Taula de continguts

1. Introducció i objectius ***4***

1.1 Introducció 5

1.2 Antecedents 6

1.3 Objectiu 7

1.4 Pla de treball 9

1.4.1 Etapa de formació 9

1.4.2 Etapa d'implementació 10

1.5 Temporalització 12

1.6 Estructura del document 13

2. Entorn mèdic ***14***

2.1 El procés del diagnòstic 15

2.1.1 Adquisició de dades del pacient. Angiografia 16

2.1.2. Fitxers DICOM 18

2.1.3. Diagnòstic 19

2.2. Segmentació del sistema vascular 20

2.2.1. Segmentació del sistema vascular a partir de superfícies 21

2.2.2. Pre-processament de les imatges 23

2.3. Reconstrucció del sistema vascular 25

2.4. Visualització 34

2.5. Interacció amb el model 34

3.La plataforma Starviewer ***35***

3.1. Estructura de la Plataforma l'Starviewer 41

3.1.1. Mòduls de la plataforma 41

3.1.2. Mòduls que es poden veure afectats pel nostre treball 42

3.2. Integració de les VMTK 42

3.3. Integrar VMTK a Starviewer 45

4. Implementació. Visió General del problema ***48***

5. Segmentació ***50***

5.1. Anàlisi de requeriments 51

5.2. Requeriments funcionals 51

5.3. Diagrames de casos d'ús 51

5.4. Disseny i implementació 55
6. Reconstrucció 61
6.1. Anàlisi de requeriments 62
6.2. Requeriments funcionals 62
6.3. Diagrames de casos d'ús 62
6.4. Disseny i implementació 64
7. Visualització 75
7.1. Anàlisi de requeriments 76
7.2. Requeriments funcionals 76
7.3. Diagrames de casos d'ús 76
7.4. Disseny i implementació 76
8. Interacció 79
8.1. Anàlisi de requeriments 80
8.2. Requeriments funcionals 80
8.3. Diagrames de casos d'ús 80
8.4. Disseny i implementació 80
9. Resultats 83
10. Conclusions i Treball Futur 88
11. Bibliografia 92

CAPÍTOL 1

Introducció i objectius

Les artèries caròtides son les artèries principals del coll que proporcionen sang al cervell. La reducció de l'amplada d'aquestes artèries es coneix com a ***estenosi de caròtida****.*

1.1 Introducció

L'estenosi de les artèries caròtides es produeix per una acumulació de placa. Aquesta placa es forma quan el colesterol, la grassa i altres substàncies cobreixen les parets de les artèries reduint-ne la seva amplada i podent provocar al llarg del temps una obstrucció parcial o total que dóna lloc a una hemorràgia cerebral que dificulta l'arribada de sang i oxigen al cervell (veure Figura 1).

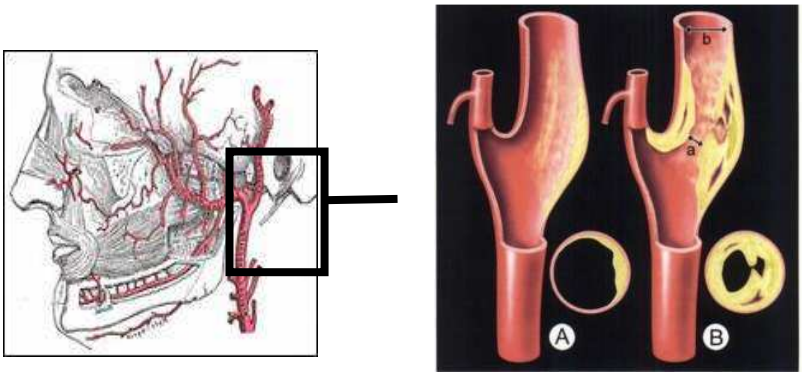


Figura 1. Estenosi d'artèries caròtides.

Una de les tècniques aplicades per diagnosticar estenosis és **l'angiografia per ressonància magnètica** amb contrast (ARM). Aquesta tècnica permet obtenir una representació del sistema vascular cerebral. El principal problema de la tècnica és el fet que els efectes del contrast són molt breus i fa que les adquisicions de les imatges s'hagin de fer de forma molt ràpida. Fins i tot en el cas de disposar de dispositius de captació d'última generació hi ha una limitació entre la resolució de les imatges, la relació senyal soroll, la cobertura anatòmica i la finestra de temps per l'adquisició vascular. Per resoldre aquestes limitacions en aquests darrers anys s'ha treballat en nous contrastos, entre ells els *Blood Pool Agent* (BPA).

Els BPA tenen una vida intravascular molt més elevada permeten fer adquisicions molt més detallades i amb molta més informació. La principal limitació d'aquests nous agents és la interpretació de tota la informació que

apareix en la imatge. El radiòleg té dificultats en saber la posició exacte de les venes i de les artèries i resulta especialment difícil interpretar la posició del vasos de menor calibratge. Tot això fa que la localització de les lesions sigui difícil.

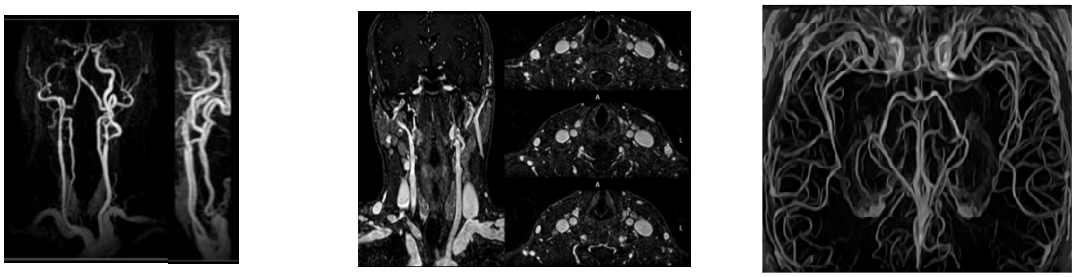


Figura 2. (a) ARM amb contrast convencional, (b)(c)ARM amb BPA

L'aplicació de tècniques de processament i visualització d'imatges pot ajudar en el procés de diagnòstic de l'estenosi de caròtida. De fet, en aquest darrers anys s'han proposat diferents estratègies per segmentar el sistema vascular. Un cop segmentat es pot visualitzar de forma independent i l'especialista pot prendre les mesures en determinades zones per determinar el grau d'estenosi del pacient i fixar així el tractament.

1.2 Antecedents

El Grup d'Informàtica Gràfica de la Universitat de Girona i l'Institut de Diagnòstic per la Imatge (IDI) de l'hospital Dr. Josep Trueta tenen un signat un conveni de col·laboració sota en qual estant desenvolupant noves eines informàtiques de suport al diagnòstic. Fruit d'aquest col·laboració ha nascut la plataforma Starviewer. Starviewer es un entorn informàtic que integra funcionalitats bàsiques i avançades pel processament i la visualització d'imatges mèdiques. Una de les limitacions de la plataforma és el no suportar el tractament de lesions del sistema vascular.

1.3 Objectiu

L'objectiu d'aquest projecte és ampliar la plataforma Starviewer integrant els mòduls necessaris per donar suport al diagnòstic de l'estenosi de caròtida permeten interpretar de forma més fàcil les imatges de Angiografia per Ressonància Magnètica (ARM).

Per assolir aquest objectiu ens cal:

- **Familiaritzar-nos amb la plataforma Starviewer i amb les diferents llibreries que s'usen en el seu desenvolupament.** Ens haurem d'integrar en l'equip de treball Starviewer i aprendre la seva metodologia de treball.

- **Estudiar les imatges de ARM per analitzar les seves característiques i identificar-ne la seva problemàtica**. Haurem d'aprendre el tipus d'operacions que cal realitzar per poder satisfer les necessitats dels especialistes.

- **Estudiar, dissenyar i implementar un mètode de segmentació que ens permeti extreure la informació del sistema vascular de l'ARM**. Hem de tenir en compte que les imatges de ARM a més de la informació del sistema vascular queden representades les altres estructures del crani. El mètode de segmentació ens ha d'ajudar a extreure la informació que es correspon només a venes.

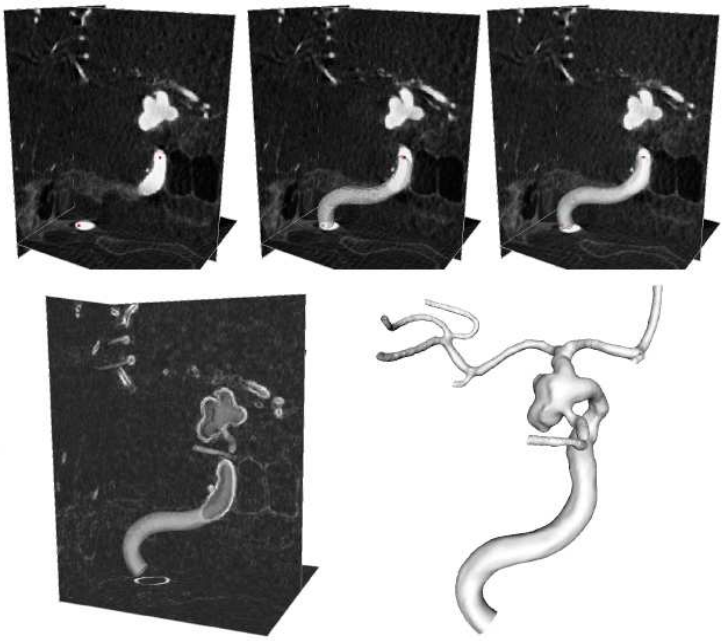


Figura 3. Exemple segmentació vascular.

- **Estudiar, dissenyar i implementar un mètode de representació de venes que ens serveixi per representar la informació obtinguda en el procés de segmentació**. Hem de tenir en compte que el mètode de segmentació ens identifica els vòxels de la ARM que contenen vena però no ens dona una representació dels sistema vascular.

- **Estudiar, dissenyar i implementar un mètode de visualització per poder mostrar la informació en la pantalla**. Una vegada segmentat i reconstruït el sistema vascular cal una tècnica per poder visualitzar-ho en pantalla.

- Fer algun mètode per tal que l'usuari pugui interaccionar amb el model pensant en prendre mesures per poder determinar el grau d'estenosi.

1.4 Pla de treball

Per poder assolir els objectius del projecte ens hem fixat el següent pla de treball dividit en dues etapes:

1.4.1 Etapa de formació

En aquesta fase estudiarem l'entorn mèdic i també l'entorn de treball en el que s'haurà de realitzar el desenvolupament.

Entorn mèdic

La seqüència del diagnòstic que realitza el personal mèdic és la següent: s'adquireixen les imatges del pacient, en el nostre cas (ARM). Aquestes informacions es guarden en un fitxer seguint el format DICOM. A continuació es visualitzen usant un entorn de visualització especialitzat i es procedeix a realitzar el diagnòstic.

Entorn de programació

El nostre entorn de programació serà la plataforma Starviewer.
El llenguatge bàsic de programació de la plataforma és el C++ que es complementa amb tres llibreries:

- ***Qt versió 4.2:*** Eina utilitzada per la creació d'interfícies gràfiques d'usuari.

- ***Insight Toolkit (ITK) versió 2.8:*** Eina pensada pel processament d'imatges com la segmentació i el registre.

- ***Visualization Toolkit (VTK) versió 5.0*****:** Eina pensada pel processat i visualització d'imatges.

Donat que cap d'aquestes llibreries esta pensada per donar suport a processament de venes estudiarem la llibreria Vascular Modeling Toolkit VMTK.

Aquesta llibreria va sorgir a l'any 2008 pensada a tractar informacions mèdiques vasculars integra eines per a la reconstrucció 3D, per a l'anàlisi geomètric, per a la generació de malles i anàlisis d'imatges basades en el modelatge de vasos sanguinis.

1.4.2 Etapa d' implementació

Una vegada familiaritzats amb el problema i l'entorn de treball ens centrarem en la fase d'implementació.

La metodologia que aplicarem serà ***l'Extreme Programming (XP)*** ja que es l'aplicada en el projecte Starviewer.

Les característiques fonamentals d'aquesta metodologia és el desenvolupament és iteratiu i incremental en el que es va fent millores poc a poc, una darrere de l'altre. Les proves són unitàries i contínues freqüentment, pel que ajuda a la resolució d'errors ja que el codi s'ha escrit recentment. Per tant sempre s'utilitza una mateixa font de dades, en el nostre ARMs per a tot el projecte i radiografies per a comprovar que els altres mòduls funcionen.

També l'ús de la metodologia XP permet que el client, en aquest cas el personal mèdic, pugui intervenir periòdicament per a que exposin el seu punt de vista a la hora de millorar segons les seves necessitats. És més fàcil refer el funcionament a mesura que el codi va evolucionant.

Com que el projecte s'ha d'adequar dins l'estructura de l'Starviewer, una plataforma ja creada i amb molts mòduls que no han de sofrir canvis durant

l’evolució del nostre mòdul, l’XP en ser una programació en equip, és sens dubte la millor opció per tal d’assegurar una major qualitat del codi.

1.5 Temporalització

A continuació veurem el diagrama de Gantt a on hi ha el temps que he dedicat a cada part.

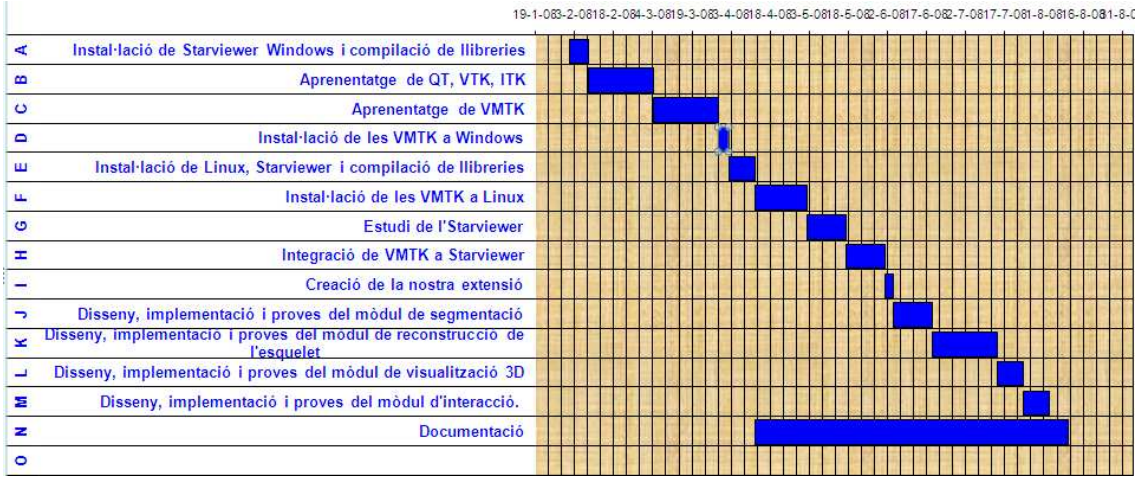


Figura 4. Diagrama de GANTT

	NOMBRE	DURACIÓN	ACTIVIDAD PRECEDENTE	INICIO	FINALIZACIÓN
A	Instal·lació de Starviewer Windows i compilació de llibreries	7		01/02/2008	07/02/2008
B	Aprenentatge de QT, VTK, ITK	25	A	08/02/2008	03/03/2008
C	Aprenentatge de VMTK	25	B	04/03/2008	28/03/2008
D	Instal·lació de les VMTK a Windows	4	C	29/03/2008	01/04/2008
E	Instal·lació de Linux, Starviewer i compilació de llibreries	10	D	02/04/2008	11/04/2008
F	Instal·lació de les VMTK a Linux	20	E	12/04/2008	01/05/2008
G	Estudi de l'Starviewer	15	F	02/05/2008	16/05/2008
H	Integració de VMTK a Starviewer	15	G	17/05/2008	31/05/2008
I	Creació de la nostra extensió	3	H	01/06/2008	03/06/2008
J	Disseny, implementació i proves del mòdul de segmentació	15	I	04/06/2008	18/06/2008
K	Disseny, implementació i proves del mòdul de reconstrucció de l'esquelet	25	J	19/06/2008	13/07/2008
L	Disseny, implementació i proves del mòdul de visualització 3D	10	K	14/07/2008	23/07/2008
M	Disseny, implementació i proves del mòdul d'interacció.	10	L	24/07/2008	02/08/2008
N	Documentació	120	E	12/04/2008	09/08/2008

Figura 5. Temporalització segons cada fase.

