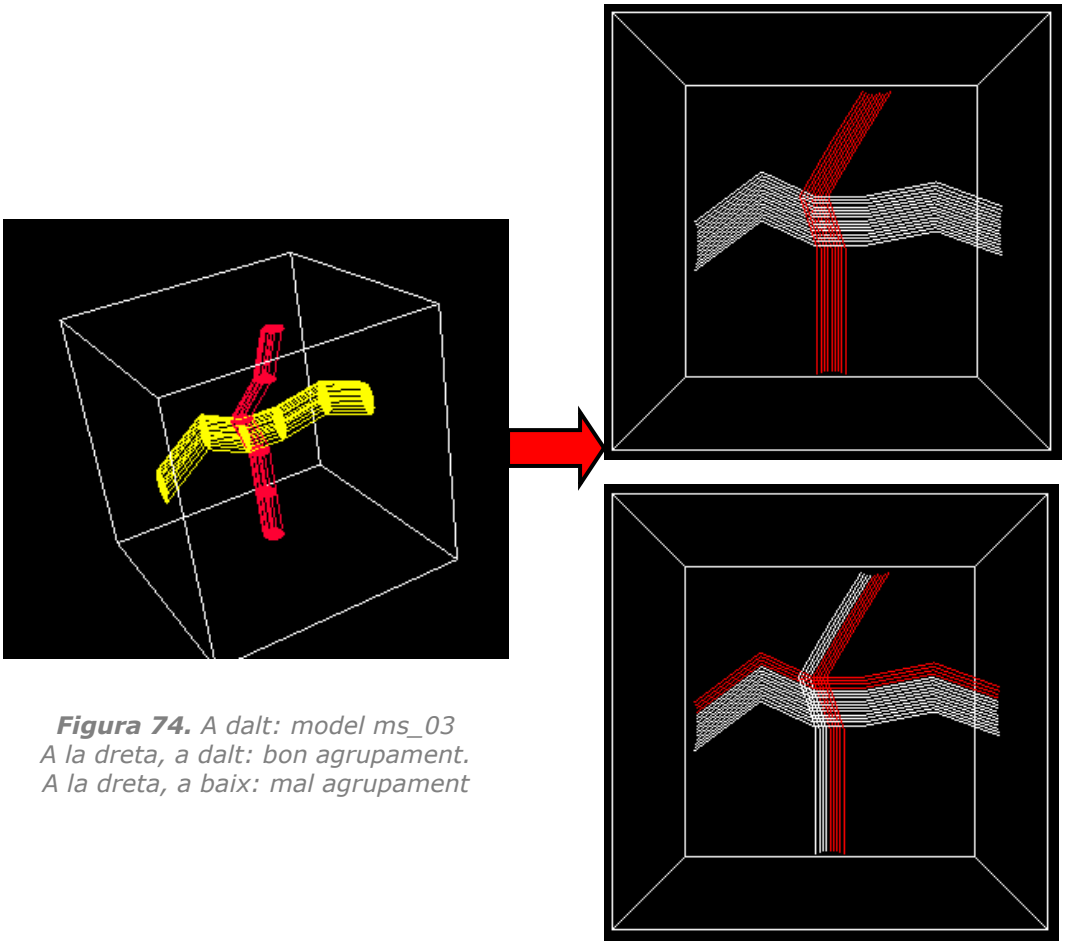


Figura 74. *A dalt: model ms_03*
A la dreta, a dalt: bon agrupament.
A la dreta, a baix: mal agrupament

A la figura X, veiem que ens trobem davant el mateix problema. El motiu el podem situar en la inicialització dels centroides. Si mirem la imatge on el resultat no és el que buscàvem, podem suposar que els dos centroides s'han inicialitzat amb fibres del mateix clúster.
De fet aquest és el major problema d'aquest algorisme així com també del k-means, on ja ens hi havíem trobat. El resoldrem de la mateixa manera en una segona implementació.

4.3.4.1.2.- 2a implementació

En el moment en que inicialitzem els centroides de forma aleatòria, hi posarem una restricció que serà la següent:
La fibra que inicialitzi el centroide haurà d'estar prou allunyada de les fibres que inicialitzen els altres centroides.
A més, aprofitem en aquesta segona implementació per desordenar les fibres com fèiem en l'algorisme k-means. Així fem més real la situació, tenint en compte que en els models reals les fibres no estan ordenades per clústers i en els nostres models sintètics, sí.

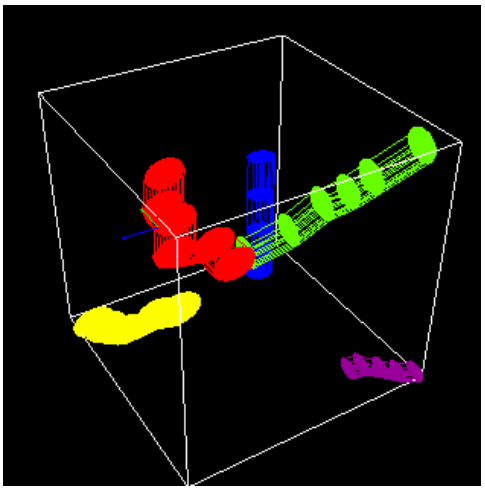


FIGURA 75. *Model ms_04.*

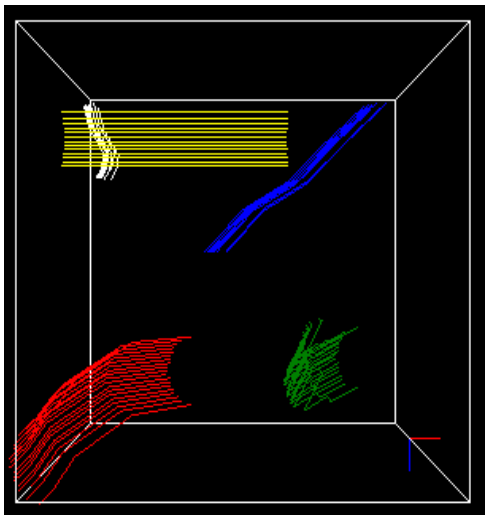
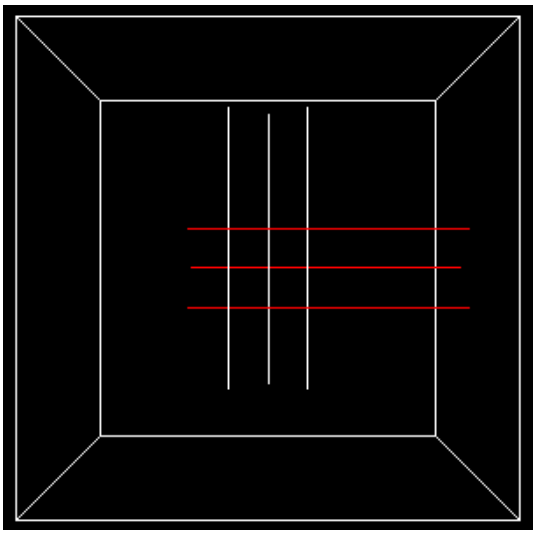


FIGURA 76. *Resultat d'aplicar l'algorisme sobre la figura 75. Pla XZ*



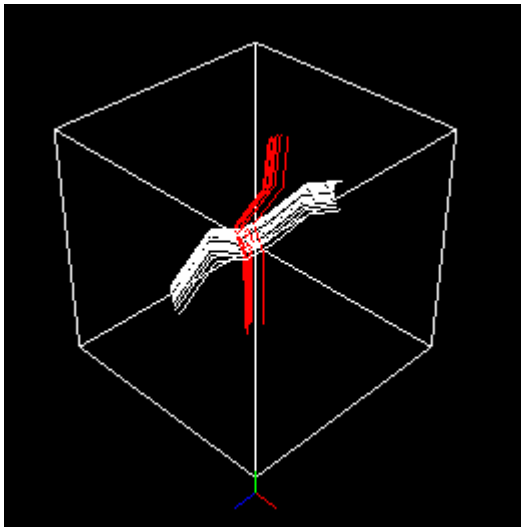


FIGURA 77. *Resultat d'aplicar l'algorisme.*
A l'esquerra sobre el model ms_02 (pla YZ)
A la dreta sobre el model ms_03 (vista 3D)

A les *figures 76 i 77* veiem com obtenim els resultats esperats després de fer repetides proves per a descartar possibles casualitats.

Cal veure que en el cas on es nota una millora considerable és en el model sintètic ms_04, ja que, a major nombre de clústers més probable és que es faci una mala assignació de centroides inicials.

Per tant, hem resolt bé el problema, i el temps d'execució no s'ha vist afectat. En l'apartat 4.3.5 estudiarem amb detall els temps d'execució.

4.3.4.2.- Resultats sobre models reals

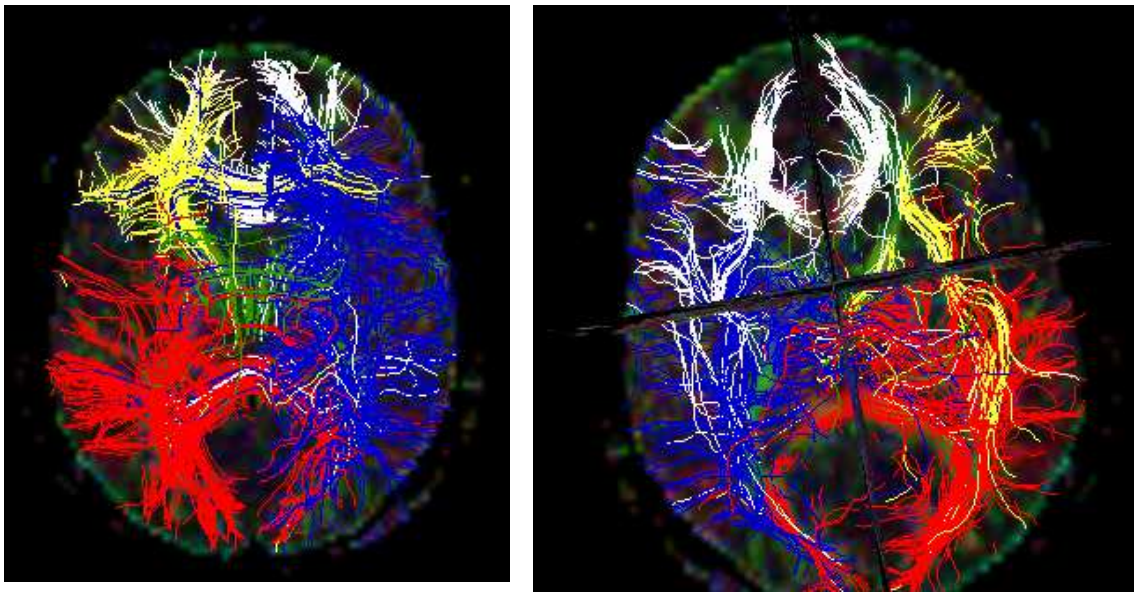


FIGURA 78. *Resultat d'aplicar l'algorisme sobre un model real. Pla XZ i vista 3D*

A la *figura 78* podem observar el resultat d'aplicar l'algorisme de clustering fuzzy c-means sobre el model real del voluntari 2. L'agrupació de les fibres ha estat força encertada. Sobretot, destaca la banda en forma de U que hi ha situada a la part superior, pintada de color blanc, que ha quedat molt ben definida.
Els paràmetres escollits han estat: 5 clústers, 0.2 pel llindar de variació de coeficients de pertinença i 8 pel llindar d'inicialització dels centroides.
Comparant els resultats amb els dels altres algorisme podríem valorar que aquesta ha estat la millor sortida, però el temps que ha trigat en realitzar-se ha estat de més de quatre hores i mitja. La causa és haver escollit un llindar massa baix. Al ser tan baix, quan el comparem amb la variació entre els coeficients vells i actuals, torna a iterar moltes vegades (massa).
A la *figura 79*, hem posat un llindar més alt, mantenint el mateix número de clústers i veiem una sortida on, també, les fibres queden força ben agrupades. La diferència ha estat en el temps. En aquest cas hem passat de quasi 5 hores a 5 minuts. Cal notar que tot i que en la segona implementació del mètode s'ha deixat de inicialitzar els centroides de forma totalment aleatòria per passar a tenir en compte que la distància entre els clústers sigui prou significativa, segueix sent un punt condicionant. En aquesta execució el clúster pintat de color vermell podria no ser massa encertat.