1.6 Estructura del document

Aquesta memòria s'ha estructurat en 10 capítols:

- Capítol 2 Entorn mèdic
- Capítol 3. La plataforma Starviewer
- Capítol 4. Implementació. Visió General del problema
- Capítol 5. Segmentació
- Capítol 6. Reconstrucció
- Capítol 7. Visualització
- Capítol 8. Interacció
- Capítol 9. Resultats
- Capítol 10. Conclusions i Treball Futur

CAPÍTOL 2

Entorn Mèdic

En aquest capítol presentarem de forma molt resumida els principals conceptes relacionats amb ***el diagnòstic de caròtida****.*

2.1 El procés de diagnòstic

En la figura 6 presentem un esquema de com es realitza aquest diagnòstic. El procés s'inicia amb l'adquisició de les dades del pacient a partir d'un dispositiu de captació. Aquest dispositiu representa la informació del pacient en forma **d'imatges tomogràfiques**, les quals s'emmagatzemen en un fitxer **DICOM**. Aquest fitxer es guarda en una base de dades especialitzada anomenada **PAC's**. L'especialista accedeix al PAC's i visualitza les imatges en un ordinador per poder realitzar el diagnòstic.

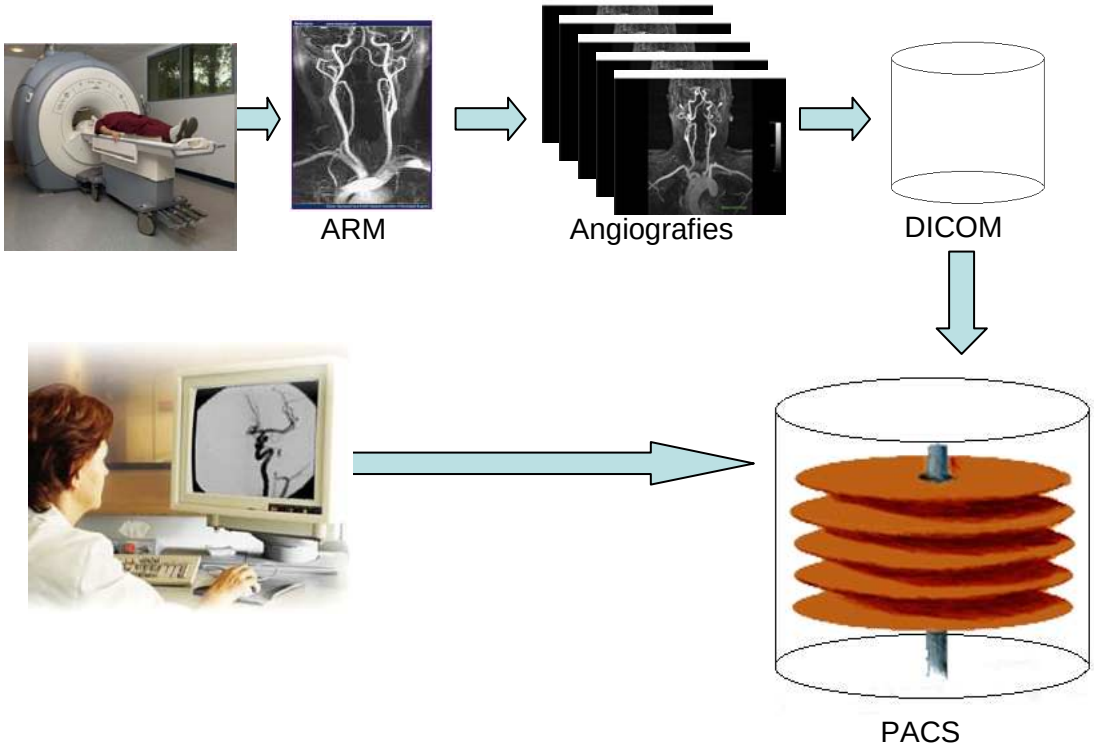


Figura 6. Esquema procés del diagnòstic.

A continuació veurem cada una d'aquestes etapes amb més detall.

2.1.1 Adquisició de dades del pacient. ANGIOGRAFIA

L' angiografia és un examen mèdic molt poc invasiu que ajuda als metges a diagnosticar i tractar malalties relacionades amb el sistema vascular. Per obtenir aquestes imatges es poden usar qualsevol de les tres tecnologies de diagnòstic més bàsiques raig X, Tomografia Computaritzada (TC), Ressonància magnètica nuclear. (RMN). Aquestes tècniques poden anar acompanyades de l'injecció d'un material de contrast per ressaltar encara més els vasos sanguinis. En la imatge de la Figura 7, mostrem un exemple del tipus d'imatge que s'obtenen usant aquestes tècniques.

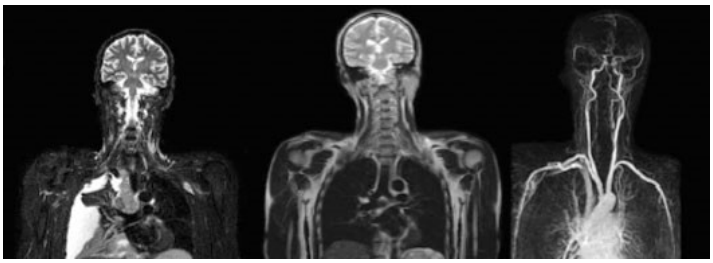


Figura 7. Diferents imatges d'angiografia obtingudes amb ressonància magnètica nuclear

En aquest projecte ens centrarem en les angiografies de ressonància magnètica (ARM).

La RM constitueix una tècnica molt sensible basada en els moviments dels nuclis d'hidrogen. Aquest fenomen s'aprofita per a generar imatges, diferenciant vòxels amb moviment nuclear en el seu si (mòbils) dels que no el presenten (estacionaris). En la ARM s'aprofiten les diferències entre els nuclis estacionaris i els mòbils, ja sigui per absorció selectiva dels polsos de radiofreqüència o pel desfasament al desplaçar-se sota gradients magnètics.

A diferència dels exàmens convencionals de rajos X i l'exploració per tomografia computaritzada (TC), la RM no depèn de la radiació sinó que el camp magnètic es produeix en passar una corrent elèctrica a través de les

bobines de cable de la majoria de les unitats de RM, per tant l'avantatge d'aquestes angiografies és que no utilitza radiació ionitzant i per tant es un tècnica no invasiva.

Es poden realitzar de dues maneres, amb o sense **material de contrast**, el qual s'injecta normalment en una vena del braç. L'efecte d'aquest material un cop introduït al torrent sanguini és una millora en claredat dels vasos sanguinis afectats ja que apareixen de color blanc brillant.

Gràcies a la seva gran aportació en visibilitat dels vasos sanguinis, les angiografies de caròtida són necessàries per als metges per a identificar l'estenosi de caròtida, i per a ajudar a preparar la intervenció quirúrgica en cas que sigui necessari.

Les artèries caròtides són les artèries principals del coll que proporcionen sang al cervell. Per tant nosaltres ens centrarem en ARM cerebrals.

Figura 8. Angiografia de ressonància magnètica cerebral

2.1.2 Fitxers DICOM

Les dades que s'han obtingut es guarden en un fitxer DICOM.

El **DICOM** (Digital Imaging and Communication in Medicine) és l'estàndard reconegut mundialment per l'intercanvi d'imatges mèdiques, pensats per l'ús, emmagatzematge impressió i transmissió d'imatges mèdiques. Inclou la definició d'un format de fitxer i d'un protocol de comunicació de xarxa. El protocol de comunicació és un protocol d'aplicació que fa servir TCP/IP per la comunicació entre sistemes. Els fitxers DICOM poden intercanviar-se entre dues entitats que tinguin capacitat de rebre imatges i dades de pacients en format DICOM.

DICOM permet la integració d'escàners, servidors, estacions de treball, impressores i hardware de xarxa de múltiples proveïdors dins d'un sistema d'emmagatzematge i comunicació d'imatges. Les diferents màquines, servidors i estacions de treball tenen una declaració de conformitat DICOM (conformance statments) que estableix clarament les classes DICOM que suporten. DICOM ha sigut adoptat àmpliament per hospitals i està fent incursió a petites aplicacions de dentistes i doctors.

Es diferencia d'altres fitxers de dades en que agrupa la informació dins d'un conjunt de dades. És a dir, una radiografia de pit conté la ID del pacient juntament amb ella, de manera que la imatge no pot ser separada per error de la seva informació.

Els fitxers DICOM consisteixen en una capçalera amb camps estandarditzats i de forma lliure, i un cos amb dades d'imatge. Un objecte DICOM sempre pot contenir solsament una imatge, però aquesta imatge pot tenir múltiples "fotogrames" (frames), permetent l'emmagatzematge de blocs de cine o qualsevol altres dades amb diferents fotogrames. Les dades d'imatge poden

estar comprimides utilitzant una gran varietat d'estàndards com JPEG, JPEG Lossles, JPEG 2000, LZW i Run-length encoding (RLE).

El **PACS** (Picture Archiving and communication system) És un sistema computaritzat per l'emmagatzematge digital d'imatges mèdiques (medicina nuclear, tomografies computades, ecografies, mamografies....) i per a la transmissió d'aquestes a estacions de visualització dedicades a traves d'una xarxa informàtica. Com a tecnologia sanitària està regulada a Europa per la directiva 93/42/EEC. Tots els fitxers DICOM que s'adquireixen es guarden en el PACS

2.1.3 Diagnòstic

Una vegada les dades estan en el PACs de l'hospital, el radiòleg ja pot realitzar el diagnòstic. Per fer-ho ha d'accedir al PACs a través d'un programa autoritzat i poder visualitzar les imatges que s'han adquirit del pacient. Si el visor que s'usa no es un visor especialitzat només podrà veure les imatges tal com les ha adquirit l'aparell, és a dir de forma seqüencial com un conjunt de plans 2D.

Als especialistes aquesta opció no és la que més els interessa. ***De fet el que interessa es poder veure una reconstrucció 3D de tot el sistema vascular i poder prendre mesures sobre ell.*** Tal i com es mostra en la imatge de la Figura 6, el càlcul s'ha de poder fer sobre un volum en tres dimensions.

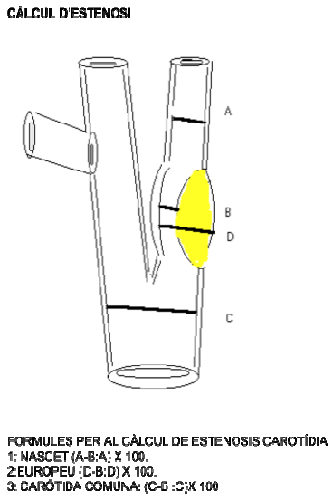


Figura 9. Càlcul d'estenosi de caròtida

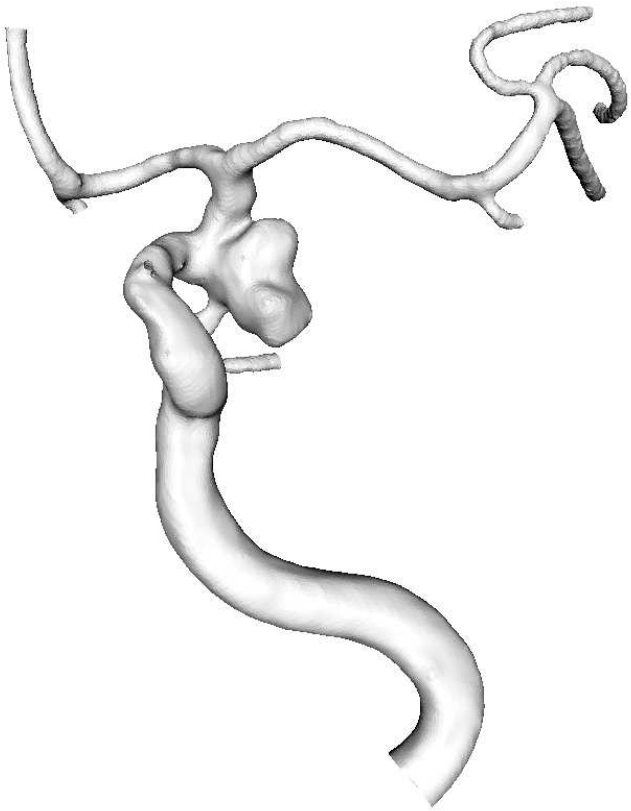


Figura 10. Exemple volum del sistema vascular.

A la figura 10 podem veure com el volum és molt més clar i a l'especialista li és més fàcil poder mesurar la distància per tal de poder diagnosticar l'estenosi.
No obstant, poder obtenir una representació del sistema vascular no es trivial i cal aplicar tècniques de segmentació, reconstrucció i visualització especialitzades. A continuació veurem quines són les diferents tècniques que s'han proposat per realitzar aquests processos.

2.2 Segmentació del sistema vascular

El procés de reconstrucció del sistema vascular no es una tasca fàcil i això ha fet que es proposessin diferents tècniques per dur-lo a terme. Abans de decidir quina tècnica aplicaríem hem fet un estudi previ dels diferents mètodes que s'han proposat. Es important tenir clar que l'objectiu de la segmentació és obtenir una partició de la imatge en regions coherents en el nostre cas

aquestes regions han de ser les que defineixin les venes. El problema que es planteja és que les venes no tenen perquè seguir regions contínues en una imatge 2D i per tant es fa difícil saber com connectar les zones. Hem de tenir en compte que el sistema vascular no segueix un camí únic i que està format per diferents bifurcacions. A més a més, la connexió entre les diferents imatges 2D que composen tot el model encara dificulta més tot el procés.

2.2.1 Segmentació del sistema vascular a partir de superfícies

En el cas de les imatges de RMA les estructures vasculars no coincideixen amb cap altre valor d'intensitat del imatge. Aquesta característica ens permet aplicar mètodes de reconstrucció de superfícies per tal de detectar i reconstruir automàticament el sistema vascular. En el cas de les angiografies basades en CT aquesta tècnica no es pot aplicar ja que el valor d'intensitat de les venes coincideix amb el valor de les estructures òssies.

Els algorismes de reconstrucció de superfícies el que fan es extreure d'un model de volum la superfície que conté tots els punts amb un valor d'intensitat determinat. L'algorisme més popular és el Marching Cubes proposat per Lorensen i Cline el 1987.

Aquest algorisme una vegada determina el valor d'intensitat de la superfície, conegut com *isovalor*, recorre tot el model de volum i classifica els vèrtexs dels vòxels com a interiors o exteriors a la superfície en funció de si tenen un valor més gran o més petit que *l'isovalor*, respectivament. Es considera que la superfície intersecta totes les arestes que tenen un extrem classificat com interior i l'altre com exterior. En funció de les arestes intersectades es defineix de forma automàtica la forma que té la superfície a l'interior del vòxel. Aquesta triangulació es fa seguint la taula de patrons que es mostra en la Figura 11.

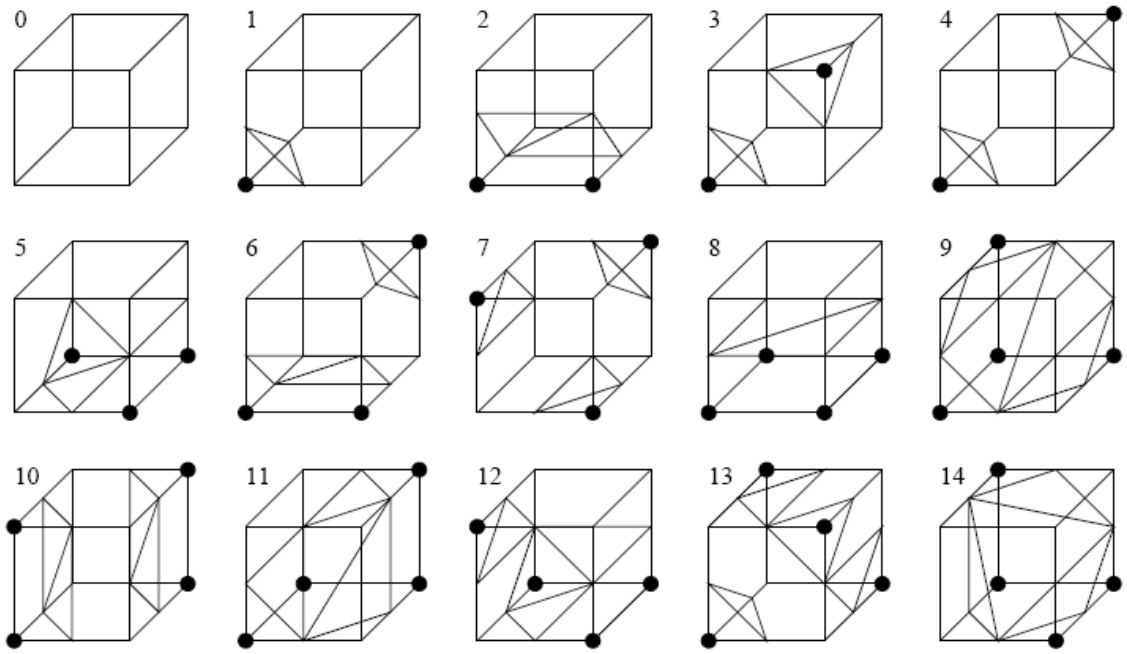


Figura 11 Casos de triangulacions possibles en un cub

Avantatges

- Produeix superfícies de molt bona qualitat amb un baix cost computacional, per lo que es una tècnica apropiada per la generació de casos de prova d'entrenadors quirúrgics.
- Fàcil renderitzat (projecció dels vòxels que formen el volum de la imatge sobre la pantalla d'un ordinador) i manipulació.
- Alta resolució.

Inconvenients:

- Quan hi ha massa informació a l'objecte d'estudi, per exemple a la ressonància magnètica d'un genoll, s'observen els ossos, meniscs, lligaments, la superfície obtinguda mostra una visió global. Si l'algorisme treballés sobre un volum prèviament segmentat, es podria obtenir la superfície 3D de la zona desitjada.
- Possibles forats a la superfície generada a causa de cares ambigües:

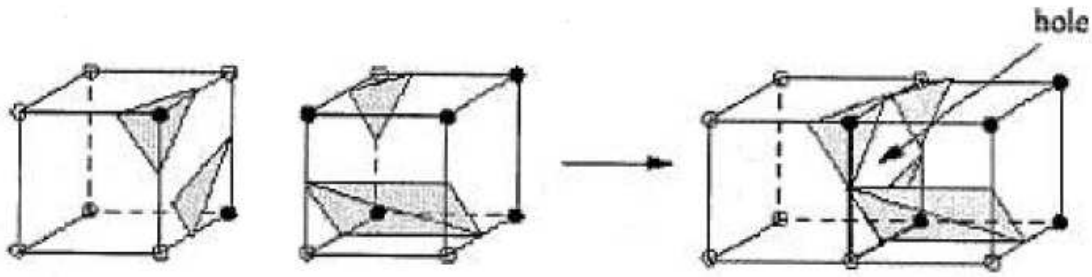


Figura 12. Exemple de forat. Es podria resoldre ampliant el nombre de casos de bàsics de 15 a 30.

- En tot moment s'analitzen cubs d'igual grandària sigui la regió que s'estigui estudiant. Això produeix que l'algorisme no obtingui un conjunt de triangles que determini de manera òptima la forma.

2.2.2 Pre-processament de les imatges

Donada la dificultat de la segmentació i la posterior reconstrucció per millorar els resultats en algunes ocasions el que es fa es pre-processar les dades originals.

En les cas de les imatges cerebrals un dels mètodes més coneguts és el proposar per Manniesing et al 2006. El mètode està basat en una difusió controlada o en un procés de suavització que genera cada vegada que s'aplica, millor versions suavitzades de la imatge original. La representació d'una imatge a nivells creixents de suavitat és anomenada escala d'espai (scale space).

En un escalat d'espai convencional, totes les estructures són suavitzades equitativament.

Amb aquest mètode es pretén reforçar les aproximacions en estructures vasculars, que s'aconsegueix primer computaritzant localment les venes semblants i la orientació de les venes utilitzant posteriorment una informació d'ordre d'imatge.

El nivell de suavització llavors és construït depenent de la similitud de les venes i la orientació. Si les venes són molt semblants, la difusió té lloc pel llarg dels eixos vasculars (anomenat difusió isotròpica). En aquesta via, les estructures vasculars són millorades i les no vasculars suprimides.

Aquí podem apreciar la diferència un cop aplicat aquest mètode:

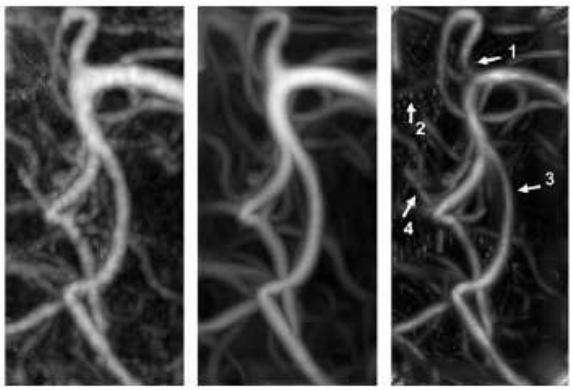


Fig. 4. A comparison between VED and the vesselness filter. The original data is shown in the first frame, the result of VED in the second frame and the result of vesselness filtering in the third frame. Arrows indicate points of interests, see Section 4.3.

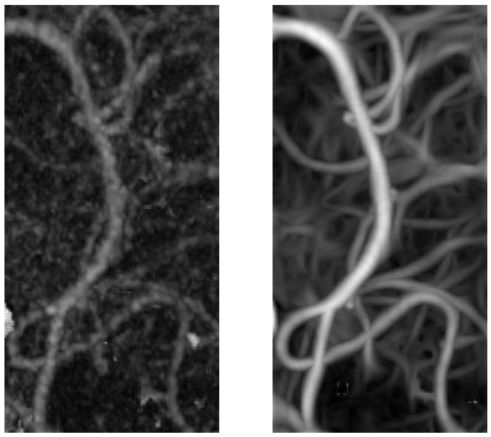


Figura 13. Exemple de diferència imatge original CTA(a) després reforç (b).

A la figura 13 podem veure l'exemple de MIPs axial d'una part del vascular cerebral. A l'esquerra la imatge original pobre d'una CTA i a la dreta el resultat després de la difusió de reforç vasal.

2.3 Reconstrucció del sistema vascular

Una vegada s'han identificat les venes de les diferents imatges 2D per poder generar el sistema vascular cal aplicar un mètode de reconstrucció que sigui capaç de connectar les diferents parts. Aquests sistemes de reconstrucció tenen com a objectiu obtenir l'esquelet del sistema vascular.

En el cas de la segmentació a partir de superfícies ja disposem d'una reconstrucció 3D de tot el model vascular però tot i això ens cal identificar quin és l'esquelet d'aquest sistema.

Per poder reconstruir l'esquelet del sistema és important introduir el concepte de ***centerline***. El centerline és la línia central (o eix) fonamental del model de cilindre generalitzat que es fa servir per representar objectes tubulars com les artèries (veure Figura 14). Aquest model és definit pel seu eix central i la seva superfície.

L'extracció de la línia central de la artèria és essencial ja que proveeix informació simplificada del vas, que permet caracteritzar la topologia de l'arbre vascular i prendre mesures quantitatives, com la longitud curvilínia de la arteria, el diàmetre o l'àrea de les seccions ortogonals al eix.

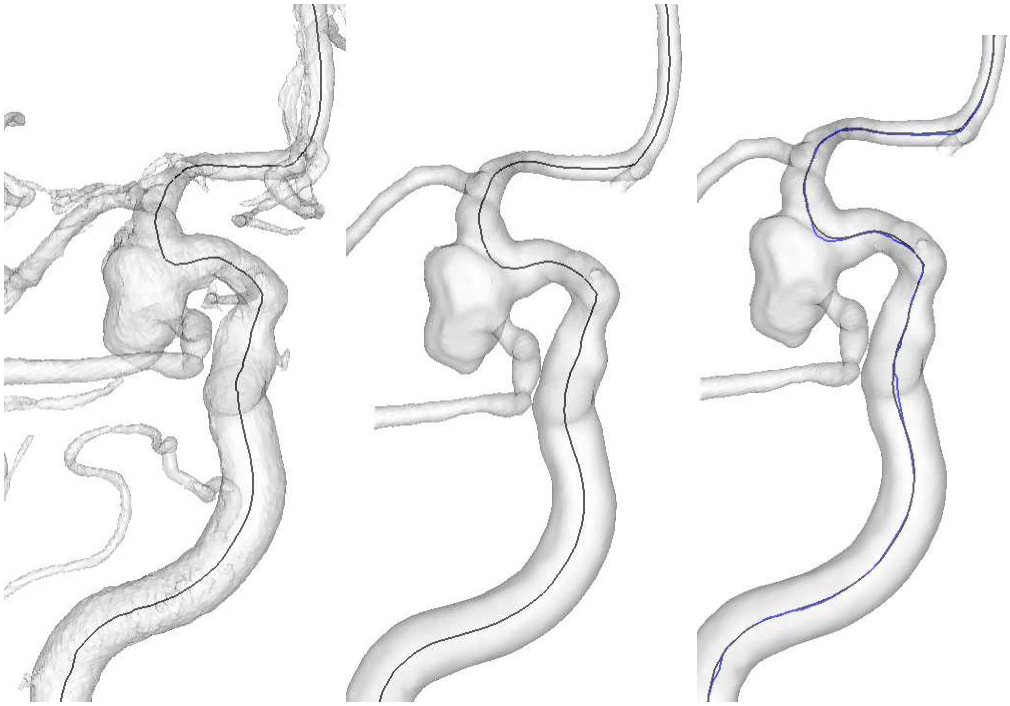


Figura 14. Exemple de centerlines

En el cas de tenir la superfície del sistema vascular a partir de superfícies es pot obtenir l'esquelet aplicant un algorisme d'anàlisis topològic. L'objectiu és identificar les bifurcacions de les venes i l'organització de la xarxa vascular i donar una aproximació de les centerlines.

La idea de l'algorisme és que una simple vena, des d'un punt de vista topològic, és un cilindre limitat per dos seccions circulars també conegudes com a geometria 2½D. Un cilindre topològic pot ser subdividit en dos peces cilíndriques més petites a partir de tallar la superfície produint seccions circulars limitades. Aquest efecte de tallar pot ser produït amb una esfera centrada en una secció limitada per un baricentre amb radi més gran que la màxima distancia entre el baricentre i el punt més llunyà de la secció.

Si comencem a partir d'una de les dues seccions limitades en un cilindre topològic a tallar subcilindres petits amb una esfera centrada a cada nova limitació creada i completament continguda, crearem una nova secció limitada a cada pas. En canvi, en presencia d'una bifurcació, dos o més noves seccions limitadores seran creades. Llavors basant-nos en aquestes observacions,

l'algorisme consisteix en visitar tota la xarxa dels segments vasculars amb una esfera que es mou, que l'anomenarem *esfera investigadora*, per identificar totes les bifurcacions i descompondre la xarxa en bifurcacions i segments vasculars singulars.

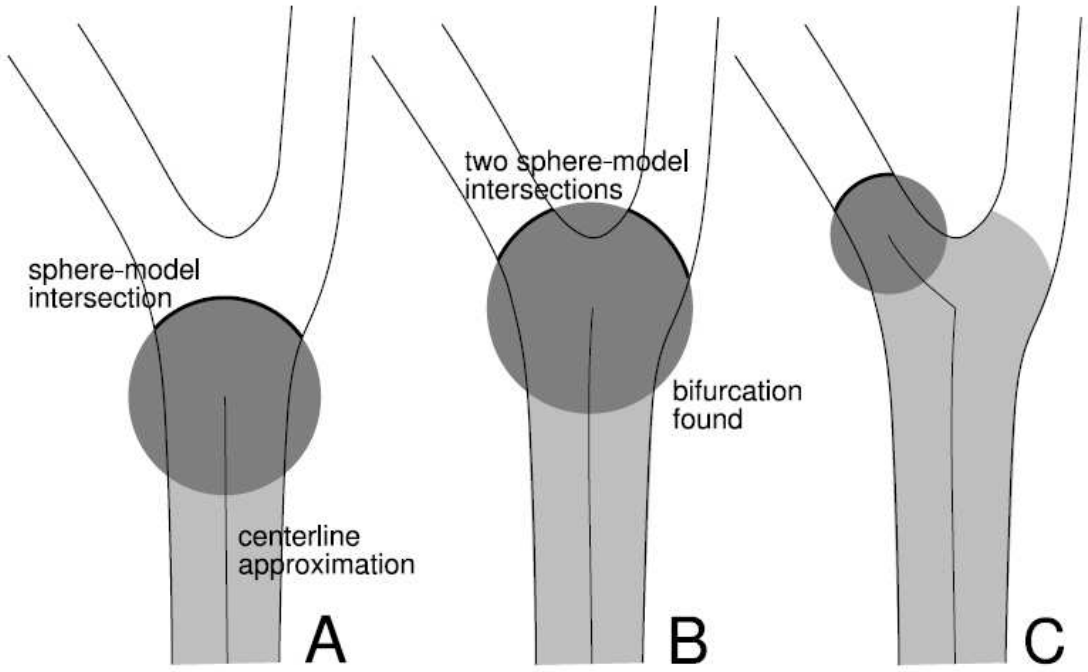


Figura 15. Exemple de funcionament de l'algorisme topològic.

A la figura 15 podem veure una representació conceptual de l'algorisme d'anàlisis topològic. En la figura de l'esquerra (a) es genera una intersecció entre l'esfera investigadora i el model i s'identifica un cilindre topològic. A la figura del mig (b) es generen dues interseccions, per tant s'identifica una bifurcació. A la figura de la dreta (c) l'esfera investigadora torna a començar un perfil d'intersecció. Llavors els centerlines són generats a partir de la connexió dels centres de l'esfera investigadora durant el seu moviment.

A més, el baricentre de les seccions limitadores ens donen una aproximació de les centerlines de les artèries, que servirà per dibuixar informació sobre la configuració espacial del vascular. En comptes de tallar els segments vasals amb l'esfera que es mou, que suposaria una retriangulació de tots els triangles que intersecten amb l'esfera, els punts de la superfície vascular són visitats

amb l'esfera investigadora i els perfils de les seccions són dibuixats de manera simple a la superfície d'aquesta manera reduïm el cost computacional requerit.

En més detall, l'algorisme comença amb l'identificació dels perfils oberts de l'estructura vasal, que són els perfils d'entrada i sortida. És per això, que si el model és una superfície tancada, almenys un perfil obert s'ha de crear.

Llavors quan es tria un perfil, el seu baricentre es calcula. Com que els perfils són línies de segments, els seus baricentres són computats tenint en compte els segments que connecten els vèrtexs polilínies a partir de la ponderació de les posicions dels punts amb la mitja longitud dels seus segments veïns. L'esfera investigadora és llavors definida amb el centre $\boldsymbol{c_s}$ del baricentre ***b*** del perfil definit ***P*** i amb un radi igual a $r_s = k\, d_{max}$ a on $d_{max} = \max\{|\mathbf{b} - \mathbf{P}|\}$ és la màxima distància d'un punt del perfil fins al baricentre i ***K*** és una constant definida més gran que 1 (per exemple ***k*** = 1.05)

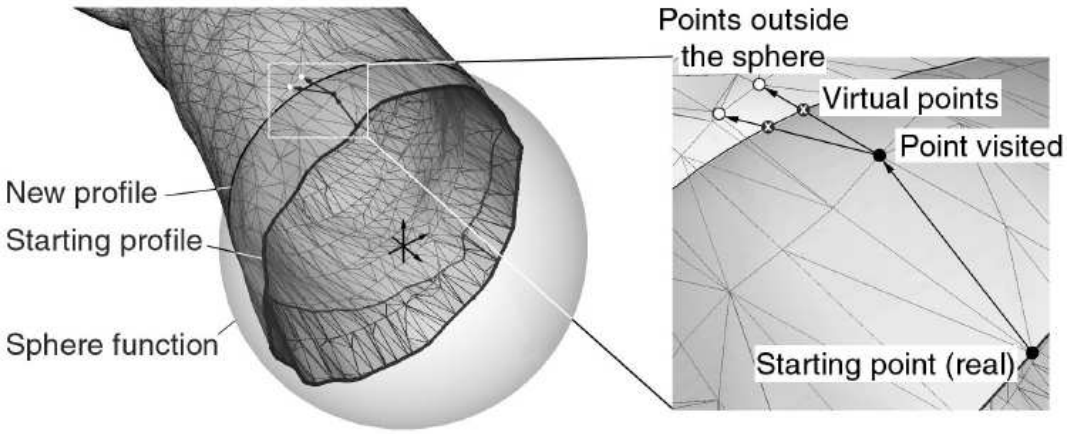


Figura 16. Un pas de l'algorisme d'anàlisis topològic. Els punts d'intersecció són anomenats com punts virtuals per evidenciar que no són vèrtexs de la superfície triangularitzada, però són creats en els triangles del marge.