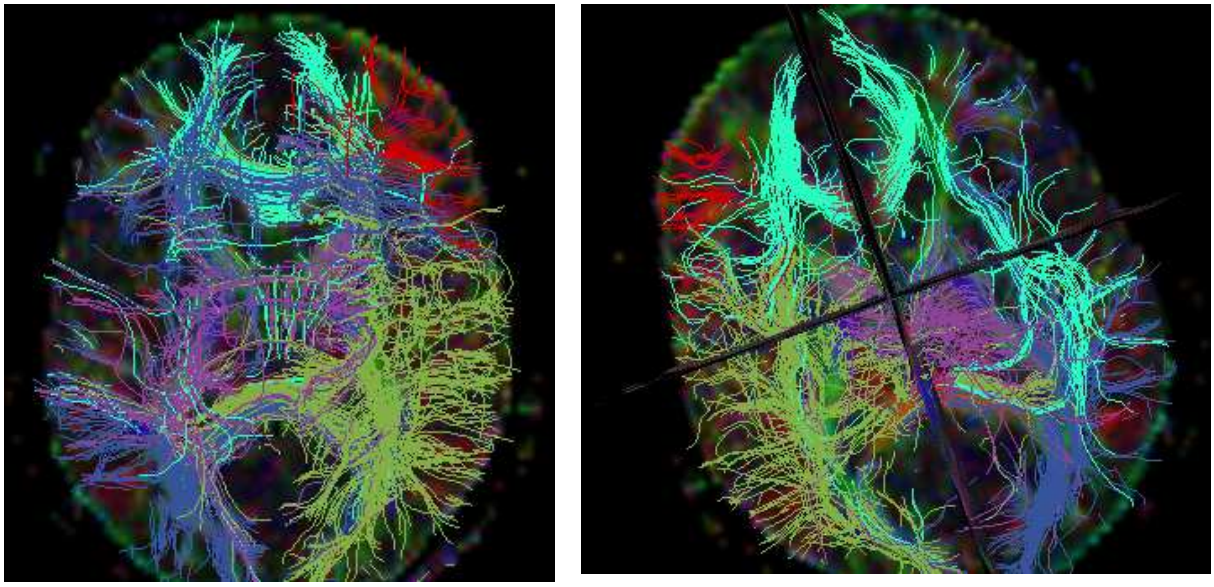


FIGURA 79. *Resultat d'aplicar l'algorisme sobre un model real. Pla XZ i vista 3D*

4.4.- Optimització del clustering a partir de la simplificació de les fibres

4.4.1.- Explicació de l'optimització del clustering

En cada un dels apartats on tractàvem els models reals hem fet notar que el temps era massa elevat i sempre donàvem la mateixa explicació: la complexitat dels models.
En algun dels apartats es proposava agafar només un tant per cent de les fibres de forma aleatòria, però vèiem que aquesta opció podria ser molt dolenta depenent de quina forma s'agafessin les fibres.
La opció que es proposa és reduir el nombre de punts de cada fibra.

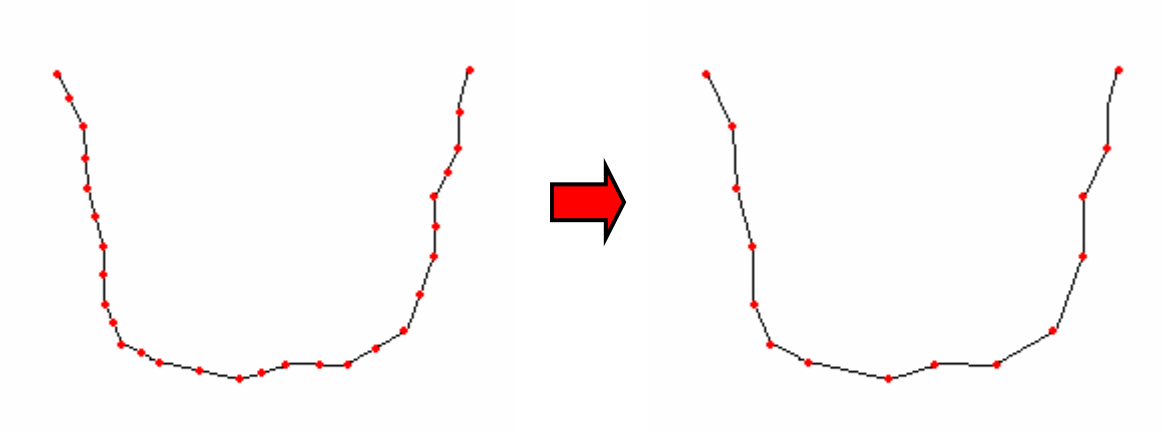


***FIGURA 80**. Reducció moderada del nombre de punts d'una fibra.*

A la *figura 80* es mostra la idea que proposem. El que hem fet ha estat reduir el nombre de punts de la fibra a la meitat. No s'aprecia cap diferència en la forma. Obtindrem el mateix resultat si fem la mateixa pràctica sobre fibres d'un model real. Hem de vigilar que agafem un nombre de punts significatiu, ja que si reduïm massa el número podem canviar la forma de la fibra. Vegem-ho a l'exemple de la *figura 81*.

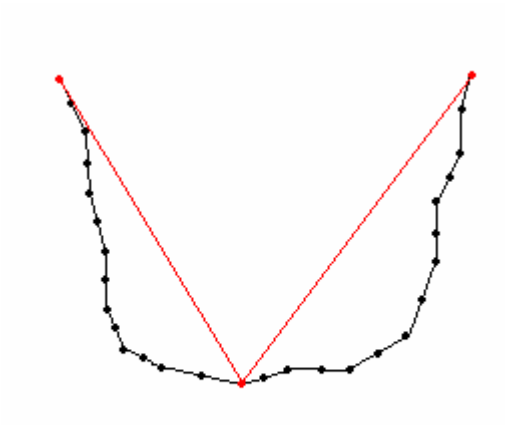


***FIGURA 81**. Reducció excessiva del nombre de punts d'una fibra.*

La línia negra amb els seus punts corresponents és la fibra original i la vermella la resultant de reduir-ne els punts. Veiem que en aquest cas l'algorisme podria no funcionar correctament. Un cas més extrem, seria agafar només el punt inicial i el punt final, de manera que en comptes d'una corba ens quedaria un segment horitzontal.

Per aquest motiu és important saber fins a quin percentatge podem reduir.
El primer que farem abans d'implementar el mètode, serà estudiar la complexitat de les fibres dels models reals. Ho podem veure a la *taula 2*, per al model real sobre el voluntari 1 i a la *taula 3*, per al model real sobre el voluntari 2.

VOLUNTARI 1

0 - 10	0	**150 - 159**	26	**300 - 309**	1
10 - 19	140	**160 - 169**	16	**310 - 319**	0
20 - 29	99	**170 - 179**	14	**320 - 329**	0
30 - 39	84	**180 - 189**	14	**330 - 339**	1
40 - 49	83	**190 - 199**	5	**340 - 349**	0
50 - 59	87	**200 - 209**	8	**350 - 359**	0
60 - 69	82	**210 - 219**	2	**360 - 369**	0
70 - 79	80	**220 - 229**	2	**370 - 379**	0
80 - 89	89	**230 - 239**	1	**380 - 389**	0
90 - 99	85	**240 - 249**	2	**390 - 399**	0
100 - 109	61	**250 - 259**	1	**400 - 1000**	0
110 - 119	62	**260 - 269**	0	**1000 - 1500**	0
120 - 129	41	**270 - 279**	1	**1500 - 2000**	0
130 - 139	42	**280 - 289**	1	**2000 - 2500**	0
140 - 149	29	**290 - 299**	1	**> 2500**	0

***TAULA 2**. Quantitat de punts de les fibres del model real 1*

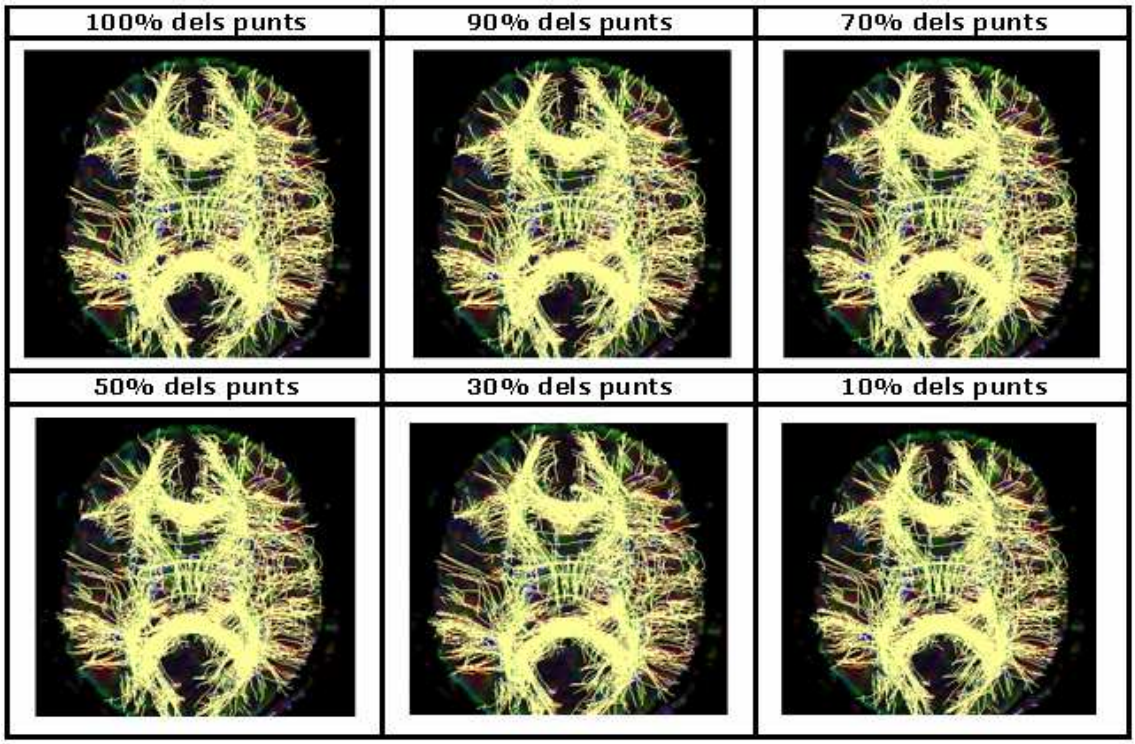
VOLUNTARI 2

0 - 10	0	**150 - 159**	28	**300 - 309**	0
10 - 19	101	**160 - 169**	22	**310 - 319**	0
20 - 29	84	**170 - 179**	23	**320 - 329**	0
30 - 39	91	**180 - 189**	16	**330 - 339**	0
40 - 49	82	**190 - 199**	10	**340 - 349**	0
50 - 59	108	**200 - 209**	8	**350 - 359**	0
60 - 69	92	**210 - 219**	5	**360 - 369**	2
70 - 79	114	**220 - 229**	8	**370 - 379**	0
80 - 89	76	**230 - 239**	6	**380 - 389**	0
90 - 99	107	**240 - 249**	2	**390 - 399**	0
100 - 109	68	**250 - 259**	3	**400 - 1000**	2
110 - 119	78	**260 - 269**	4	**1000 - 1500**	3
120 - 129	72	**270 - 279**	4	**1500 - 2000**	1
130 - 139	51	**280 - 289**	1	**2000 - 2500**	0
140 - 149	38	**290 - 299**	0	**> 2500**	1

***TAULA 3**. Quantitat de punts de les fibres del model real 2*

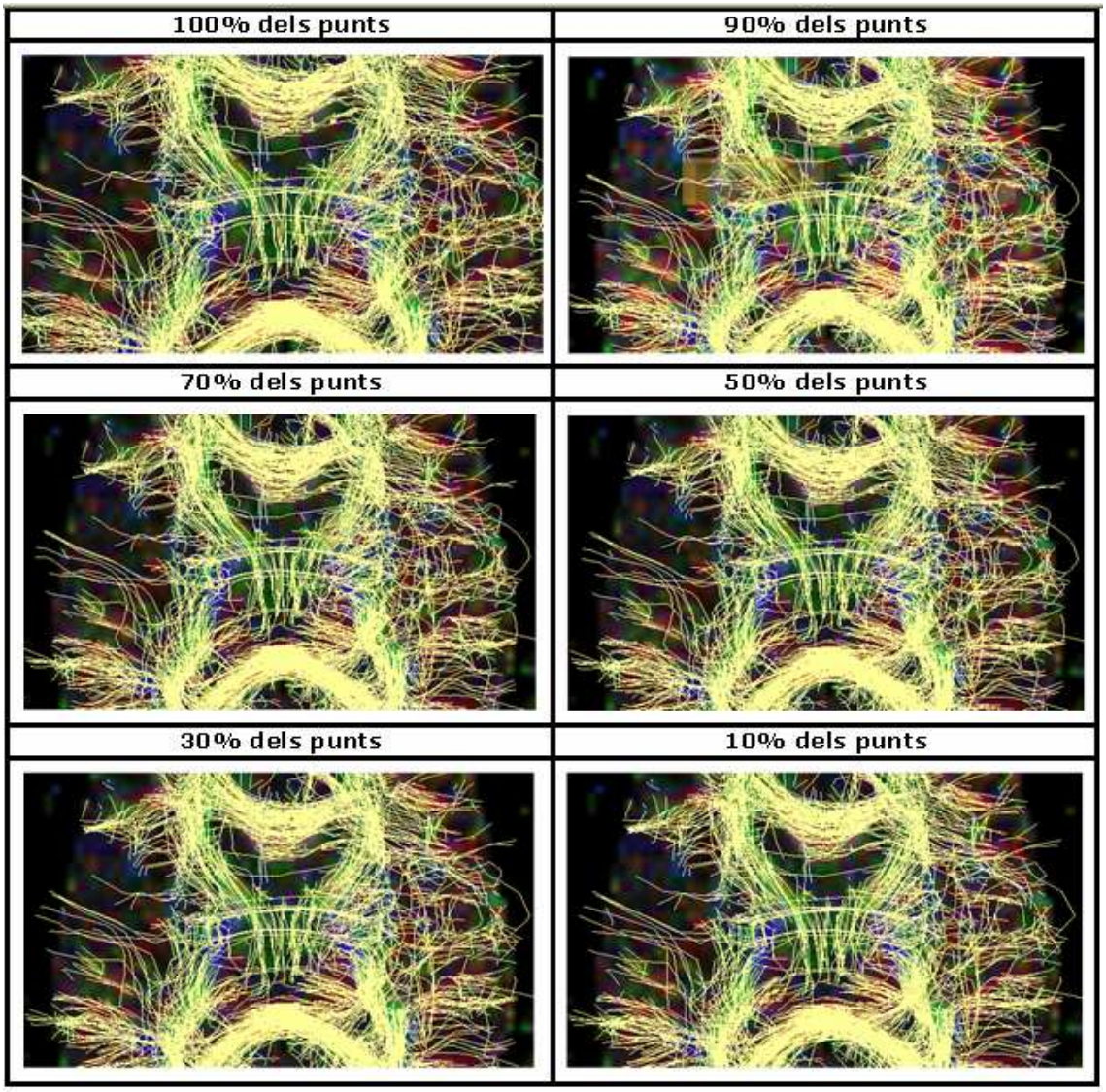
Les columnes en negreta de l'esquerra de les taules ens indiquen el número de punts i les de la dreta, el nombre de fibres d'aquell model que estan dins aquell rang de punts. Si ens fixem en el resultat dels dos models, la quantitat de punts que tenen la majoria de les fibres es concentren entre els 10 i els 150 punts (primera columna). A mesura que va augmentant el nombre de punts, el nombre de fibres disminueix, però tot i així n'hi ha un nombre considerable.
Abans de procedir a reduir el nombre de punts de les fibres hem de tenir en compte que sempre el mínim de punts en una fibra ha de ser 2, ja que, amb un únic punt no podrem construir la fibra.

El primer pas és escollir un tant per cent que significarà el nombre de punts que agafarem del total. Si suposem un 20%, llavors per cada cinc punts n'agafarem 1. Construirem una nova fibra amb aquesta reducció de punts. Farem el mateix per totes les fibres del model.
A la *taula 4* es veu com queden les fibres després de reduir-ne els punts amb diferents percentatges.



***TAULA 4**. Quantitat de punts de les fibres del model real 2*

La taula consta del resultat de fer el traçat de fibres amb la totalitat dels punts, un 90%, un 70%, un 50%, un 30% i un 10%, respectivament. En tots els casos la imatge que mostrem és una captura que correspon al pla XZ.
Tot i haver reduït el tamany de les fibres, amb aquesta vista general, semblen no haver-se vist modificades, per aquest motiu s'ha construït una altra taula (*taula 5*) amb les mateixes imatges augmentades.



***TAULA 5**. Quantitat de punts de les fibres del model real 2*

En aquesta taula on hem fet un zoom a totes les imatges de la taula anterior, sí podem apreciar diferències de unes respecte les altres. Veiem que quan més alta és la reducció de punts de les fibres, més perden la curvatura original. La diferència és clara si ens centrem en les imatges del 100% i el 10%.

Un cop fetes aquestes observacions, podem afirmar que a la pràctica ens serà molt útil, ja que hem disminuït el nombre de punts significativament, sense alterar gaire la forma del model original.

4.4.2.- Resultats sobre models reals

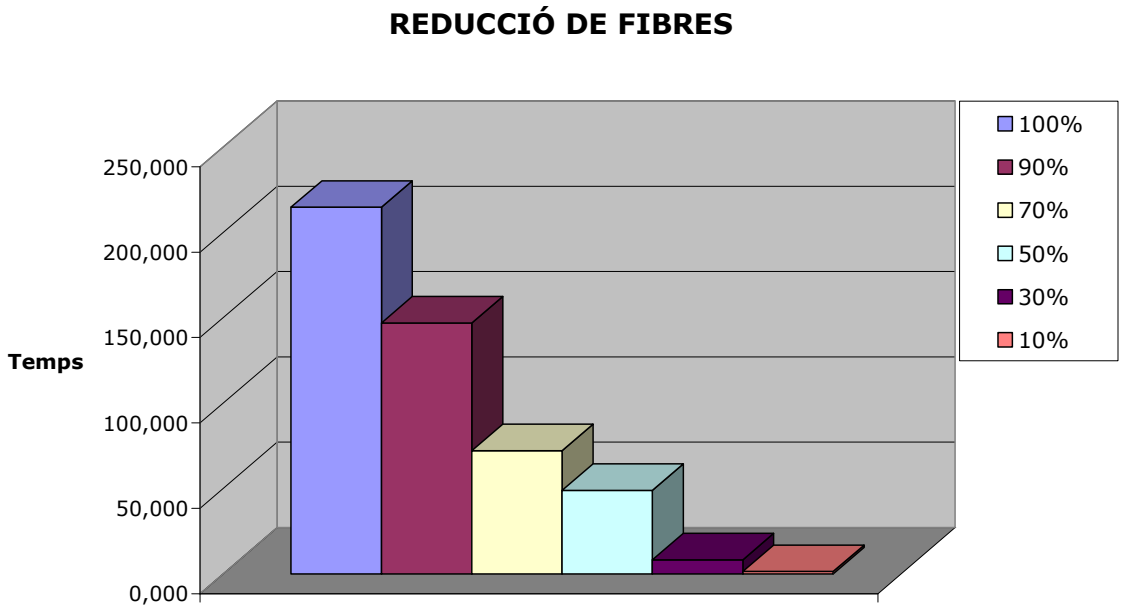
Per fer el clustering utilitzant la tècnica de reducció de punts, emprarem dos vectors de fibres. En el primer hi afegirem el conjunt de les noves fibres reduïdes i el segon serà el de les fibres originals. El nou vector el farem servir durant tot l'algorisme per reduir el cost en temps, mentre que el vector original el guardarem per construir l'escena una vegada hagi finalitzat l'algorisme d'agrupament.

Les proves han estat fetes utilitzant el fuzzy c-means com a algorisme de clustering i la mitjana dels punts més propers com a mètode de distància. Agafem el c-means per ser el que millor resultats ha donat en les proves i el que ha trigat menys.

		PROVES			
		1	**2**	**3**	**Mitjana**
NÚMERO DE PUNTS	**100%**	266,391	240,562	137,469	**214,807**
	90%	208,078	93,703	139,203	**146,995**
	70%	54,735	82,796	78,930	**72,154**
	50%	50,157	53,875	42,813	**48,948**
	30%	7,188	8,844	8,407	**8,146**
	10%	1,516	0,813	1,656	**1,328**

***TAULA 6**. Temps d'execució de l'algorisme de clustering c-means per a fibres amb diferent nombre de punts..*

A la *taula 6* es mostra el resultat d'aplicar l'algorisme agafant els mateixos percentatges per als que hem provat el traçat de fibres. Per a cada percentatge hem fet tres execucions i hem calculat la mitjana per a obtenir un resultat més precís. Els temps estan expressats en segons i es pot veure com disminueix a mesura que agafem una menor quantitat de punts. Amb el *gràfic 1* s'aprecia de forma més clara aquest guany. La barra que fa referència al 10% dels punts és insignificant si la comparem amb la barra que indica el 100% dels punts.



***GRÀFIC 1**. Temps d'execució de l'algorisme de clustering c-means per a fibres amb diferent nombre de punts..*

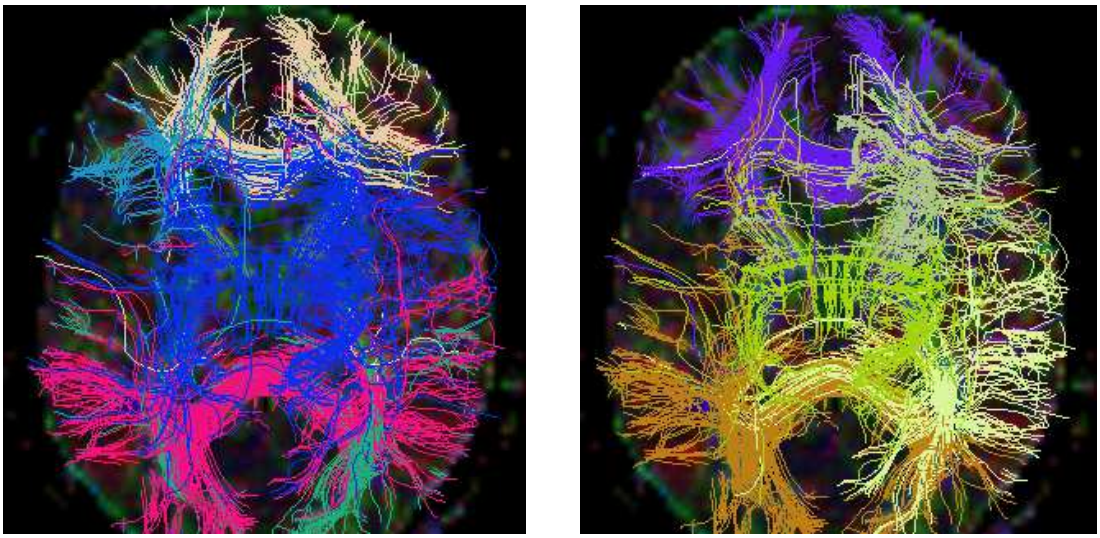


***FIGURA 82**.Resultats d'aplicar clustering sobre un model real amb les fibres escurçades al 30% (esquerra) i al 10% (dreta). Pla XZ*

La figura 82 correspon a les captures del pla XZ utilitzant el 30% i el 10% de les fibres, respectivament, fent servir els mateixos paràmetres que s'han utilitzat al provar el mètode fuzzy c-means sobre models reals. Podem considerar que el resultat és satisfactori ja que la sortida és similar. Com ja sabem, no serà idèntica per la gran dependència que té el mètode de la inicialització.