En aquest punt, es calcula la intersecció de l'esfera amb la superfície connectada amb el perfil. S'aconsegueix a partir de visitar iterativament els vèrtexs veïns de la superfície de la malla començant pel perfil definit i comprovant si: $|\mathbf{p} - \mathbf{c}_s| > r_s$ a on ***p*** és la posició d'un vèrtex d'una superfície

de la malla genèrica. Els vèrtexs de la malla que han sigut visitats però que no satisfan la condició anterior, són marcats com a *visitats*. Tan aviat com un punt satisfà la condició, significa que la intersecció entre l'esfera investigadora i la superfície és localitzada en un dels marges connectats amb el punt, i que el seu altre punt és un punt visitat que està dins de l'esfera. El punt exacte de la intersecció és determinat amb les següents relacions:

$$s = (r_s^2 - h^2)^{\frac{1}{2}} - (t^2 - h^2)^{\frac{1}{2}} \qquad \text{if} \quad |\mathbf{p} - \mathbf{q}|^2 < (u^2 - h^2)$$
$$s = (r_s^2 - h^2)^{\frac{1}{2}} + (t^2 - h^2)^{\frac{1}{2}} \qquad \text{if} \quad |\mathbf{p} - \mathbf{q}|^2 \geq (u^2 - h^2)$$

Figura 17. Fórmula per determinar el punt exacte de la intersecció.

A on les quantitats són especificades a la figura 17 i el pla dibuixat és definit pel marge ***(p – q)*** i el centre de l'esfera $\boldsymbol{c_s}$. Cada punt d'intersecció constitueix un vèrtex de les polilínies representant nous perfils generats. Per tal de reconstruir la connectivitat de les polilínies, cada dos punts d'intersecció són connectats per un costat si els dos recauen en dos costats del mateix triangle.

Un cop els perfils són reconstruïts poden ser calculats. Si es genera un nou perfil, la superfície entre la definida i el nou perfil forma un cilindre topològic, i com a conseqüència, el nou perfil és pres com a perfil definit i s'executa un nou pas.

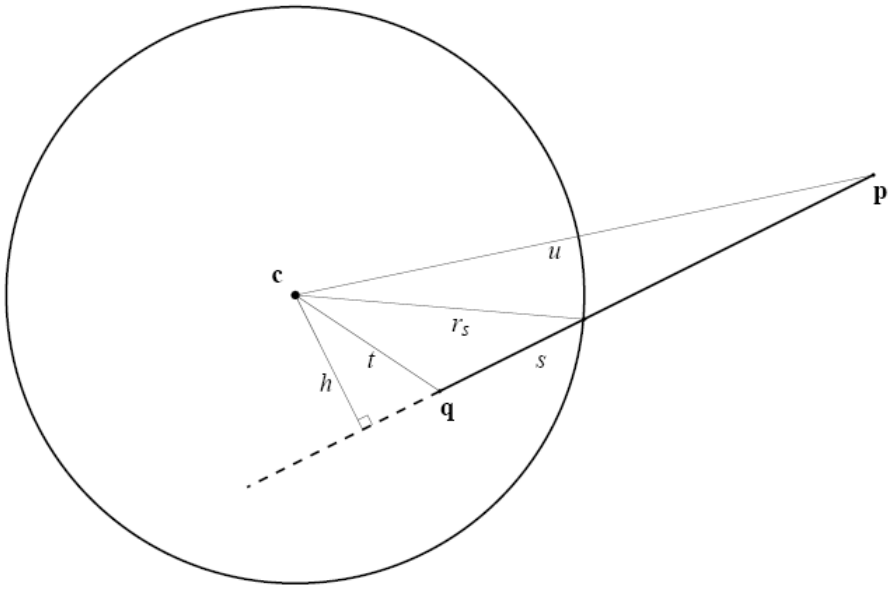


Figura 18. Determinació de la intersecció entre l'esfera investigadora i un marge.

Si no es generen perfils, significa que ens trobem en un final mort d'un segment vascular, és a dir, el conjunt que mira l'esfera investigadora són guardats com a segment vascular i l'anàlisi continua amb un nou segment d'un perfil disponible (p. ex. no visitat).

Si dos o més perfils són generats, la superfície continguda a l'esfera és una bifurcació, per això el segment vascular és guardat i els nou perfils tornen a ser disponibles. Això també pot passar durant el procés de visita d'un punt si un perfil és trobat. Significa que el segment actual finalitza en una bifurcació que ja ha sigut identificada o és una sortida del model.

Cada cop que es troba una bifurcació, se li assigna una única *id*, i els perfils són marcats amb ell. Segons això, sempre que un segment ha sigut construït, els seus punts finals són associats amb la id de la bifurcació que l'engloba. Per això, la xarxa topològica vascular pot ser reconstruïda sense necessitat de cap ordre concret en les visites dels segments.

Durant l'anàlisi dels segments es calcula una estimació de la mitjana del radi a cada pas de l'esfera investigadora, a partir de la computació de la distància

mitjana dels punts visitats en aquell pas concret, des de la línia definida pel baricentre de la definició fins els perfils intersectats. Aquest resultat és una aproximació de la centerline. En les bifurcacions, es calcula el baricentre de la superfície de la bifurcació i llavors s'ajunta amb els punts finals de les branques aferents.

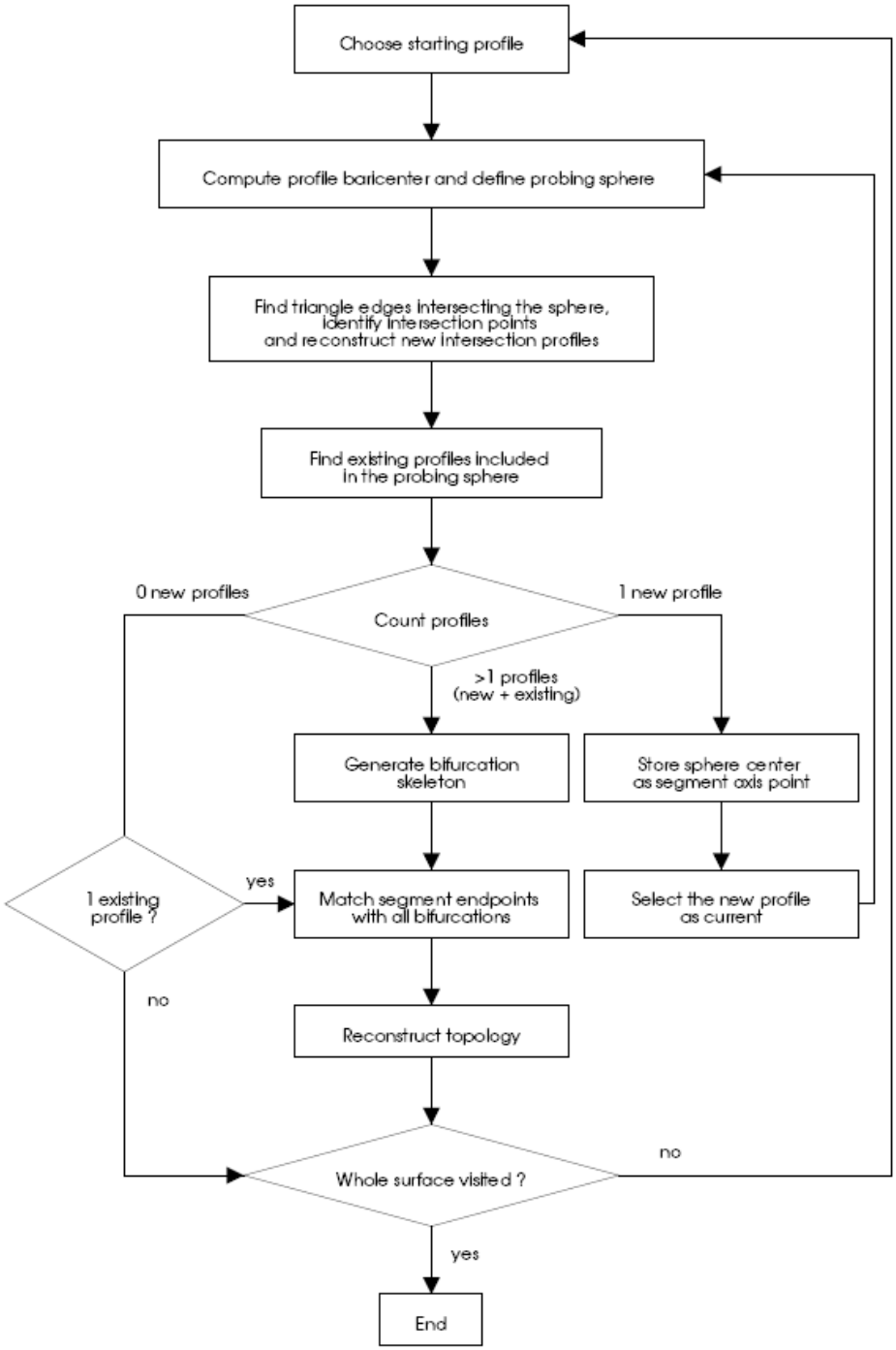


Figura 19. Organigrama de l'algorisme d'anàlisi topològic.

Es pot veure un organigrama de l'algorisme a la figura 19. És molt eficient a causa que la seva complexitat és lineal amb el nombre de vèrtexs de la superfície de la malla, ja que els punts són visitats una vegada. De totes maneres l'algorisme presenta algunes limitacions. Des del punt de vista topològic, es pot mal interpretar en dues situacions:

- Primer, quan un segment es bifurca en dos segments que es tornen a ajuntar un altre cop, les dues bifurcacions poden ser ignorades si el radi de l'esfera investigadora és major que la distància entre les bifurcacions.

- En segon cas, pel mateix motiu quan existeixen diverses bifurcacions aniuades amb una separació relativament curta entre elles, pot ser que s'agrupin en una sola bifurcació comprenent totes les artèries de totes les bifurcacions.

Aquestes limitacions deriven del fet que la mida de l'esfera investigadora pot créixer quan el perfil definit té una forma el·líptica, que podria passar quan s'acosta alguna bifurcació.

No obstant, aquests casos no són limitats, ja que totes les branques que sobresurten són comptades amb l'apropiada connectivitat global. D'altra banda, les limitacions del rendiment de l'anàlisi geomètric són més importants. De fet no garanteix que el segment dels centerlines recaigui en les venes associades, encara que la definició de centerlines adopti com que és el lloc dels baricentres dels perfils definits. En realitat, els centerlines poden anar lligats fora de les venes quan les seccions tenen una forma particular no-convexa.

Malgrat això, el major problema geomètric és que no es pot deduir res de les geometries de les bifurcacions un cop han sigut incloses a l'esfera investigadora. Normalment, els centerlines de les venes singulars són connectades amb el baricentre de la superfície en bifurcació, però això pot ser una aproximació geomètrica massa simplificada.

Aquí podem veure algun exemple de com quedaria:

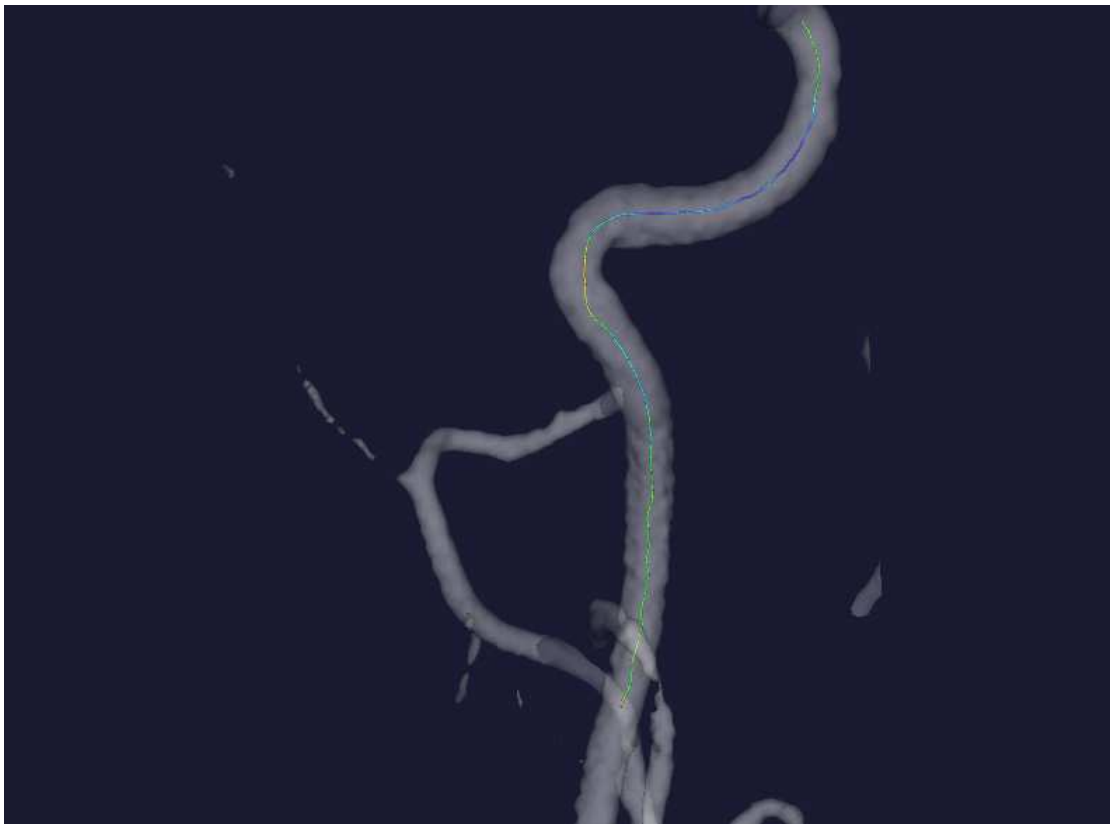


Figura 20. Exemple centerlines en artèria caròtida.

2.4 Visualització

Per poder visualitzar el model vascular podem aplicar dos tipus de tècniques de visualització de volums:

Tècniques de visualització directa de volums, és a dir tècniques que representen tot el model de volum en la planta

Tècniques de visualizatzació de superfiícies, és a dir tècniques que representen la informació que es vol visualitzar com una malla de triangles i només representen aquesta malla. La resta d'informació corresponen a les altres zones del volum no es represeta.

Donat que en el nostre cas ens interessa només el sistema vascular i aquest es pot representar com una malla de triangles optarem per la segona tècnica, la visulització de superfícies.

2.5 Interacció amb el model

Per poder detectar on està l'estenosi el metge el que fa es calcular el diàmetres del vas en diferents punts, tal i com es mostra en la imatge de la figura. En funció de la diferencia que hi ha entre els diferents diàmetres es determina quin és el grau d'estenosi del pacient i com cal actuar.

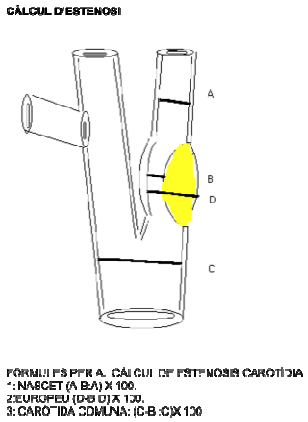


Figura 21 Fórmules per el càlcul de l'estenosi de caròtida.

CAPÍTOL 3

La plataforma Starviewer

El treball realitzat en aquest projecte s'integrarà a la plataforma Starviewer. En aquest capítol presentarem la plataforma amb les eines que utilitza ja que seran les que haurem d'usar..

Com bé hem apuntat a la secció 1.4, el llenguatge principal de programació és el c++ però s'ajuden de tres llibreries que són Qt, VTK i ITK. A continuació farem una breu explicació de cadascuna d'elles:

QT

Qt és una llibreria multiplataforma per desenvolupar interfícies gràfiques d'usuari i també per el desenvolupament de programes sense interfície gràfica com eines de consola i servidors. Qt és utilitzada principalment en KDE, GoogleEarth, Skype...

És produït per la divisió de software Qt de Nokia, que va entrar en vigor després de l'adquisició per part de Nokia de l'empresa noruega Trolltech, el productor original de Qt.

Qt utilitza el llenguatge de programació c++ de forma nativa però addicionalment poden ser utilitzats en diversos llenguatges de programació a través de bindings[1]

És suportada en les diferents plataformes de Microsoft Windows, Unix/x11, Mac OS X i Linux.

Existeixen dos edicions diferents:

- ***Qt Commercial Editions***, que s'utilitza per a software comercial.

- ***Qt Open Source Edition*** per a només el desenvolupament de software gratuït i lliure.

[1] Adaptació d'una biblioteca per ser utilitzada en un llenguatge de programación diferent de aquell en el que ha estat escrita.

VTK

The **Visualization Toolkit (VTK)** és un software lliure i gratuït per a la computació de gràfics en 3D, processament d'imatge i visualització. Funciona tant per Linux, Windows, Mac com per les diferents plataformes de Unix.