4.5.- Resultats dels temps d'execució

En aquest apartat es pretén analitzar el comportament de cadascun dels mètodes d'agrupament en funció de cada mètode de distància, sobre tots els models. La finalitat és estudiar-ne el correctesa tenint en compte el factor temps i la correctesa de l'agrupament.
Primer estudiarem els resultats sobre models reals i llavors, havent ja fet una primera selecció, sobre els models reals.

4.5.1.- Resultats sobre models sintètics

Totes les taules d'aquest apartat han estat realitzades seguint la mateixa estructura. La primera columna és el model sintètic (*Veure apartat 4.3.1.1*). La segona columna fa referència al número de prova. Per cada model i mètode s'ha realitzat tres proves, que són resultat d'una mateixa execució, per treure un resultat més fiable. El resultat que es mostra en negreta és la mitjana dels temps de les 3 proves. Al costat dels temps (en segons) es veu la correctesa. La resta de columnes són els diferents mètodes.

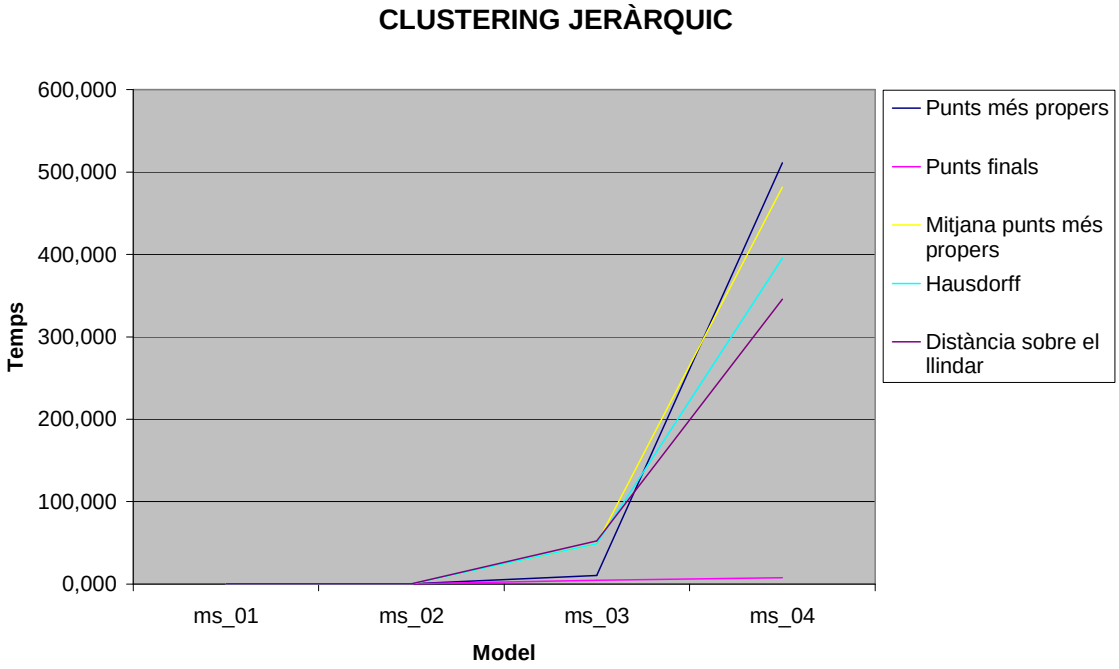
CLUSTERING JERÀRQUIC

		MÈTODE DE DISTÀNCIA									
MODEL	PROVA	Punts més propers		Punts finals		Mitjana punts més propers		Hausdorff		Distància sobre el llindar	
ms_01	1	0,375	ok	0,203	ok	0,390	ok	0,391	ok	0,328	ok
	2	0,297	ok	0,141	ok	0,328	ok	0,313	ok	0,281	ok
	3	0,312	ok	0,172	ok	0,329	ok	0,528	ok	0,315	ok
		0,328	**ok**	**0,172**	**ok**	**0,349**	**ok**	**0,411**	**ok**	**0,308**	**ok**
ms_02	1	0,141	x	0,157	ok	0,282	ok	0,172	ok	0,157	x
	2	0,282	x	0,125	ok	0,141	ok	0,157	ok	0,110	x
	3	0,140	x	0,125	ok	0,250	ok	0,141	ok	0,141	x
		0,188	**x**	**0,136**	**ok**	**0,224**	**ok**	**0,157**	**ok**	**0,136**	**x**
ms_03	1	10,450	x	4,750	ok	48,203	ok	50,031	ok	48,406	ok
	2	10,260	x	4,671	ok	51,172	ok	51,812	ok	53,781	ok
	3	10,922	x	4,734	ok	47,922	ok	45,484	ok	54,719	ok
		10,544	**x**	**4,718**	**ok**	**49,099**	**ok**	**49,109**	**ok**	**52,302**	**ok**
ms_04	1	532,016	ok	7,812	ok	424,109	ok	372,204	ok	343,453	ok
	2	501,734	ok	7,406	ok	523,188	ok	417,828	ok	351,734	ok
	3	499,192	ok	7,919	ok	497,922	ok	395,750	ok	342,625	ok
		510,981	**ok**	**7,712**	**ok**	**481,740**	**ok**	**395,261**	**ok**	**345,937**	**ok**

***TAULA 7**. Temps d'execució i correctesa de l'algorisme de clustering jeràrquic.*

A la *taula 7* podem observar els temps, expressats en segons, i la correctesa resultants d'executar l'algorisme de clustering jeràrquic.

Per a veure la tendència de cada mètode de distància en cada model es pot observar la *gràfica 2* que ens dóna una idea general.



***GRÀFIC 2**. Temps d'execució de l'algorisme de clustering jeràrquic.*

Veiem que per als models ms_01 i ms_02, que són els de menys fibres, l'algorisme s'executa instantàniament, per a qualsevol dels mètodes de distància escollits. És en els models ms_03 i ms_04 en els que la complexitat és més elevada, on els temps es disparen. En el cas del model ms_03 si escollim la distància dels punts finals o la dels punts més propers els temps encara són força baixos.

Seguint les sèries (línies) per separat:

- Punts més propers: Per al model ms_03 el temps és baix, però llavors es dispara en el ms_04, sent el mètode que fa que l'algorisme de clustering trigui més. Pel que fa a la correctesa veiem a la taula que ha fallat en els models 2 i 3, que són aquells on s'encreuen les fibres.
- Punts finals: Es pot considerar que aquest mètode té un pendent constant, que no es dispara, i que és el que en el model de més fibres aconsegueix un temps per sota dels 10 segons. A més, mirant la taula, veiem que totes les sortides han estat les desitjades. Seria el millor mètode doncs, però l'haurem de descartar per el que ja s'havia explicat anteriorment, els models reals no són tan "perfectes" com els sintètics i les bandes de fibres no estan tan clarament diferenciades.

- Mitjana dels punts més propers: Als models 3 i 4 treu temps més elevats, però sempre correctes.
- Hausdorff: Té un comportament similar al mètode de la mitjana dels punts més propers, sent el temps per al model 4 una mica inferior.
- Distància sobre el llindar: Es comporta com els dos anterior, tot i que veiem que la sortida no ha estat l'esperada en el model 2. El problema d'aquest mètode és la tria del llindar.

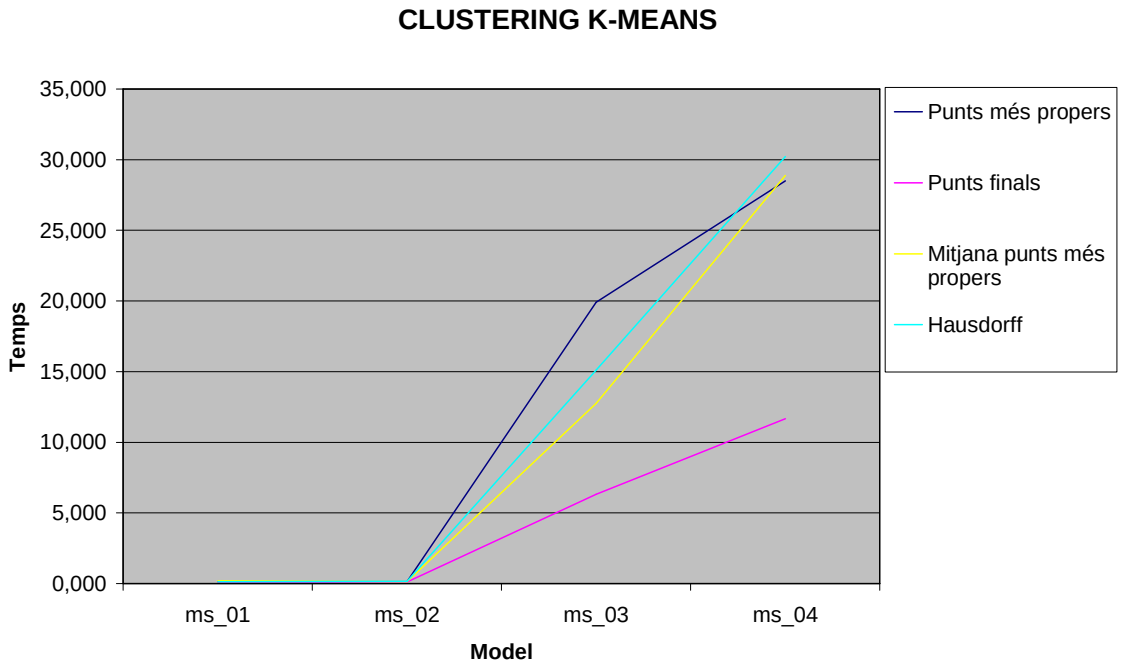
A continuació la *taula 8* mostra els temps d'execució i correctesa de l'algorisme de clustering k-means.

CLUSTERING K-MEANS

		MÈTODE DE DISTÀNCIA							
MODEL	**PROVA**	Punts més propers		Punts finals		Mitjana punts més propers		Hausdorff	
ms_01	1	0,140	x	0,172	ok	0,190	ok	0,141	ok
	2	0,156	x	0,125	ok	0,213	ok	0,125	ok
	3	0,151	x	0,125	ok	0,198	x	0,143	ok
		0,149	**x**	**0,141**	**ok**	**0,200**	**ok**	**0,136**	**ok**
ms_02	1	0,125	x	0,125	ok	0,156	ok	0,157	ok
	2	0,134	x	0,125	ok	0,125	x	0,125	ok
	3	0,156	ok	0,125	ok	0,125	ok	0,250	ok
		0,138	**x**	**0,125**	**ok**	**0,135**	**ok**	**0,177**	**ok**
ms_03	1	21,766	x	6,891	ok	13,735	ok	12,719	ok
	2	18,627	x	5,953	ok	12,172	ok	19,797	ok
	3	19,355	x	6,141	ok	12,344	ok	12,813	ok
		19,916	**x**	**6,328**	**ok**	**12,750**	**ok**	**15,110**	**ok**
ms_04	1	28,547	ok	12,782	ok	29,640	ok	29,547	ok
	2	28,453	ok	8,375	ok	28,563	ok	30,547	ok
	3	28,485	ok	13,860	ok	28,422	ok	30,578	ok
		28,495	**ok**	**11,672**	**ok**	**28,875**	**ok**	**30,224**	**ok**

***TAULA 8**. Temps d'execució i correctesa de l'algorisme de clustering k-means*

A la *taula 8* veiem els resultats de l'algorisme k-means. A primera vista podem notar que hi ha una columna menys, que és la que feia referència al mètode de distància sobre el llindar. L'hem tret ja que el comportament era molt similar a la mitjana de punts més propers i a la distància de Hausdorff, i tenia la particularitat d'haver de triar un valor llindar per a cada execució, fet que complica les coses.
A diferència de la taula X, en aquesta veiem que una prova sobre un mateix model i amb el mateix mètode de distància pot anar bé o malament. La explicació està en el mètode de clustering; el k-means depèn de la inicialització dels clústers mentre que el jeràrquic, no.



***GRÀFIC 3**. Temps d'execució de l'algorisme de clustering k-means*

Al *gràfic 3* veiem la tendència general del k-means. Per als dos primers models succeeix el mateix que amb el clustering jeràrquic. La tendència que segueixen les línies en els model 3 i 4 també és semblant, sent la distància dels punts finals la que aconsegueix els temps menors. Cal notar que l'escala de temps és més reduïda, perquè en aquest algorisme els temps més elevats es situen sobre els 30 segons, mentre que en el jeràrquic eren de 6-7 minuts.
Finalment, dir que el mètode dels punts més propers també ha donat resultats que no s'ajusten als que busquem.

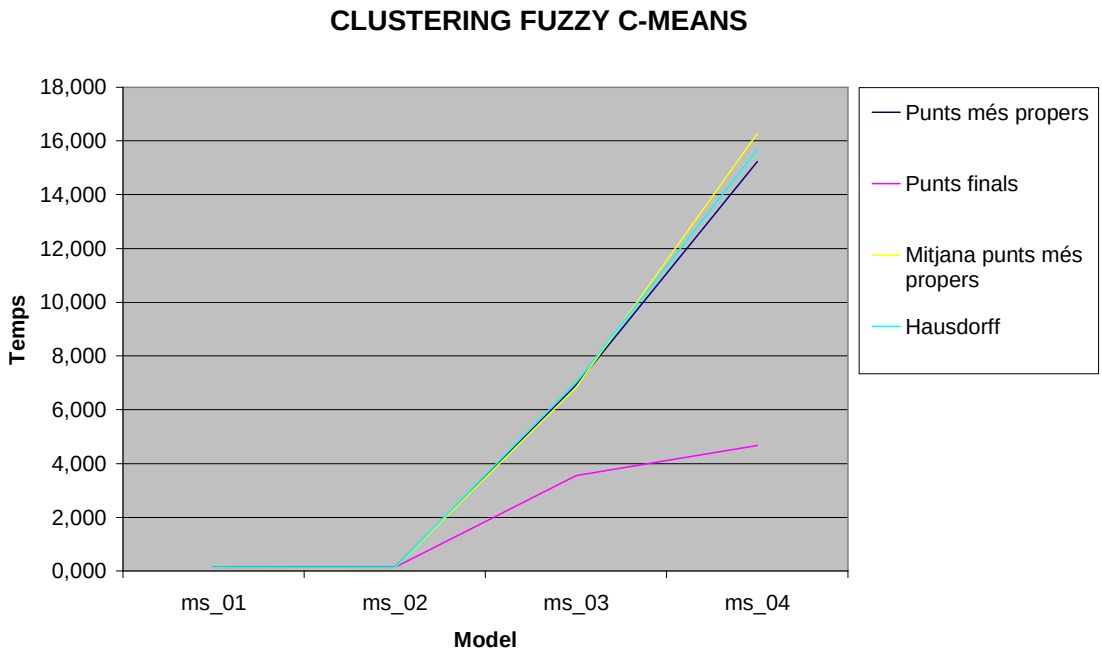
CLUSTERING FUZZY C-MEANS

		MÈTODE DE DISTÀNCIA							
MODEL	PROVA	Punts més propers		Punts finals		Mitjana punts més propers		Hausdorff	
ms_01	1	0,172	ok	0,141	ok	0,156	ok	0,172	ok
	2	0,141	x	0,125	ok	0,141	ok	0,140	ok
	3	0,157	ok	0,141	ok	0,125	ok	0,156	ok
		0,157	**ok**	**0,136**	**ok**	**0,141**	**ok**	**0,156**	**ok**
ms_02	1	0,140	x	0,156	ok	0,141	ok	0,156	ok
	2	0,141	x	0,141	ok	0,109	ok	0,141	ok
	3	0,125	x	0,125	ok	0,109	ok	0,125	ok
		0,135	**x**	**0,141**	**ok**	**0,120**	**ok**	**0,141**	**ok**
ms_03	1	7,156	x	4,031	ok	7,187	ok	7,000	ok
	2	6,921	x	3,328	ok	6,656	ok	6,859	ok
	3	6,703	x	3,297	ok	6,687	ok	7,235	ok
		6,927	**x**	**3,552**	**ok**	**6,843**	**ok**	**7,031**	**ok**
ms_04	1	15,782	ok	4,657	ok	16,922	ok	15,406	ok
	2	15,250	ok	4,859	ok	16,437	ok	16,250	ok
	3	14,657	ok	4,485	ok	15,500	ok	15,343	ok
		15,230	**ok**	**4,667**	**ok**	**16,286**	**ok**	**15,666**	**ok**

***TAULA 9**. Temps d'execució i correctesa de l'algorisme de clustering fuzzy c-means*

Finalment, hem seguit el mateix procediment per a l'algorisme de clustering fuzzy c-means, proporcionant la *taula 9* i el *gràfic 4*, que ens informen sobre la correctesa i els temps d'execució numèricament i gràficament.
Tots els mètodes de distància han donat un resultat correcte, exceptuant el dels punts més propers, que ha fallat completament en els models ms_02 i ms_03, i en una ocasió en el model ms_01.

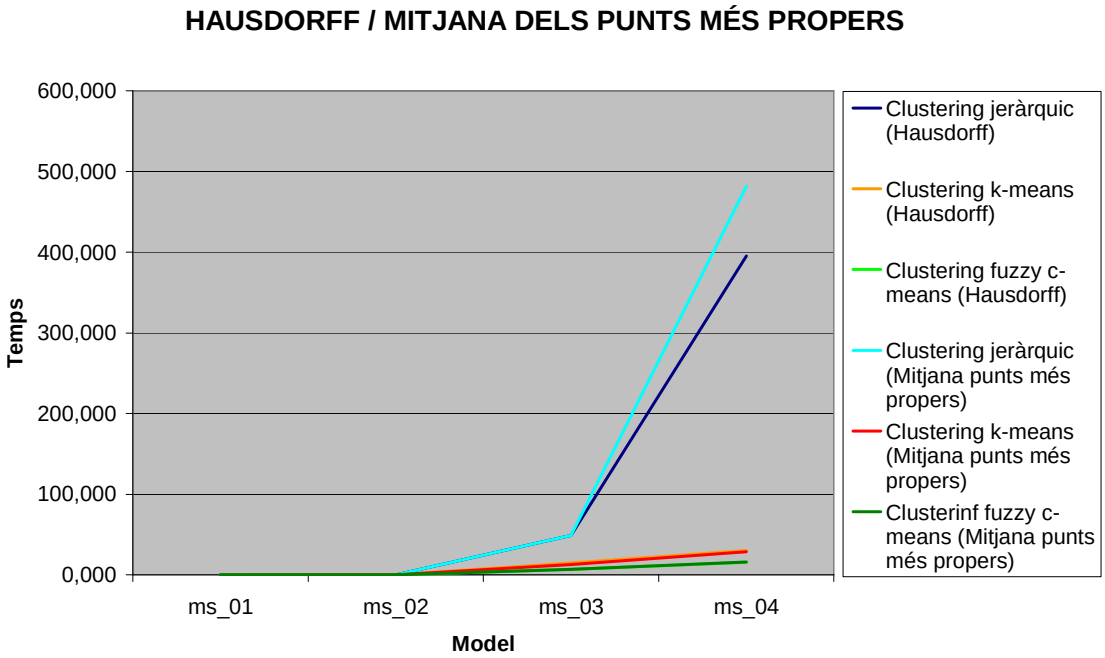
Si observem el gràfic, apreciem que es manté la tendència. Per als dos models de poques fibres és instantani i a mesura que aquest augmenta, els temps creixent de forma no lineal.
El mètode dels punts finals segueix sent el menys costós i els tres restants continuen donant resultats molt semblants.



***GRÀFIC 4**. Temps d'execució de l'algorisme de clustering fuzzy c-means*

Hem de destacar l'interval de temps en que es mou aquest algorisme, que és menor que en el k-means, el qual al seu torn ja ho era molt més que el jeràrquic. Els temps més elevats estan al voltant dels 16 segons en aquest cas.

Per apreciar millor aquesta diferència de temps entre algorismes d'agrupament, s'ha creat el *gràfic 5*.



***GRÀFIC 5**. Temps d'execució dels diferents algorismes de clustering utilitzant les distàncies de Hausdorff i la mitjana dels punts més propers.*

Hi ha dues línies per a cada algorisme d'agrupament, una per a Hausdorff i l'altra per a la mitjana dels punts més propers.
Les sèries blaves fan referència a l'algorisme jeràrquic; les vermelles, al k-means i les verdes, al fuzzy c-means.
Les línies que fan referència al k-means queden una sobre l'altre. El mateix passa amb les que simbolitzen el fuzzy c-means. Això vol dir que els temps són quasi els mateixos tan si utilitzem Hausdorff com la mitjana dels punts més propers.
En el cas del jeràrquic si que les podem diferenciar i veiem que és menys costós utilitzar Hausdorff en aquest algorisme.
La informació més rellevant que ens proporciona el gràfic és la clara diferència entre temps d'execució segons l'algorisme que emprem: el més eficient és el fuzzy c-means, seguit molt de prop pel k-means. Podríem descartar el jeràrquic, per la evident diferència de temps que es pot observar.

4.5.2.- Conclusions (models sintètics)

Després de recopilar tota aquesta informació, podem fer una tria tant d'algorismes d'agrupament com de mètodes de distància per a una posterior prova dels temps d'execució en models reals. No tindria sentit tornar a fer proves amb mètodes que no van bé per a les nostres dades.
Agafarem, doncs, els següents mètodes de distància:

- Mitjana dels punts més propers
- Distància de Hausdorff

I els següents mètodes d'agrupament de fibres:

- K-means
- Fuzzy c-means

4.5.3.- Resultats sobre models reals

En aquest apartat el que hem fet ha estat aplicar clustering sobre els models reals, únicament fent proves amb els algorismes k-means i c-means i prenent com a mesures de distància la mitjana dels punts més propers i la distància de Hausdorff.

Observem els resultats a la *taula 10*.

		K-MEANS		FUZZY C-MEANS	
		Hausdorff	**Mitjana punts més propers**	**Hausdorff**	**Mitjana punts més propers**
PROVES	**1**	-	-	2,969	1,516
	2	-	-	1,094	0,813
	3	-	-	0,922	1,656
	Mitjana	**> 10 minuts**	**> 10 minuts**	**1,662**	**1,328**

***TAULA 10**. Temps d'execució dels algorismes k-means i fuzzy c-means amb fibres escurçades*

Com en les taules anteriors els resultats numèrics, fan referència als temps d'execució, expressats en segons i s'han fet 3 proves per a cada mètode d'agrupament i cada mesura de distància per treure'n una mitjana.

El fuzzy c-means ha tingut uns resultats molt bons tant quan l'hem executat utilitzant Hausdorff com quan hem utilitzat la mitjana dels punts més propers. Els temps han estat molt semblants; la diferència de les mitjanes és menor que mig minut.
Per contra, el k-means, ha donat resultats molt negatius. Hem parat l'execució al veure que el temps era superior als 10 minuts. S'ha comportat de la mateixa manera amb les dues mesures de distància.