VTK consisteix en una biblioteca multiplataforma de classes de c++ i moltes interfícies interpretades per a Tcl/Tk, Java i Phyton. Suporta una gran varietat d'algorismes de visualització incloent mètodes d'escalars, de vectors, de tensors, de textures i de volumètrics; i unes avançades tècniques de modelatge com modelatge implicit, reducció de poligons, suavitzat de superfícies, mètodes de contorns i la triangulació de Delaunay. A més, VTK té molts widgets d'interacció 3D, suporta processaments en parallel i integra diversos bases de dades en llibreries de GUI com Qt i Tk.

El suport professional i els productes són proporcionats per Kitware.

ITK

La llibreria nacional de medicina **Insight Segmentation and Registration Toolkit (ITK)** és un sistema lliure i multiplataforma que proporciona als desenvolupadors una gran quantitat d'eines per a l'anàlisi d'imatges. Desenvolupat a partir de metodologies d'extrem programming, ITK utilitza

algorismes avançats de registrament i segmentació informació multidimensional.

Coding style

Per a poder programar la nostra extensió a l'Starviewer s'ha de seguir una guia d'estil ja que com que cada programador fa servir el seu propi estil i s'està implementant una aplicació que potser molts altres l'utilitzaran és més eficient que sigui net i llegible per a que s'entengui relativament ràpid. D'aquesta manera quan un mateix torni a mirar el codi després d'un cert temps, no es perdi gaire temps en recordar com anava. També interessa que tot l'Starviewer tingui un aspecte uniforme.

El coding style apunta diverses pautes com per exemple a la hora de: estructurar directoris, integrar interfícies, anomenar fitxers (en anglès), no posar abreviacions, utilitzar el namespace anomenat udg per tal que els noms de les funcions, mètodes, classes no interfereixin entre elles, ja que en cas que fossin de diferents namespaces podrien tenir noms iguals...

A més de les llibreries usades per l'Starviewer nosaltres usarem unes noves llibreries les VMTK. Donat que aquest és el primer projecte en el que s'usen aquestes llibreries haurem de realitzar totes les tasques d'integració en la plataforma per poder garantir que tot funcioni correctament.

VMTK

Les eines que utilitzem per a realitzar els blocs de reconstruccó, segmentació i visualització estan basades en les llibreries noves VMTK.

A continuació farem una breu explicació sobre la seva arquitectura amb l'ajuda de la figura 22.

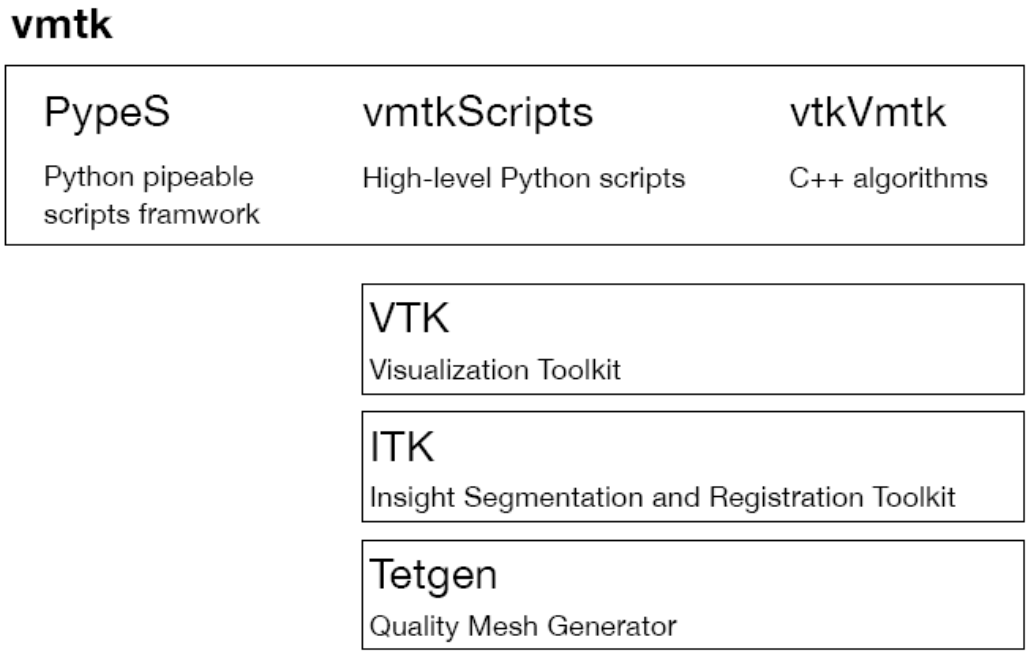


Figura 22. Arquitectura de les vmtk

Un cop s'instal·len les vmtk la seva forma d'execució és mitjançant el que s'anomena PypeS.

Les *PypeS* són la forma en que s'ajunten es diferents *scripts* de vmtk. Permet que els scripts siguin escrits d'una manera senzilla i tenen una interfície amena, però sobretot permeten la interacció entre scripts de manera que fa les vmtk molt modulars i flexibles.

Les PyPes estan constituïdes per dues classes principals:

- ***PypeScript***: és la classe base de cada script d'alt nivell. És desenvolupat per parsejar i instanciar mètodes d'input/output i manté una estructura consistent dels scripts.

- ***Pype***: la classe que gestiona la interacción de les PypeScripts. Permet unir els diferents PypeScripts i té cura de connectar els arguments d'un

script a un altre de forma correcte. Es pot cridar a partir d'una línia de comandes donant arguments de pype o instanciant automáticamente el pypescript.

Els scripts estan realitzats en llenguatge phyton que estan implementades amb l'ajuda de diverses classes en c++.

Veurem un petit esquema per entendre millor el funcionament de les PypeS. Per exemple pertal de realitzar el marching cubes amb un nivell de 200 i poder-ho escriure-ho en un fitxer de sortida, la línia de comandes seria aquesta:

```
vmtkimagereader -ifile foo.vti --pipe
vmtkmarchingcubes -l 200 --pipe
vmtksurfacesmoothing -iterations 30 -passband 0.1 --pipe
vmtksurfacewriter -ofile bar.vtp
```

I la seva respectiva execució seria seguint aquest procés:

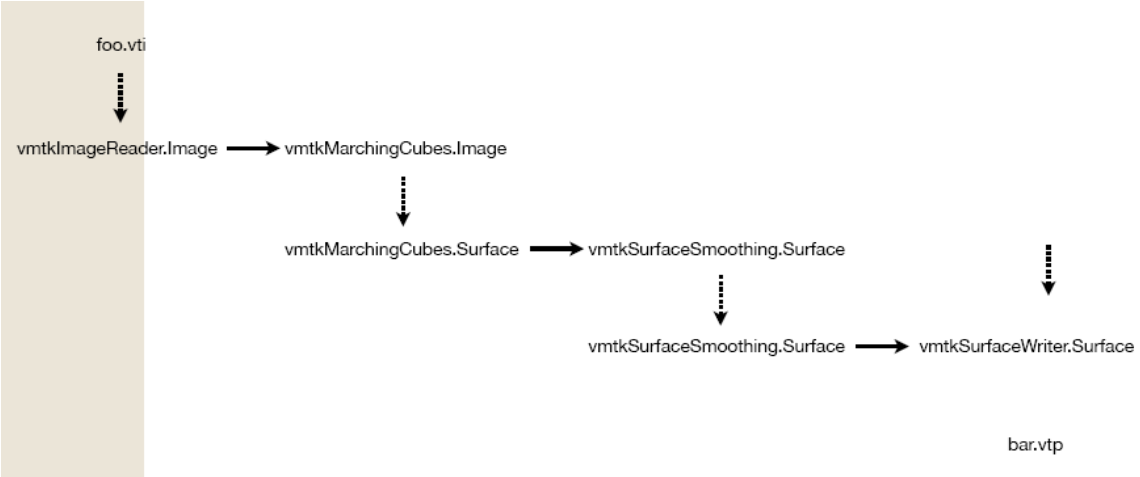


Figura 23. Exemple execució de les PypeS

Malauradament, la plataforma Starviewer en primera instància no suportava les llibreries VMTK per tant un primer pas serà la seva integració. Podem veure la seva instal·lació a l'apartat 3.2.

3.1 Estructura de la Plataforma l'Starviewer

A continuació explicarem els mòduls els quals està format la plataforma Starviewer per entendre millor el seu funcionament i per poder integrar el nostre projecte d'una manera més eficient.

3.1.1 Mòduls de la plataforma

Starviewer està definit per 4 mòduls diferents que són:

- **Aplicació principal**: És a on es dedica la interfície gràfica i les funcionalitats de totes les utilitats possibles d'Starviewer. Hi ha una classe gestora de la pantalla principal anomenada *QApplicationMainWindow.* Llavors la que s'encarrega de gestionar els visors oberts mentre s'executa el programa és *ExtensionWorkspace.*

- **Entrada/Sortida**: Aquest mòdul és molt important ja que és d'on es basarà tot el sistema d'importació i exportació d'exploracions. La classe *ExtensionHandler* demana a la classe *AppImportFile* que s'importi un fitxer determinat. Llavors aquest el demana a la classe *Import* fins que es posa en contacte amb *VolumeRepository* perquè guardi el volum nou.

- **Gestió de fitxers i volums oberts**: Tots els fitxers importats es guarden en el format de la classe Volum. Cada volum té una referència independent i es guarda tota la informació a V*olumeSourceInformation*. Per tal de gestionar els volums oberts en l'aplicació de Starviewer, existeix la classe *VolumeRepository*.

- **Gestió de visors**: Els visors són molt importants a l'aplicació ja que depenent del que l'usuari vulgui realitzar i analitzar, li convindrà fer servir

un visor o un altre. Existeixen els visors 2D i 3D, i dins de cada extensió n'hi poden haver diversos.

Les classes que són visors acaben en *Viewer* i hereten de *QViewer* una classe de Qt. La classe acabada en *ExtensionBase* són les que ens proporcionen la interfície gràfica juntament amb les *tools* que ens ajudaran a fer diferents funcions als visors. I finalment la acabada en *Extension* fa la comunicació entre el visor i la interfície gràfica.

3.1.2 Mòduls que es poden veure afectats pel nostre treball

La nostra extensió no afecta gaire a la totalitat de la plataforma Starviewer ja que forma un cas independent a les altres extensions.

En l'únic que es podria veure afectat és a l'hora de la visualització. Després d'importar el nostre volum, quan es genera una superfície el tipus d'objecte que es té és un *VtkPolyData*, per tal de visualitzar-ho de forma tridimensional, s'aprofitarà el visor 3D d'Starviewer amb algunes modificacions.

3.2 Integració de les VMTK

La instal·lació de les llibreries VMTK, l'eina bàsica d'aquest projecte, no va ser una tasca gens fàcil. Un dels problemes que tenia era que en ser quasi noves, no disposaven de gaire documentació. No obstant, per sort a la pàgina oficial hi havia un seguit de passos que ajudava la seva adquisició.

Primerament vam intentar instal·lar-les al sistema operatiu de Windows. Seguint les pautes vam arribar molt lluny, però quan ja quasi ho teníem enllestit, ens vam adonar que no podíem continuar amb la instal·lació. Vam buscar el perquè i si hi havia alternatives possibles. Llavors indagant vam descobrir que les *VMTK* no eren compatibles amb les llibreries *VTK's* que utilitzava l'Starviewer, ja que aquest té una versió més antiga. Vam arribar a pensar que no les podríem fer servir mai, ja que realitzar la traducció de tot Starviewer a una versió nova de les *VTK* suposaria molta feina que en aquells moments no estava prevista, ja que molts desenvolupadors estaven treballant en aquesta plataforma i es veurien afectats.

Llavors ens vam posar en contacte amb els creadors de les llibreries. Li vam exposar el nostre cas i ens van saber donar una solució.

La solució va ser canviar de sistema operatiu, ja que les VMTK's estan fetes servir majoritàriament sota Linux.

Crèiem que tindríem el mateix obstacle, perquè les VMTK seguien necessitant la versió que no tenim de les VTK, però al final ens va sorprendre el creador dient que en Linux no ens seria un problema.

Així doncs vam utilitzar el sistema operatiu Linux, concretament Kubuntu 9.04 sota KDE. Vam oblidar els passos que havíem seguit abans per fixar-nos en uns altres i ho vam aconseguir, les vam instal·lar.

Els passos per a la instal·lació són:

1. Afegir aquest repositori:

   sudo wget http://packages.simula.no/jaunty.list -o /etc/apt/sources.list.d/simula.list

2. Afegir la clau publica següent:

   wget http://packages.simula.no/pubring.gpg -O- | sudo apt-key add -

3. Actualitzar els paquets:

   sudo apt-get update

4. I instal·lar el paquet vmtk

   sudo apt-get install vmtk0-bin

Llavors s'ha de fer:

5. Descarregar i descomprimir l'últim release de vmtk de Sourceforge:

   tar -zxvf vmtk-0.x.tar.gz

6. Crear un directori per construir tot i entrar-hi.

   mkdir vmtk-build

   cd vmtk-build

7. Executar CMake amb el directori a on el codi font de les vmtk estan localitzades. Seguir les instruccions que diu i generarà els Makefiles per a la seva construcció.

   ccmake vmtk-source-directory

8. Compilar i instal·lar vmtk. Assegurar-se que es té els permisos necessaris per instal·lar.

   make

   make install

Llavors s'hauria de deixar el ccmake d'aquesta manera:

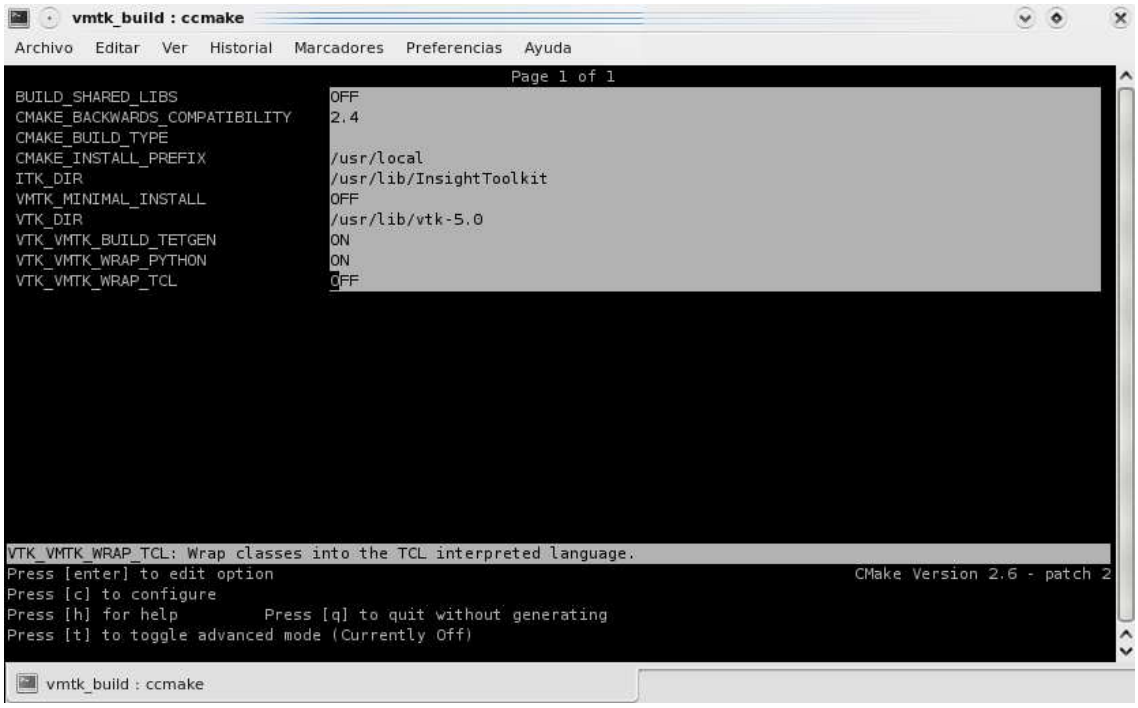


Figura 20. Opcions del CCMake durant la instal·lació vmtk.

Pressionar "*c*" per configurar i llavors "*g*" per generar. I finalment posar *make install.*

Un cop fet tot això les vmtk ja estaran instal·lades i llestes per a fer servir.