4.5.4.- Conclusions (models reals)

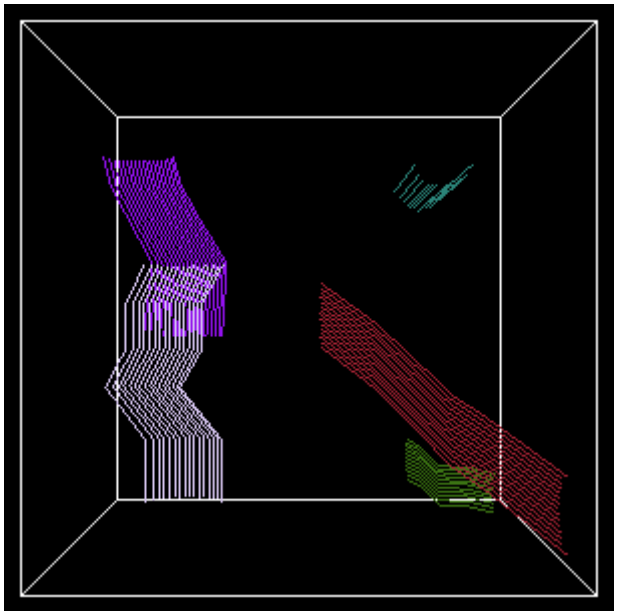
La nostra tria serà, doncs, l'algorisme fuzzy c-means amb qualsevol de les dues mesures de distància provades.
Cal notar la gran diferència que hi ha entre utilitzar el k-means amb models sintètics o amb models reals. Aquesta diferència no ha estat tan pronunciada en el cas del fuzzy c-means.

4.6.- Implementació de la visualització de clústers

En aquest apartat s'explica la implementació de la visualització de clústers un cop ja s'ha fet el tracking de fibres i el clustering d'aquestes agrupant-les en bandes. Hem fet dues implementacions. La primera consistent en mostrar les fibres una a una i la segona en mostrar les bandes com una única part.

4.6.1.- Visualització de clústers fibra a fibra

En aquesta primera implementació es mostren les fibres una a una de tot el model, distingint-les en diferents colors segons la banda a la qual pertanyin. Tindrem, doncs, tants colors com clústers hagin resultat de l'algorisme d'agrupament.
Per construir l'escena començarem recorrent el vector de clústers. Per a cada clúster crearem una aparença on hi definirem el color. Per definir el color s'han generat tres valors aleatoris, un per a la quantitat de vermell, un per a la de verd, i l'altre per a la de blau.
Un cop tenim el color definit, recorrem tota la fibra guardant cadascun dels seus punts per a després fer-ne un LineStripArray, és a dir, una línia. I a aquesta línia li apliquem l'aparença que havíem definit abans. Repetirem el procés per a totes les línies afegint-les com a filles d'un branchgroup. Aquest branchgroup és el que retornarà el mètode, que permetrà construir l'escena.



La *figura 83* mostra la visualització fibra a fibra un cop s'ha fet el procés de clustering. En aquest cas,el model escollit ha estat el ms_04 i l'algorisme el k-means.
Veiem que cadascun dels 5 clústers resultants tenen un color diferent i a més podem distingir-ne les línies que representen cada fibra del model.

FIGURA 83.
Visualització de clústers fibra a fibra.

A continuació el *diagrama de seqüència 11* ens mostra el mètode.

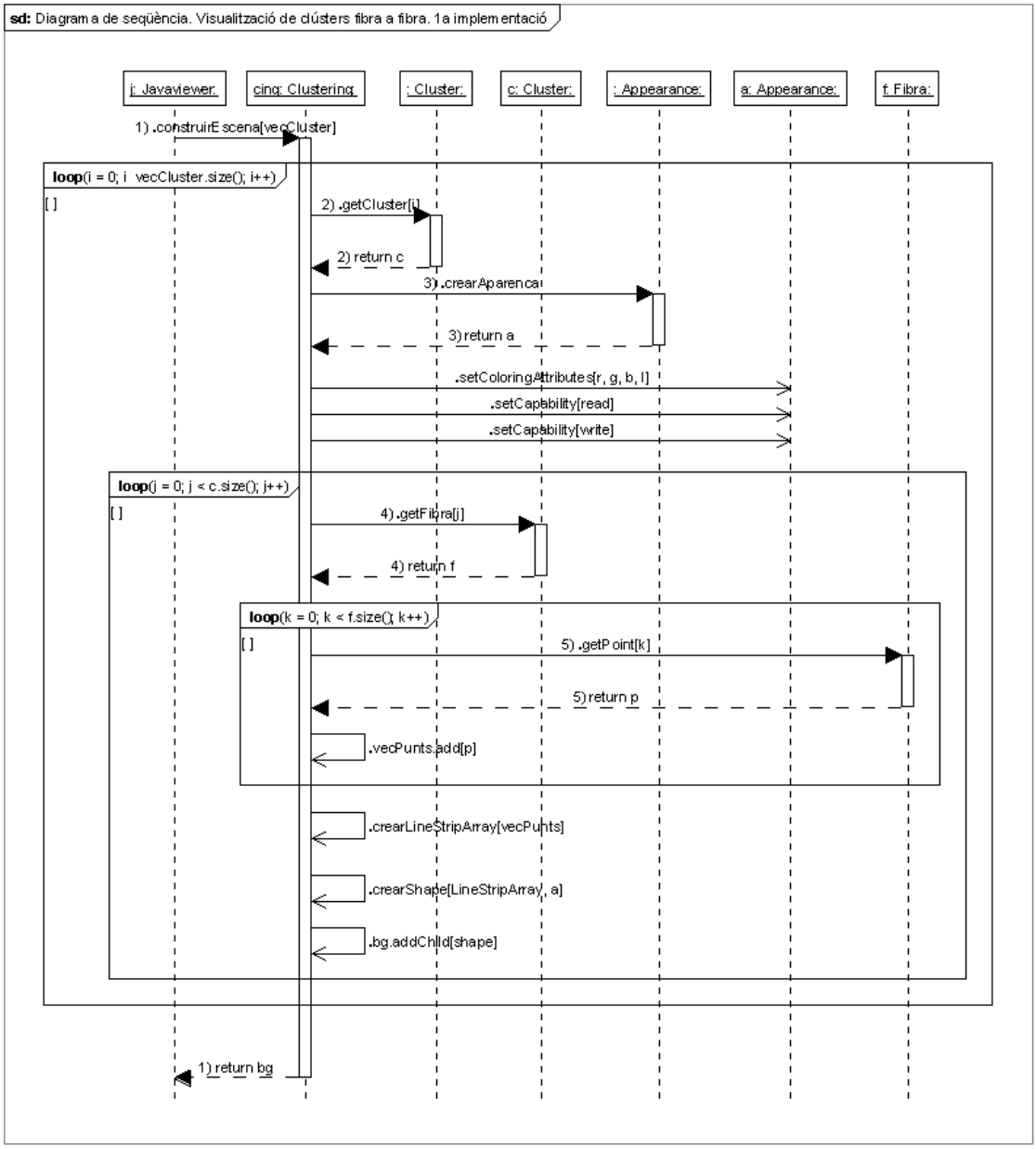


Diagrama de seqüència 11. *Visualització de clústers fibra a fibra. 1a implementació.*

Com ja hem explicat al principi d'aquest apartat, els colors són generats a partir de tres valors aleatoris. El problema de l'atzar és que tan se'ns poden generar colors molt diferents com generar colors molt semblants o, fins i tot, els mateixos. En el cas de que els colors siguin molt propers podem no distingir clarament els grups de fibres, cosa que seria una contradicció, perquè precisament el principal objectiu d'utilitzar tècniques d'agrupament és poder veure amb la major claredat possible els clústers de que disposa el model.

A la figura 84 es veu un exemple, on la generació dels colors no és la desitjada.

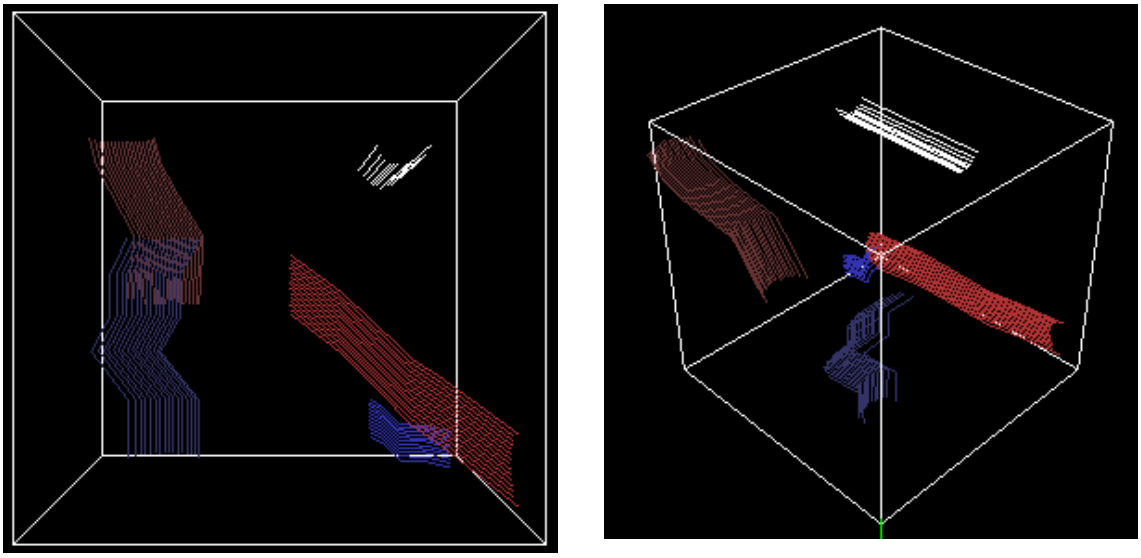


FIGURA 84. *Visualització de clústers fibra a fibra amb colors confusos. A l'esquerra pla YZ. A la dreta vista 3D. Model ms_04.*

Dels 5 clústers, el pintat de color blanc es distingeix clarament. Però dels altres 4, n'hi ha dos amb una tonalitat semblant de vermell i els altres dos amb una tonalitat semblant de blau. I, en aquest cas, encara són distingibles, però si ens imaginem un nombre molt elevat de fibres i totes juntes segurament no apreciarem la diferència. Això seria el que passaria en un model real.

Proposem, doncs, tenir colors preestablerts. Tindrem 10 colors ja definits, perquè en principi si l'algorisme funciona correctament, no n'hi hauria d'haver més. En tot cas, si el número de clústers resultants, fos superior a 10, llavors, la resta, es generarien de forma aleatòria.
Agafarem 10 colors que siguin perfectament diferenciables.
Una consideració a tenir en compte és que no podem agafar com a color el negre, ja que és el fons de pantalla com hem pogut veure en totes les captures que s'han anat fent al llarg d'aquest projecte. Així descartem directament el negre i també qualsevol color que sigui massa fosc.

Color	Nom	Valor Hexadecimal	Valor RGB	Valor [0,1]
	Blanc	FF FF FF	255 - 255 - 255	1.0 - 1.0 - 1.0
	Vermell	FF 00 00	255 - 0 - 0	1.0 - 0.0 - 0.0
	Verd	00 80 00	0 - 128 - 0	0.0 - 0.5 - 0.0
	Blau	00 00 FF	0 - 0 - 255	0.0 - 0.0 - 1.0
	Groc	FF FF 00	255 - 255 - 0	1.0 - 1.0 - 0.0
	Taronja fosc	FF 8C 00	255 - 140 - 0	1.0 - 0.55 - 0.0
	Magenta	FF 00 FF	255 - 0 - 255	1.0 - 0.0 - 1.0
	Cian	00 FF FF	0 - 255 - 255	0.0 - 1.0 - 1.0
	Lima	00 FF 00	0 - 255 - 0	0.0 - 1.0 - 0.0
	Lila	80 00 80	128 - 0 - 128	0.5 - 0.0 - 0.5

TAULA 11. *Colors predefinits que fem servir per pintar els clústers.*

La *taula 11* mostra el color, el nom, el valor hexadecimal, el valor RGB i el valor entre 0 i 1 dels colors que hem escollit com a predefinits per a pintar els clústers. Són colors bàsics que no presenten confusió.
Els colors se solen definir en valors hexadecimals i/o valors RGB. Nosaltres hem calculat el valor entre 0 i 1 al que correspon, ja que el mètode que assigna colors a una aparença defineix els seus paràmetres com a *floats* entre 0.0 i 1.0, on 0 és l'absència de color i 1 la màxima quantitat de color.

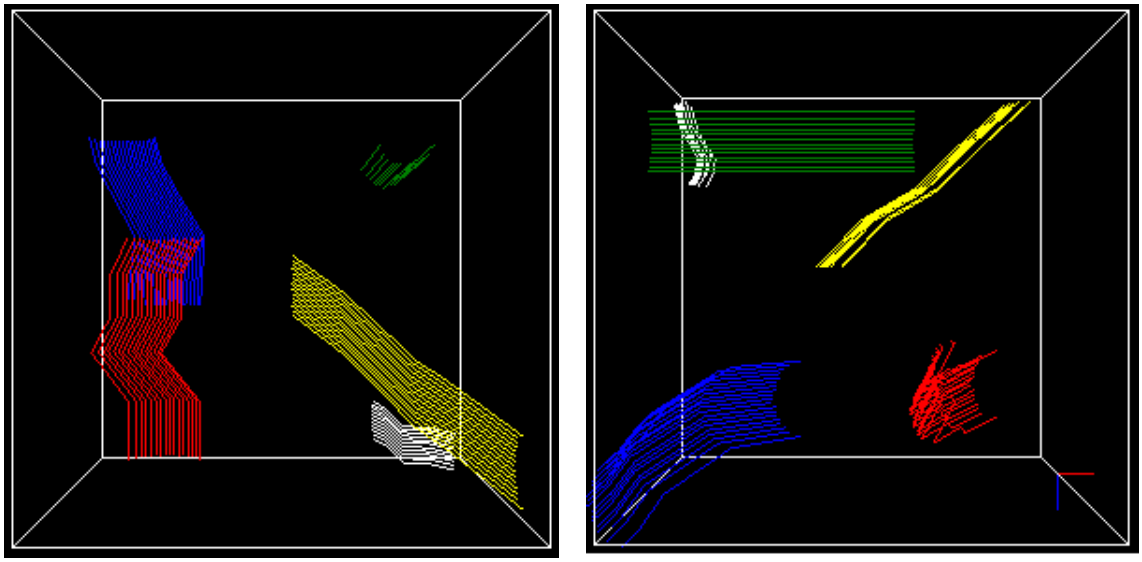


FIGURA 85. *Visualització de clústers fibra a fibra utilitzant colors predefinits.*

A la figura 85 veiem que el resultat és l'esperat. Com que hi ha 5 clústers ha agafat els 5 primers colors que tenim definits. Si mirem la taula podrem comprovar que són el blanc, el vermell, el verd, el blau i el groc.
La sortida que tenim ara és molt més clara que no pas quan generàvem els colors de forma aleatòria.

Hem escollit el model ms_04 perquè és el que té més bandes. Si el model només té dues bandes és probable que els colors generats aleatòriament siguin força diferents, però, com és lògic, quan més grups tinguem, més colors haurà de generar i més possibilitats hi haurà de que surtin colors propers.

4.6.2.- Visualització de clústers per bandes

Partim de la situació en que tenim ja el clustering fet i, per tant, les fibres estan agrupades en bandes o clústers. Fins ara, totes les fibres que formaven part d'una banda concreta tenien el mateix color i es visualitzen totes elles.

L'objectiu que volem aconseguir és no veure totes les fibres que formen una banda, sinó veure l'espai que ocupen com si fos un únic cos. En comptes de veure 100 fibres de color vermell, volem un sol cos de color vermell.
L'avantatge serà una visualització molt més clara. No necessitem saber informació de cada fibra, sinó del conjunt, per això les agrupem en bandes.

Com que el que volem és veure les fibres com un sol espai, podem pensar en fer un tub. Com que el cost en memòria de Java 3D és molt alt, el que farem serà simular un tub. Per a això, farem servir una malla de triangles que agruparà totes les fibres. Llavors li posarem unes tapes (el·lipses) als extrems i li donarem una textura que consistirà en un degradat per a donar sensació de profunditat.
Es defineix una malla de triangles com un conjunt connex de triangles. Si el nombre de triangles és prou gran, les malles poden representar amb suficient aproximació objectes corbats o cilíndrics, superfícies de Bézier, NURBs, etc.

Per a fer la malla de triangles utilitzarem el triangle strip, que permet dibuixar una cadena de triangles units per algun dels seus costats. També ho podríem fer dibuixant els triangles un a un, però això tindria un cost més elevat ja que enviaríem molts més vèrtexs, cosa que amb el triangle strip ens estalviem.
Hi ha diferents casos: tenir una sola fibra, tenir un conjunt de fibres, tenir un conjunt de fibres en forma de U…

4.6.2.1.- Implementació simple

Per veure el funcionament de la malla de triangles, hem començat creant un interfície molt simple per a realitzar-hi alguna prova prèvia.
Hem realitzat primer una prova a partir d'una sola fibra i llavors amb un conjunt de fibres, que serà el que tindrem en els models reals.

- Fibra única:

Per a fer la malla de triangles, necessitem punts exteriors a la fibra. Per cadascun dels punts de la fibra li apliquem un desplaçament positiu i un de negatiu.

Podem topar amb el següent problema. Si tenim una fibra totalment vertical, podem pensar que el desplaçament haurà de ser en la component x del punt. Però si després ens apareix una fibra totalment horitzontal, ja no ens servirà aquest desplaçament en la component x, sinó que hauria de ser en la component y del punt. Com que tenim casos diferents, hem de fer una regla general.

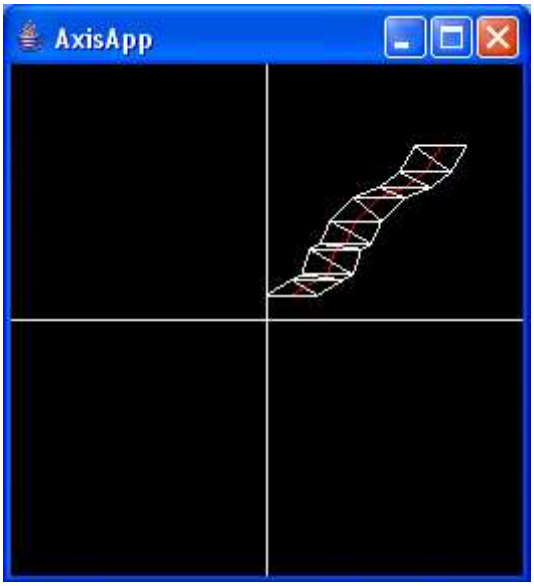


FIGURA 86.
Malla de triangles d'una sola fibra.

Haurem doncs, de mirar quina és la variació de cadascuna de les components entre punt i punt de la fibra. Agafarem les dues components que menys variïn per a fer el desplaçament. Farem el desplaçament inversament proporcional a la variació d'aquestes dues components. Exemple: si les dues components que menys varien són X i Y i la variació d'aquestes entre dos punts és de 1 i 2 respectivament, el desplaçament que haurem de fer correspondrà a 2/3 per a X i 1/3 per a Y. Això servirà per a que les línies exteriors de la malla de triangles ens quedin paral·leles a la fibra.

El resultat el podem veure a la *figura 86* on la línia vermella representa la fibra de la qual disposàvem i es veuen dues línies blanques paral·leles a aquesta que s'han trobat aplicant-hi un desplaçament. A partir dels punts de les línies blanques, s'ha creat la malla de triangles.

- Varies fibres:

Si tenim varies fibres, hem de trobar quines són les exteriors per a fer la malla de triangles.
Agafem, doncs, tots els punts finals de cadascuna de les fibres. Creem 3 arrays d'objectes que tinguin com a atributs un real i un enter. El real per la component i l'enter per a saber a quina fibra pertany. Cadascun dels arrays guardarà valors de X, de Y o de Z. Les fibres també estaran en un array. D'aquesta manera podrem saber a quina fibra pertany la x més gran, per exemple.
Un cop tenim els arrays fets, els ordenarem en funció del número real i així podrem veure la diferència entre el valor menor i el major. Els dos rangs amb major variació ens determinaran el pla en el que ens trobem. Per a dibuixar l'el·lipse ens falta saber el radi major i menor. Agafarem els valors més alt i més baix de les dues components que ens determinen el pla en que treballem. La diferència més gran entre el valor més

alt i el més baix ens indicarà el radi major i la més petita, el radi menor. Així, ja podem dibuixar l'el·lipse, i a més sabem que els valors extrems del radi major de l'el·lipse són els punts finals de les fibres extremes. Un cop tenim aquesta informació, ja podem dibuixar la malla de triangles. En aquest cas no hem de buscar punts com quan només teníem una fibra, perquè farem servir els punts de les dues fibres exteriors.
La *figura 87* mostra tres línies vermelles que representen tres fibres que ens són donades i que pertanyen a un mateix clúster. A partir de les dues exteriors es crea la malla de triangles (en blanc).
La *figura 88* és igual a la *figura 87* però amb una el·lipse a la part superior.

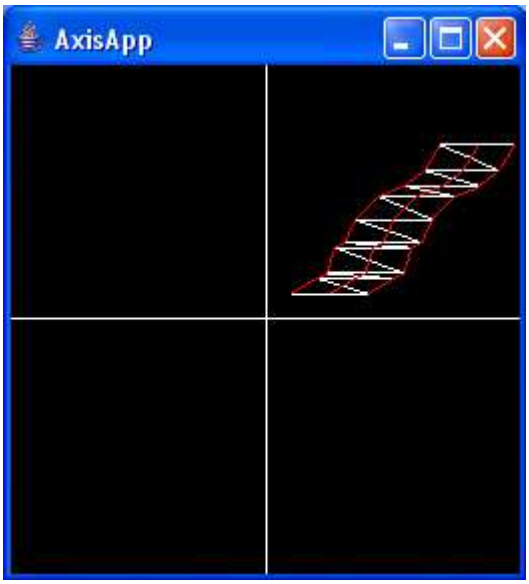


FIGURA 87. *Malla de triangles de varies fibres*

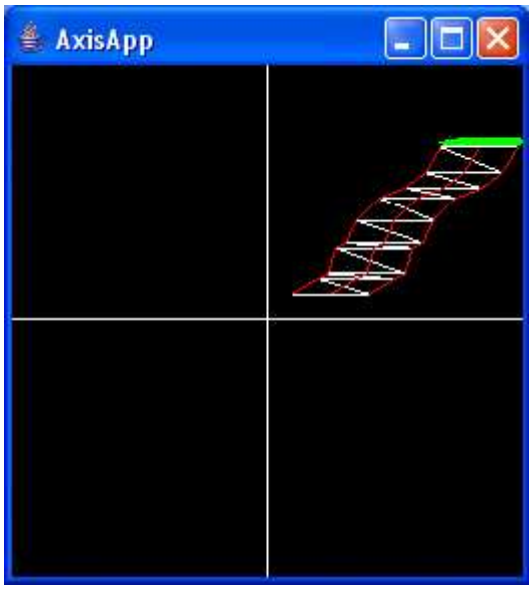


FIGURA 88. *Malla de triangles de varies fibres amb el·lipse*

4.6.2.2.- Implementació sobre models sintètics

Ara que ja tenim una mica més clara la idea, gràcies a l'apartat anterior, provarem de fer el mateix, però amb els models sintètics amb els que hem estat treballant durant tot el projecte.