3.3 Integrar VMTK a Starviewer

Ara ja tenim les VMTK instal·lades al nostre sistema però per tal que Starviewer aconsegueixi detectar-les s'han de modificar un seguit de fitxers del seu codi font:

1. Afegir al fitxer ***defaultdirectories.inc*** la localització de les llibreries al nostre ordinador. Com que de moment les VMTK només funcionen a Linux, la única opció serà "unix: ...", sense la de Windows ni Mac:

```
# VMTK Libraries

VMTKLIBDIR = $$(VMTKLIBDIR)
isEmpty(VMTKLIBDIR){
  unix:VMTKLIBDIR = /usr/lib
}

VMTKINCLUDEDIR = $$(VMTKINCLUDEDIR)
isEmpty(VMTKINCLUDEDIR){
  unix:VMTKINCLUDEDIR = /usr/local/include/vmtk
}
```

Figura 24. Afegir aquestes línies a starviewer/core/defaultdirectories.inc

2. Afegirem un fitxer nou anomenat ***vmtk.inc*** apuntant a on es troben els arxius *.so / *.a de la nostra llibreria nova.

```
include(defaultdirectories.inc)

INCLUDEPATH += $${VMTKINCLUDEDIR}

LIBS += -L$${VMTKLIBDIR} \
    -lvtkvmtkCommon \
    -lvtkvmtkCommonPythonD \
    -lvtkvmtkComputationalGeometry \
    -lvtkvmtkComputationalGeometryPythonD \
    -lvtkvmtkDifferentialGeometry \
    -lvtkvmtkDifferentialGeometryPythonD \
    -lvtkvmtkIO \
    -lvtkvmtkIOPythonD \
    -lvtkvmtkITK \
    -lvtkvmtkITKPythonD \
    -lvtkvmtkMisc \
    -lvtkvmtkMiscPythonD \
    -lvtkvmtkSegmentation \
    -lvtkvmtkSegmentationPythonD
```

Figura 25: Creació fitxer starviewer/core/vmtk.inc

3. Finalment per tal que cada extensió, les detecti automàticament, s'ha d'afegir el seu include al fitxer ***basicconfextensions.inc***.

```
TRANSLATIONS += $$EXTENSION_DIR/translations_ca_ES.ts \
                $$EXTENSION_DIR/translations_es_ES.ts \
                $$EXTENSION_DIR/translations_en_GB.ts

DESTDIR = ./

INCLUDEPATH += ../../../core
DEPENDPATH += ../../../core
MOC_DIR = ../../../../tmp/moc
UI_DIR = ../../../../tmp/ui
OBJECTS_DIR = ../../../../tmp/obj
RCC_DIR = ../../../../tmp/rcc
CONFIG += release \
          warn_on \
          qt \
          opengl \
          thread \
          x11 \
          staticlib \
          exceptions \
          stl
TEMPLATE = lib

include(../vtk.inc)
include(../itk.inc)
include(../dcmtk.inc)
include(../vmtk.inc)
include(../log4cxx.inc)
include(../compilationtype.inc)
```

Figura 26. Afegir include a starviewer/src/extensions/basicconfextensions.inc

Tot això serà suficient per a poder cridar qualsevol funció de les llibreries VMTK des de qualsevol classe de la plataforma Starviewer.

No obstant, Starviewer actualment no està pensat per a poder suportar phyton i recordem que quasi totes les classes útils que volem fer servir d'aquesta llibreria són scripts implementats en phyton. Per tant, la nostra tasca serà traduir els scripts necessaris a c++ per a poder adaptar-lo a l'extensió.

CAPÍTOL 4

Implementació. Visió General del problema

Abans d'entrar en detall en l'implementació volem donar una visió general del problema que cal resoldre per tenir clars els mòduls que hem d'implementar per assolir els nostres objectius.

El primer que cal fer és carregar el model del volum per tal d'obtenir les imatges en 2D. Gràcies als anteriors desenvolupadors de la plataforma, aquest pas ja és una opció que el mateix Starviewer proporciona.

El segon bloc és la segmentació vascular. Per tal de realitzar-ho usarem un sistema basat en la reconstrucció de superfícies, el marching cubes, com s'explica a l'apartat 2.2 del capítol 2.

A continuació una vegada hem reconstruït el model de superfície que representa les venes haurem d'obtenir el seu esquelet, per això usarem el mètode de les centerlines explicat a l'apartat 2.3 del capítol 2.

Finalment, visualitzarem el model i li donarem l'oportunitat de prendre mesures sobre ell (això has de dit com ho faràs) encara no sé ben be com

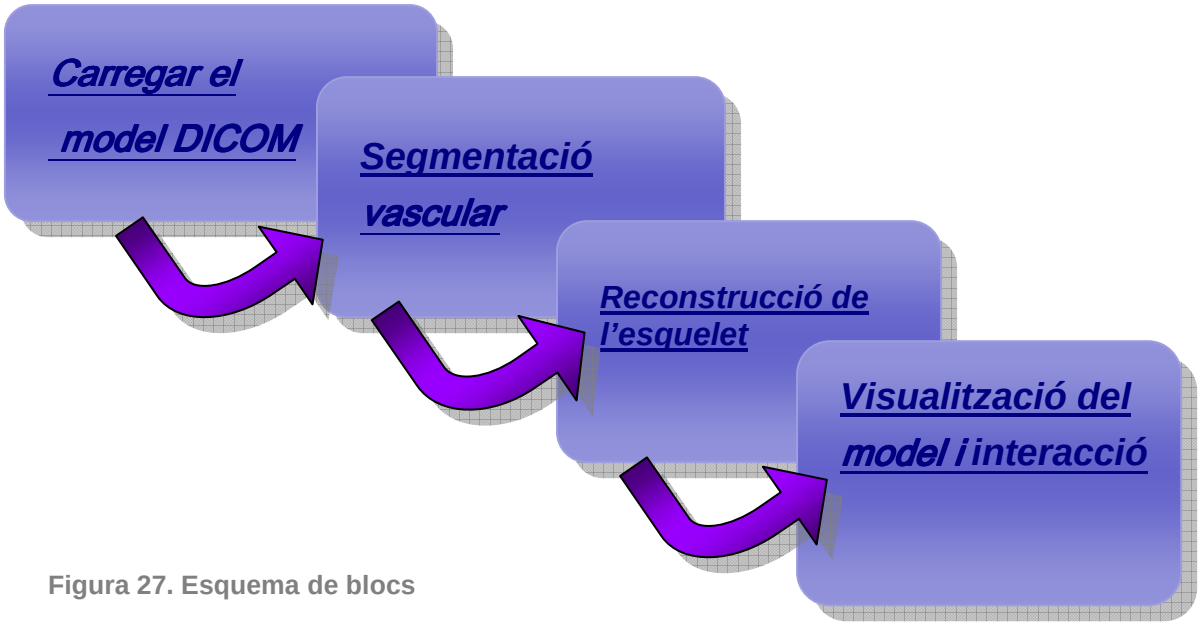


Figura 27. Esquema de blocs

Tots els mòduls que implementarem els integrarem en un extensió de l'Starviewer que anomenarem ***Vascularmodeling***

CAPÍTOL 5

Segmentació

En aquest capítol es mostra la part de la implementació de la segmentació. En els primers apartats tenim els anàlisis i requeriments amb una explicació detallada a partir de diagrames de casos d'ús i seqüència. Després apuntem com és el disseny empleat per realitzar la seva implementació.

5.1 Anàlisi de requeriments

5.2 Requeriments funcionals

A partir del volum que tenim de la informació de la ARM, el que voldrà l'usuari és poder interaccionar amb una superfície en tres dimensions del sistema vascular únicament. L'objectiu de l'extensió és mesurar el diàmetre de la caròtida per a poder diagnosticar l'estenosi.

Primerament l'usuari veurà el volum bidimensional, i llavors haurà de marcar la opció per a poder veure la segmentació en tres dimensions realitzat a partir del marching cubes. L'usuari haurà de buscar el nivell de vòxel màxim a partir de l'eina d'informació de vòxel a l'extensió de l'Starviewer i introduir-ho en un apartat especial per tal que el programa agafi aquest marge i faci tot el càlcul. Apareixerà al segon visor, la reconstrucció tridimensional del sistema vascular.

5.3 Diagrames de casos d'ús

A continuació els casos d'ús ens ajudaran a comprendre la comunicació i el comportament del sistema a partir de la seva interacció amb els usuaris. Ens il·lustraran els requeriments del sistema en mostrar com reacciona una resposta a events que es produeixen en el mateix.

Tal i com hem vist en l'esquema de la figura 24 en el que donàvem una visió global del problema el primer que ens cal es segmentar el sistema vascular. Per fer-ho usarem l'algorisme de ***marching cubes***. Aquesta tècnica permet reconstruir la superfície que passa per tots els punts que tenen un determinat valor de propietat anomenat isovalor. En el nostre cas per les característiques de les nostres imatges de ARM i pel fer que s'ha injectat contrast al pacient aquest valor coincideix amb el valor més alt de propietat. Tenint en compte això:

L'usuari ha de marcar el valor de vòxel més alt per a que el sistema pugui aplicar l'algorisme de marching cubes a partir d'aquest isovalor.

Nom	***Generació de la superfície 3D***
Actor	Usuari
Precondició	Haver obert un directori DICOM d'una exploració dinàmica
Flux Principal	1. Buscar valor màxim de vòxel amb l'eina informació del vòxel. 2. Escriure valor màxim. 3. Activar Marching Cubes.
Flux alternatiu	
Postcondició	Visualitzar la generació de la superfície 3D del sistema vascular.
Comentaris	

5.3.1 Diagrama de cas d'ús

La figura 28 mostra el diagrama de cas d'ús de la generació de la superfície 3D.

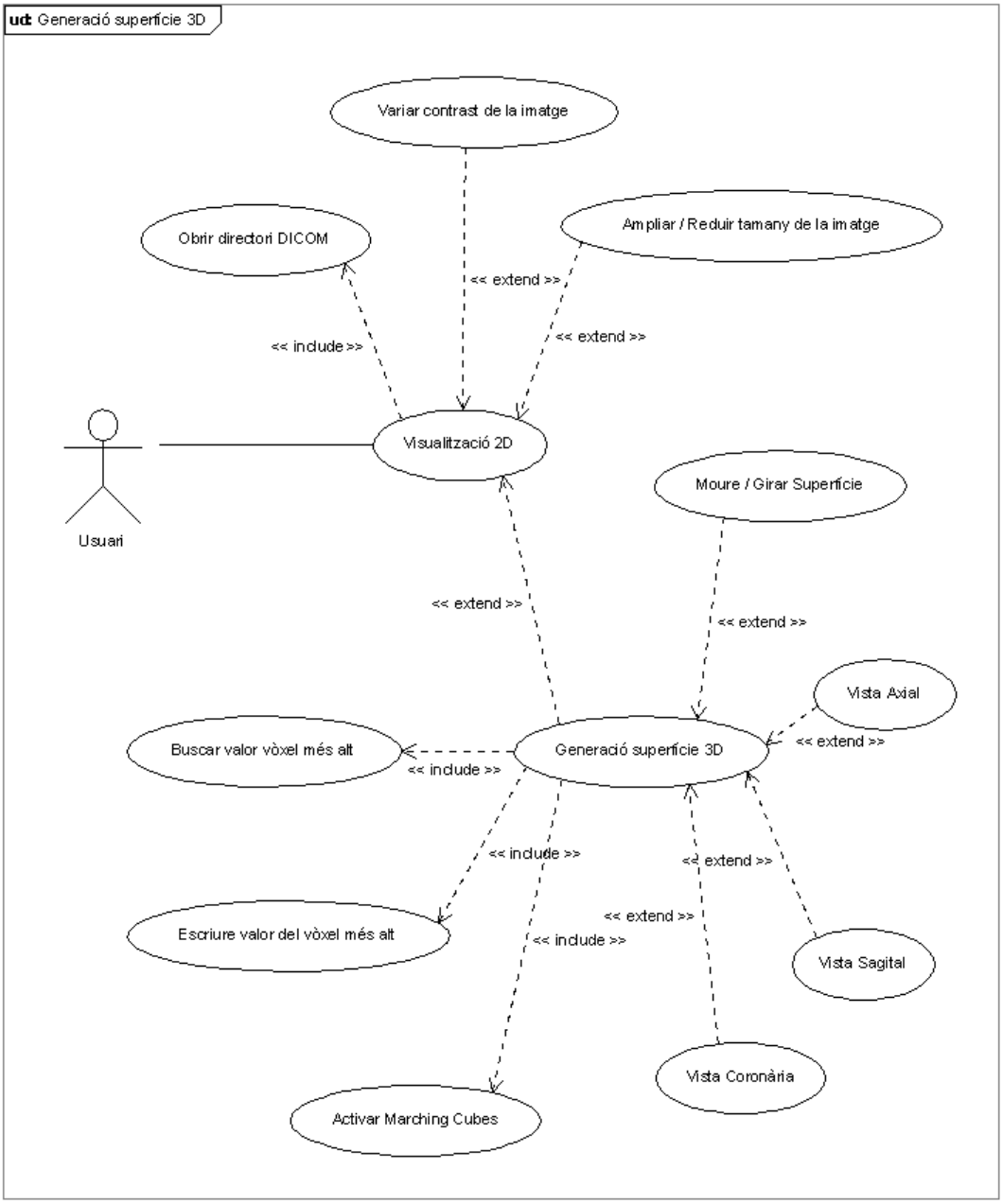


Figura 28. Diagrama de cas d'ús de generació de superfície 3D.

5.3.2 Diagrama de seqüència

La figura 29 mostra el diagrama de seqüència de la generació de la superfície 3D.

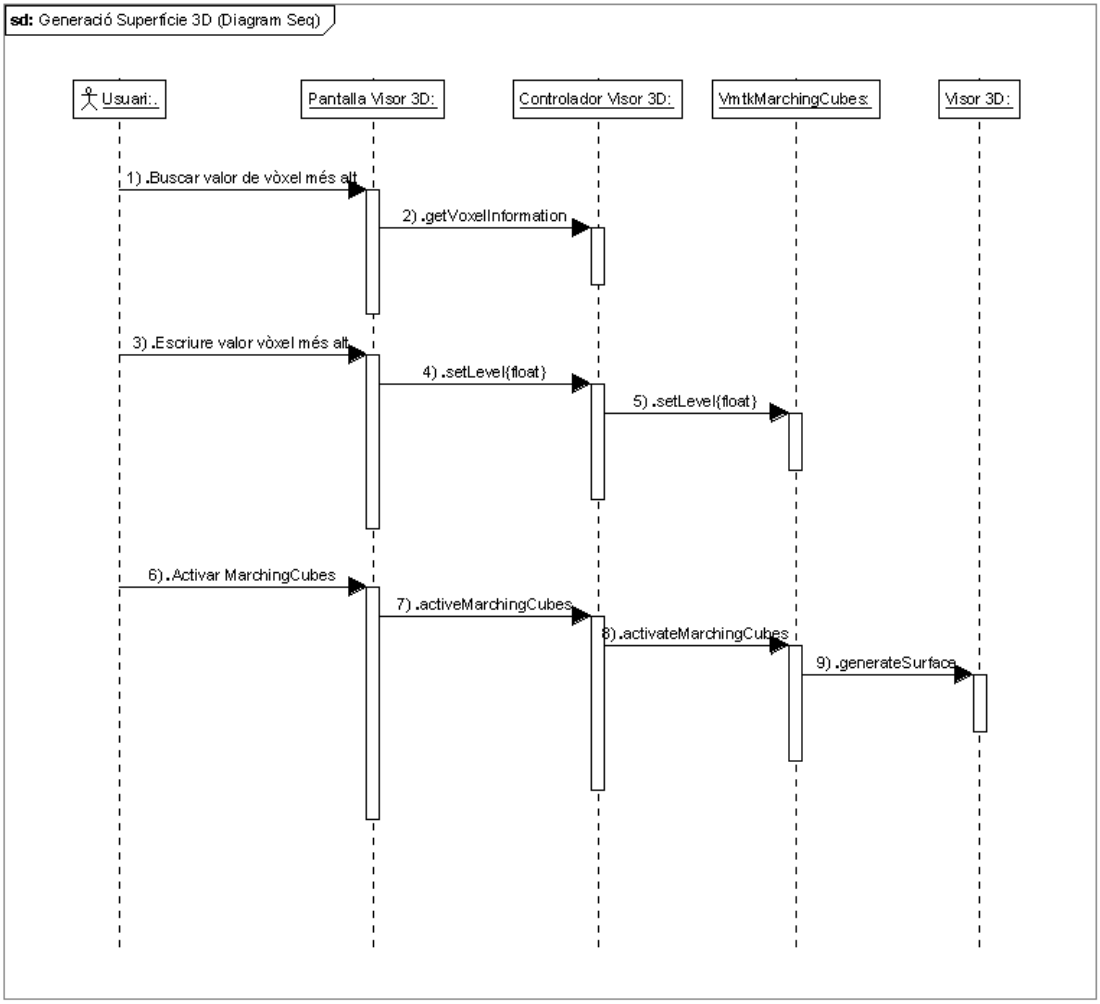


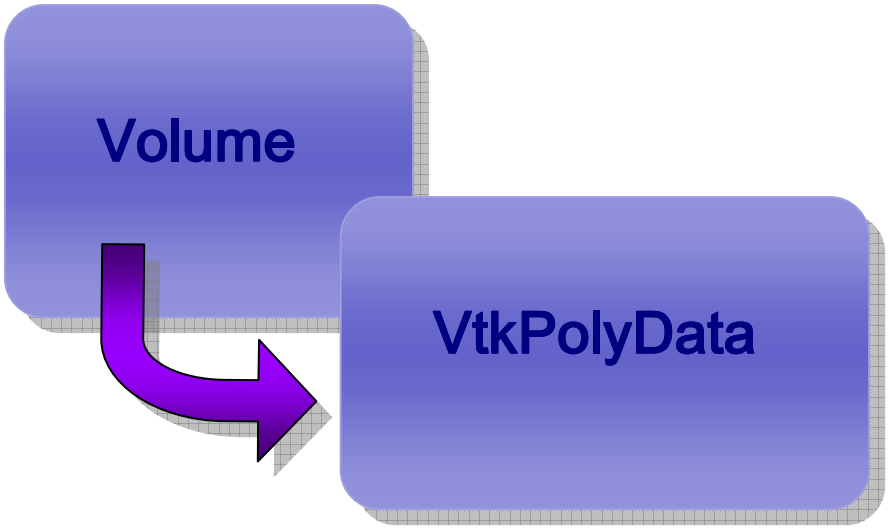
Figura 29. Diagrama de seqüència de generació de la superfície 3D.