Agafem el model més senzill, l'ms_01, i li apliquem un algorisme de clustering de manera que ens quedin correctament agrupades les fibres.
Un cop tenim tots els clústers, el primer pas consisteix en buscar la fibra mitjana de cada un d'ells. A la *figura 89* es veuen dos plans de buscar les fibres mitjanes dels clústers.

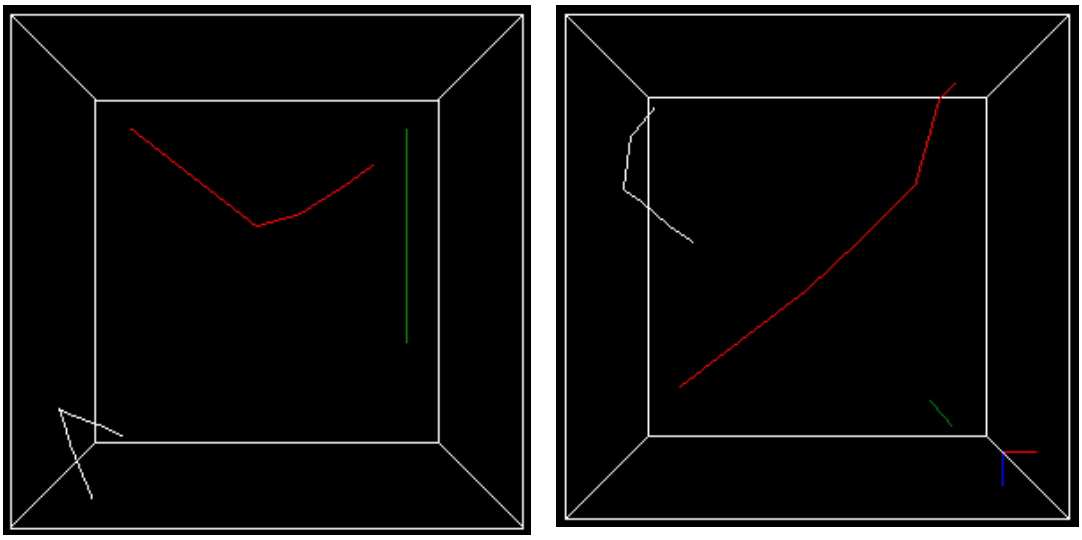


FIGURA 89. *Fibres mitjanes de cadascun dels clústers.*

Llavors, hem de buscar els punts exteriors dels clústers. Podríem pensar en un primer moment en buscar les fibres exteriors, però hem de tenir en compte que si les fibres s'entrecreuen, els punts exteriors seran els de varies fibres.

Per a buscar els punts exteriors farem el següent: Per a cada punt de la fibra mitjana mirem quin són els punts més propers amb cadascuna de les fibres del clúster. I de tots aquests punts, agafem els més allunyats. D'aquesta manera, un cop haguem repetit el procés per a tots els punts de la fibra mitjana, tindrem els punts exteriors del clúster dels dos costats.

Aquests punts els guardarem en dos arrays, un pels punts exteriors de la dreta i l'altre pels punts exteriors de l'esquerra. Durant tot el projecte, hem estat treballant amb vectors, però ara necessitem l'ús dels arrays perquè la utilització dels trianglestrip així ho demana.

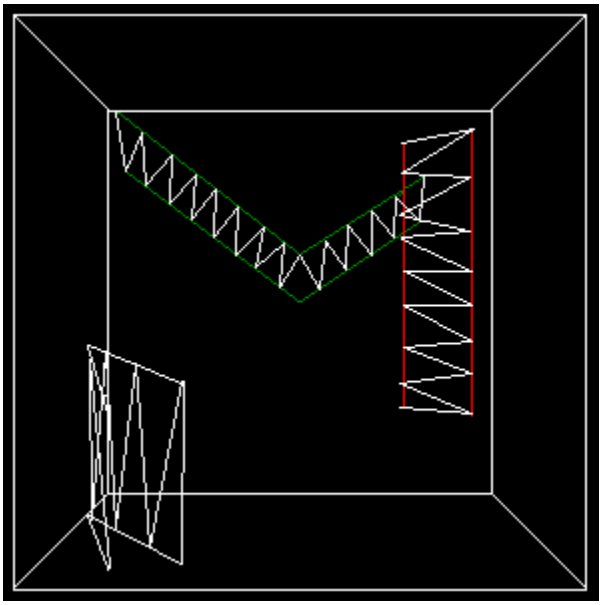


FIGURA 90. *Malla de triangles del model ms_01.*

A la *figura 90* es mostren els clústers amb la malla de triangles, de manera que ja no veiem totes les fibres.

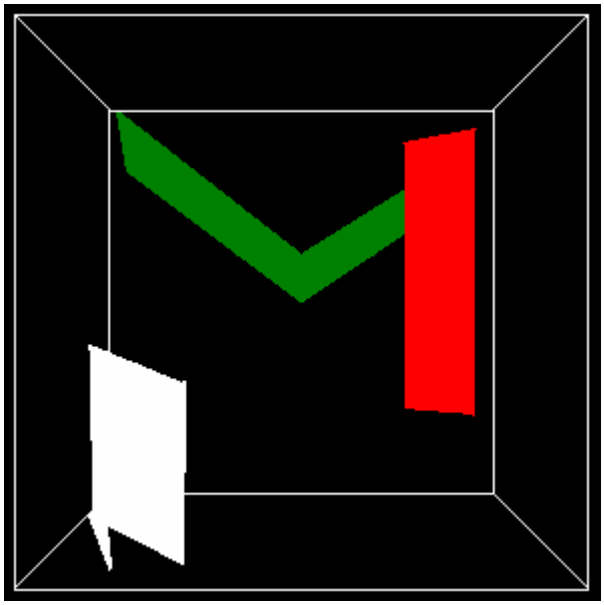


FIGURA 91. *Clústers amb textura*

I un cop feta la malla de triangles hem procedit a posar-li una textura com es veu a la *figura 91*. La textura només ha consistit en un color pla.
El següent pas seria fer que la textura fos un degradat de colors, en comptes, d'un color planer.
Llavors posaríem també unes el·lipses als extrems. D'aquesta manera, aconseguiríem que semblés un tub.

Aquesta idea de pintar els clústers com un bloc té l'inconvenient de haver de estar sempre pendent del punt de vista des del que ens ho mirem. És a dir, si agafem un altre pla (en comptes de l'XY), no veuríem els clústers d'aquesta manera, podríem veure només una fina línia. Per tant no és un tema trivial.

Per evitar aquesta situació podem pintar cada clúster amb una el·lipse a cada extrem i anar pintant també el·lipses a tot el llarg de la fibra mitjana que ja fèiem servir en aquesta idea que hem explicat abans.

5.- CONCLUSIONS

En aquest apartat es fan unes últimes consideracions, tenint en compte el treball fet.
També es fa incís en com es podria continuar, fent millores, sobre aquest projecte.

5.1.- Conclusions generals

A l'inici del projecte ens vam proposar una sèrie d'objectius els quals hem assolit en la seva majoria.

L'objectiu principal d'aquest projecte era desenvolupar diferents funcionalitats que permetessin l'agrupament de fibres i la visualització de clústers.
Els resultats són positius ja que s'han realitzat tots els objectius proposats al començar a treballar en el projecte. A continuació s'exposen els diferents objectius i s'explica de quina manera s'han assolit:

Objectiu: Implementar diferents mètodes de distància

Han estat implementats els següents cinc mètodes de distància:

- Distància dels punts més propers.
- Mitjana de les distàncies dels punts més propers.
- Distància dels punts finals.
- Distància de Hausdorff.
- Distància sobre el llindar.

Hem fet un estudi sobre quins eren els més adequats i quins podíem descartar per no ajustar-se a les nostres necessitats.
Hem testejat els mètodes utilitzant models sintètics i els hem aplicat, llavors, sobre casos reals.

Objectiu: Implementar diferents algorismes de clustering

Han estat implementats els tres següents algorismes de clustering:

- Algorisme de clustering jeràrquic
- Algorisme k-means
- Algorisme fuzzy c-means

Hem fet un estudi sobre quin era el més adequats en el nostre context.
Al igual com en la implementació dels mètodes de distància primer s'ha treballat sobre models sintètics i, més tard, s'ha aplicat en models reals.

Per a triar quins eren els mètodes que més ens convenien, hem fet proves de tots els algorismes de clustering amb tots els mètodes de distància possibles. Per a acceptar

la combinació de mètodes com a vàlida, a part de treure un bon agrupament de fibres, també el temps havia de ser acceptable.

Objectiu: Visualització de clústers

S'han fet dos tipus de visualitzacions:

- Visualització de les fibres una a una. Es veuen totes les fibres i es distingeixen les dels diferents clústers per ser de diferent color. S'han fixat uns colors determinats, per assegurar que es distingeixin de forma clara uns clústers dels altres.

- Visualització de les fibres per bandes. Es veuen els clústers com un únic bloc, en comptes de veure les fibres separadament. S'ha aconseguit utilitzant malles de triangles i textures. Aquesta implementació és molt senzilla i només serveix de guia per saber on hem de continuar en un futur desenvolupament.

Aquest apartat de visualització, és un apartat molt complex, que ja per si sol, podria ser un projecte de fi de carrera.

Objectiu: Integració a la plataforma webDTI

Tot ha estat implementat sobre la plataforma, adaptant-ho a l'estructura ja existent.

Volem remarcar que en aquest projecte no només es tracta d'implementar els mètodes, sinó que la part que ha ocupat gran part del temps ha estat fer moltes proves per a treure unes bones conclusions.

D'altra banda, a mesura que fèiem la implementació dels algorismes d'agrupament, i veient que el temps que trigava en fer-se era molt alt, ha sorgit la idea de minimitzar-lo, reduint el nombre de punts de les fibres. Ens referim a l'**apartat d'optimització del clustering**. Aquest apartat ha estat molt important ja que hem vist una millora considerable.

5.2.- Treball futur

Proposem alguns aspectes que es podrien considerar per a un treball futur:

- Implementar l'algorisme de clustering probabilista i observar-ne el funcionament.
- Seguir amb l'apartat de visualització de clústers per bandes, a partir del punt on ho hem deixat en aquest projecte.

5.3.- Conclusions personals

De forma personal, dir que tot i la complexitat del projecte, és un tema molt interessant ja que estem tractant un òrgan del nostre cos tan important com és el cervell.
He pogut entendre una mica millor com funciona la captació de dades i la forma de treballar amb aquestes. També he après una mica com funciona el Java 3D i la utilització de llibreries gràfiques (swing, awt), que no es toquen massa en el transcurs de la carrera.

Contents amb la feina feta i els resultats que hem obtingut, som conscients que la plataforma es pot ampliar i millorar, però deixem aquests objectius per a un treball futur

6.- Bibliografia

A continuació se citen les referències de les quals s'ha tret la informació.

- Wikipedia

Per a l'estudi dels mètodes de distància i els mètodes de clustering:

The Visualization of Tensor Fields in Biomedicina - Chapter 7
http://club.telepolis.com/ildearanda/combina/combsin_marco.htm
http://home.dei.polimi.it//matteucc/Clustering/tutorial_html/index.html
http://fconyx.ncifcrf.gov/~lukeb/diclust.html
http://galahad.plg.inf.uc3m.es/~docweb/ad/transparencias/apuntesAnalisisDatos.pdf
http://cism.kingston.ac.uk/people/shihab/dissertation.pdf
http://es.wikipedia.org/wiki/Colores_HTML

Streamtubes and streamsurfaces:

http://www.princeton.edu/~asmits/Bicycle_web/streamline.html
http://www.cs.brown.edu/people/dhl/pdf/vis2000_dti.pdf
http://www.cs.brown.edu/people/dhl/pdf/hbp00dti.pdf