5.4 Disseny i implementació

Com hem dit abans la nostra extensió s'anomenarà ***vascularmodeling***.

Per integrar el mètode de marching cubes en l'extensió hem usat les VMTK. Aquestes llibreries disposen d'un script implementat en phyton anomenat ***vmtkmarchingcubes.py*** que ens realitza la tasca de la segmentació a on ens identificarà el sistema vascular.

El funcionament d'aquest script és que a partir d'un ***DICOM*** transforma el volum mitjançant el mètode de marching cubes a un formant ****.vtp***, que és l'extensió de ***vtkPolyData.***



Donat que Starviewer no suporta phyton hem hagut de traduir el mètode a C++ i fer diferents ajustament per tal de poder-lo adaptar a la plataforma.
El mètode que hem implementat l'hem anomenat ***vascularmodelingvmtkmarchingcubes*** que és a on hi ha la part de la segmentació i generació de la isosuperfície

Per poder mostrar la informació que genera el mètode l'hem hagut d'ajustar a la plataforma. Com s'ha explicat anteriorment, un visor està format per tres classes: la interfície perquè l'usuari interactuï amb el visor; el visor que és la classe que manipula el model i la classe que executa les peticions de l'usuari.

Diagrama de classes

Podem veure un diagrama de classes simplificat d'aquest mòdul a la figura 30.

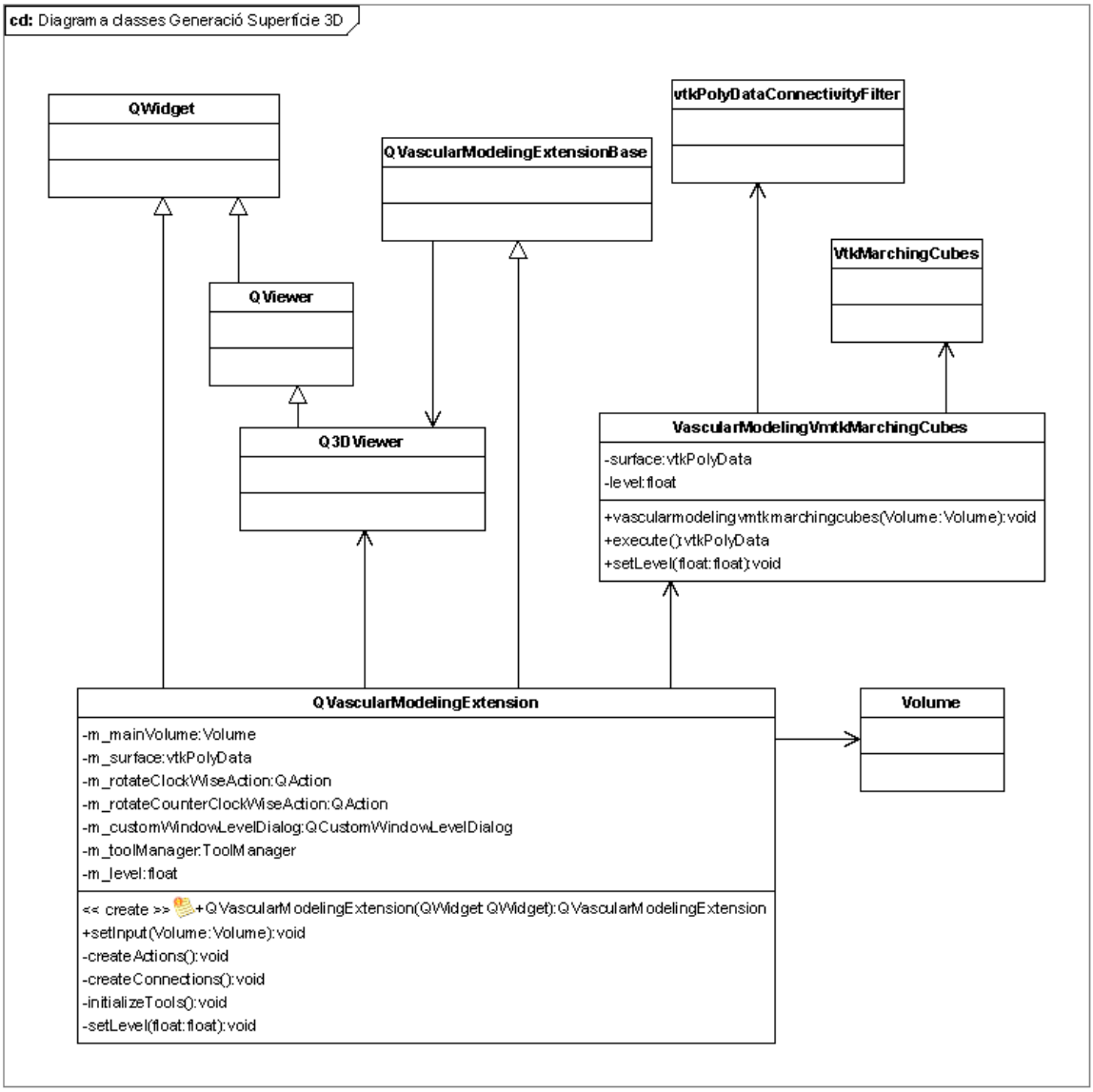


Figura 30. Diagrama de classes simplificat de la generació de superfície 3D.

El que fa la nostra classe ***vascularmodelingvmtkmarchingcubes*** és convertir el volum que tenim original en 2D a ***VtkPolyData***. Per tal de dur-ho a terme, es fa servir l'algorisme ja esmentat a l'apartat 2.2.1 del capítol 2, el Marching Cubes, gràcies a la classe implementada de les ***VTK*** que porta el mateix nom: ***vtkMarchingCubes***.

Es pot veure la seva procedència a partir dels dos gràfics següents:

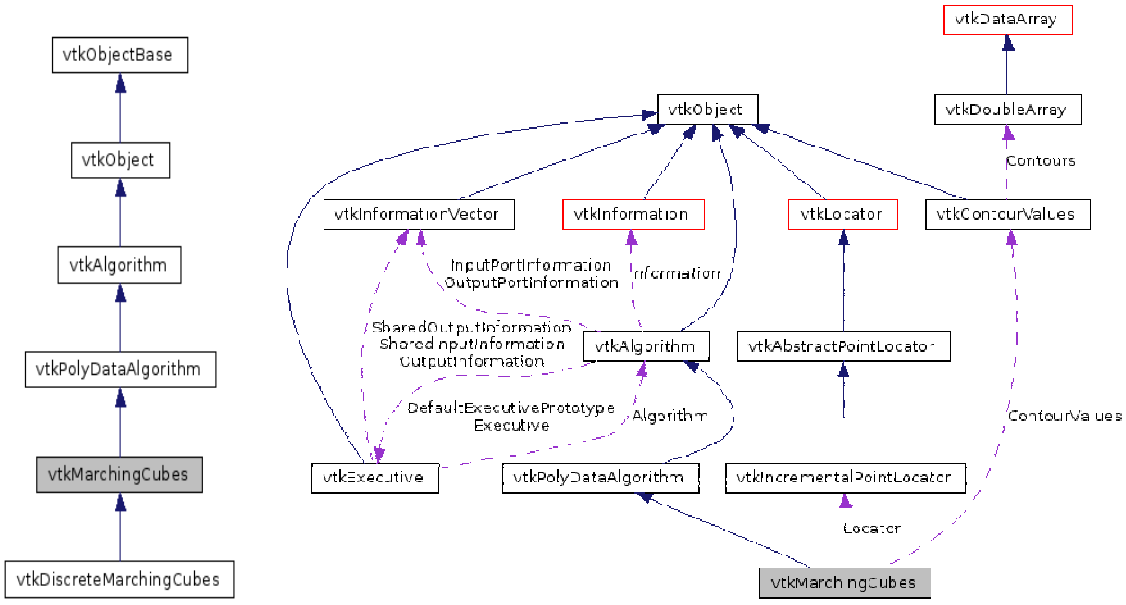


Figura 31. Diagrama d'herències i de col·laboració de vtkMarchingCubes.

El que necessita no obstant, és un valor de tipus ***float*** que l'anomenem ***level***. Aquí és a on l'usuari ha d'intervenir per tal de proporcionar al sistema el valor màxim de vòxel que hi ha a la imatge en 2D.

El que ha de fer, és mirar els valors que tenen els vòxels de les artèries amb l'eina que ens procura l'Starviewer, el voxel information, i determinar quin és el valor més alt. Llavors l'ha de posar a la casella corresponent i el sistema l'agafarà i el modificarà a partir de la funció ***setLevel(float).***

Després d'obtenir el ***VtkPolyData***, les ***VMTK*** s'ajuden d'un filtre connector, per tal d'aconseguir una superfície més neta. Utilitzen la classe de les ***VTK vtkPolyDataConnectivityFilter*** (figura 32).

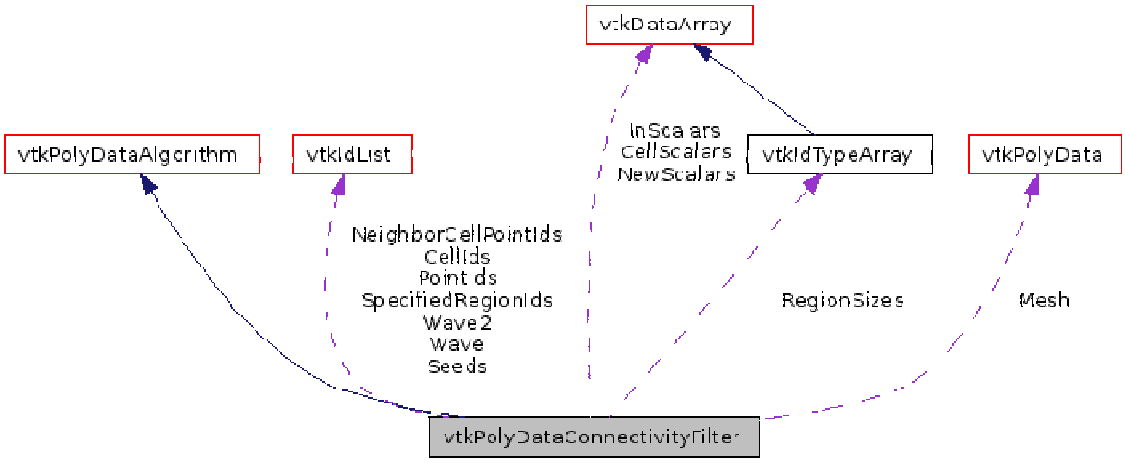
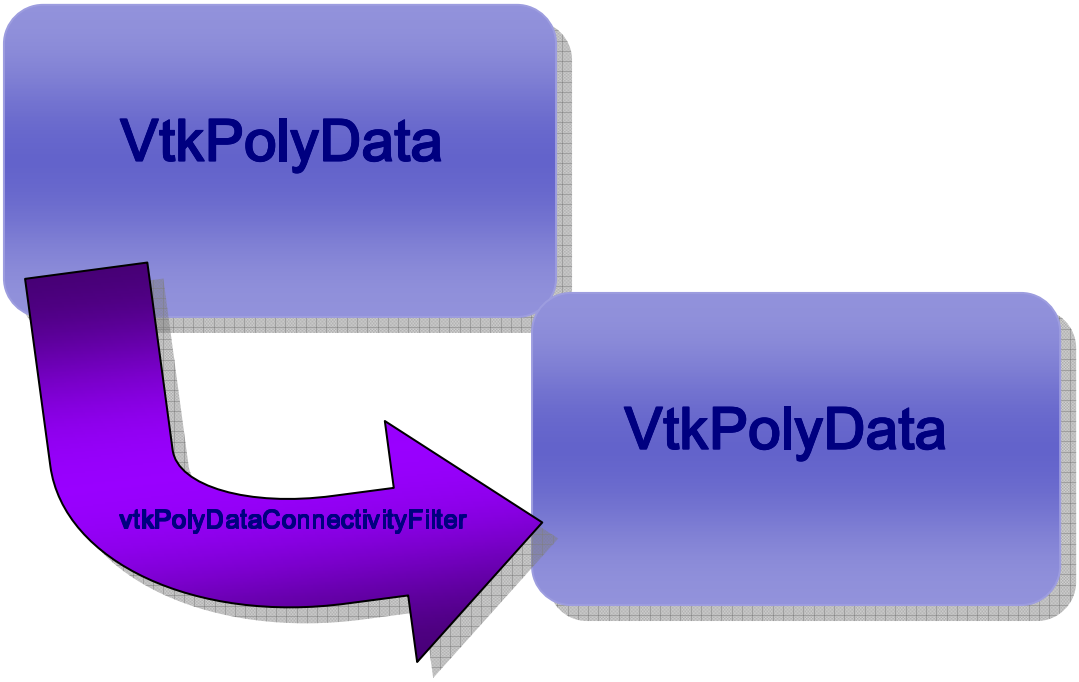


Figura 32. Diagrama d'herències i de col·laboració de vtkPolyDataConnectivityFilter.

Finalment obtenim la superfície refinada, que és la que el sistema visualitzarà amb el visor 3D.



Pseudocodi

El pseudocodi de la classe ***vascularmodelingvmtkmarchingcubes*** implementada per nosaltres llavors seria:

```
vtkMarchingCubes *m_cube = new vtkMarchingCubes()
    m_cube->SetInput(vol->getVtkData())
    m_cube->SetValue(0,level)
    m_cube->Update()
    Surface = m_cube->GetOutput()
```

I a continuació li passem el filtre de connectivitat:

```
vtkPolyDataConnectivityFilter *connectivityFilter = new vtkPolyDataConnectivityFilter()
    connectivityFilter->SetInput(Surface);
    connectivityFilter->SetExtractionModeToLargestRegion();
    connectivityFilter->Update();
    Surface = connectivityFilter->GetOutput();
```

Si l'usuari segueix els passos correctament, el sistema generarà superfícies com aquesta:

Figura 33. Exemple de resultat de segmentació del sistema vascular amb marching cubes

CAPÍTOL 6

Reconstrucció

En aquest capítol es mostra la part de la implementació de la reconstrucció. En els primers apartats tenim els anàlisis i requeriments amb una explicació detallada a partir de diagrames de casos d'ús. Després apuntem com és el disseny empleat per realitzar la seva implementació.

6.1 Anàlisi de requeriments

6.2 Requeriments funcionals

Un cop tenim la superfície, ens interessa poder extreure l'esquelet del model. El primer pas per reconstruir el l'esquelet és extreure els centerlines a partir d'un algorisme d'anàlisis topològic. L'usuari haurà de buscar els punts per on voldrà que comencin les línies centerlines i els punts finals per on voldrà que acabi. Llavors ja obtindrà la reconstrucció de tres dimensions amb l'esquelet marcat a sobre la superfície.

6.3 Diagrames de casos d'ús

El bloc que segueix la segmentació a l'esquema de la figura 24 a on hi ha la visió global del problema, és el bloc de reconstrucció de l'esquelet. Per fer-ho utilitzarem un algorisme d'anàlisis topològic que s'explica a l'apartat 2.3 del capítol 2. Aquesta tècnica permet reconstruir un esquelet a partir de la isosuperfície que ja disposem generada per el marching cubes. Però necessita la interacció de l'usuari ja que ha de decidir quins punts tindran importància a la hora d'obtenir l'esquelet. Tenint en compte això:

L'usuari ha de marcar els punts a on vol que comenci el recorregut del centerline (Source Points) i llavors els punts a on voldrà que acabi (Target Points). Llavors el sistema ja podrà computar els centerlines.

El sistema fa un seguit de càlculs complexos i llavors visualitza per pantalla la superfície juntament amb el seu centerlines.

Nom	*Computació de les centerlines*
Actor	Usuari
Precondició	Haver generat una superfície 3D.
Flux Principal	1. Marcar els punts que seran l'inici del centerline. 2. Marcar el punt a on acabarà el centerline. 3. Activar Centerlines.
Flux alternatiu	
Postcondició	Computació de les centerlines de la isosuperfície en tres dimensions.
Comentaris	

Diagrama de cas d'ús

La figura 34 mostra el diagrama de cas d'ús de la computació de les centerlines.

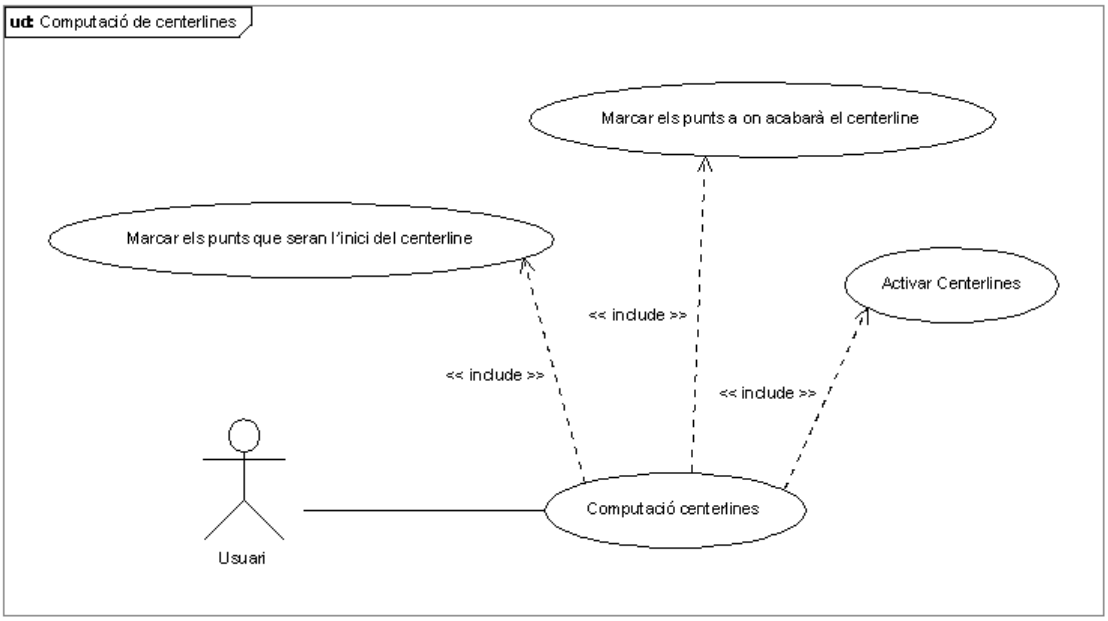


Figura 34. Diagrama de cas d'us de computació de centerlines.

6.4 Disseny i implementació

Per tal de computar els centerlines ens hem dedicat a mirar el funcionament de la classe ja implementada en ***phyton*** anomenada ***vmtkcenterlines.py***.

Aquesta classe és força complexa ja que intervenen moltes classes de les ***VTK*** i de les pròpies ***VMTK*** implementades en ***c++*** i en ***phyton***.

Ens ajudarem d'un diagrama simplificat de l'estructura del mòdul de les centerlines per entendre el seu funcionament (figura 35).

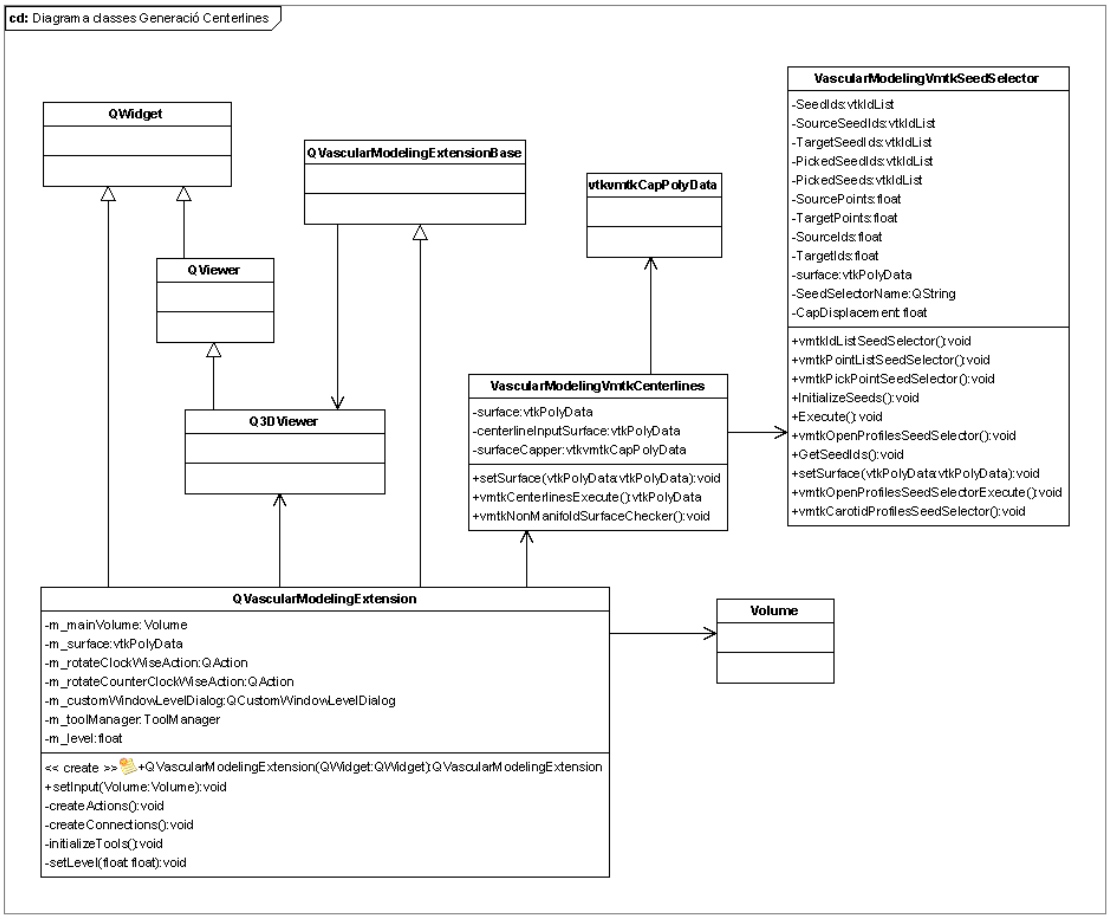


Figura 35. Diagrama de classes simplificat de la generació de centerlines sobre una superfície en 3D.

La base de la complicació de l'algorisme, era entendre quines eren les classes que utilitzava i lligar-ho tot de manera que quedi ben assemblat.

El diagrama de classes simplificat de la figura 36 ajuda a entendre les dependències que té amb les ***VTK*** i ***VMTKs*** la nostra classe implementada ***vascularmodelingcenterlines***.

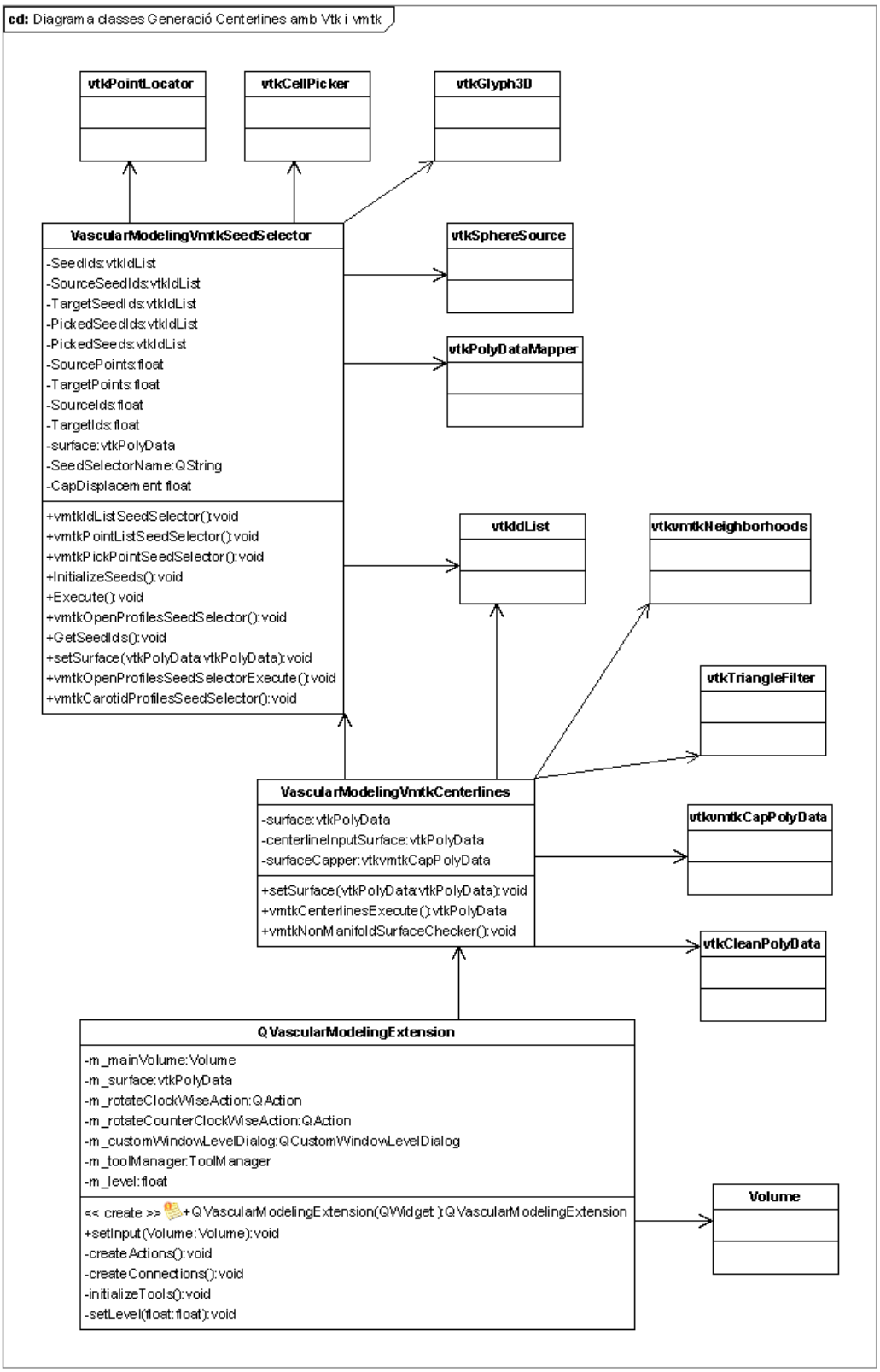
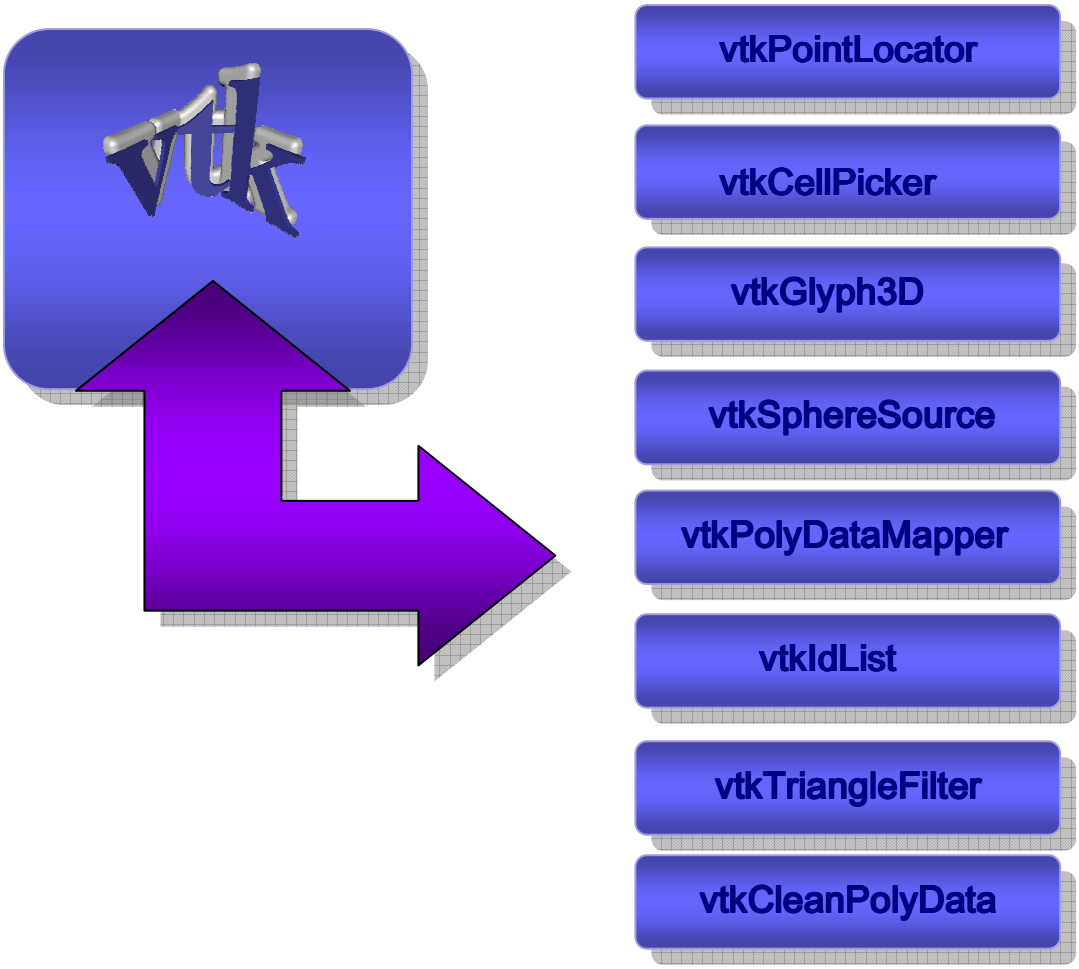
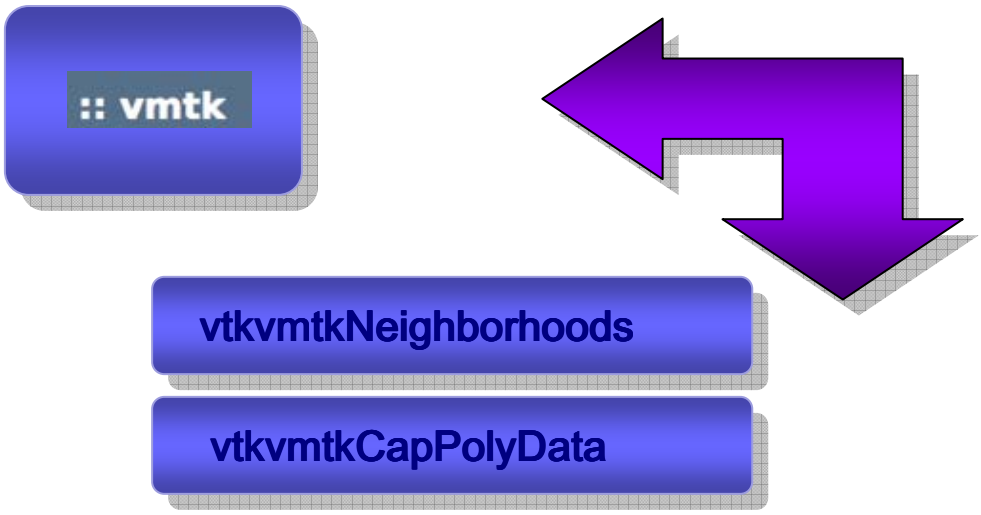


Figura 36. Diagrama de classes de generació de centerlines amb vtk i vmtk.

Com es pot veure al diagrama les classes que utilitza de les VTK són:



De les classes implementades en c++ de les VMTKs fa servir:



A continuació veurem uns apunts d'aquestes classes juntament amb els seus diagrames de col·laboració per a poder entendre millor la seva utilitat:

vtkPointLocator

És un objecte de cerca espacial que ajuda a localitzar punts en una superfície en 3D ràpidament. Treballa dividint una regió específica en una ***array*** regular de cubs i llavors manté una llista de punts que està dins de cada cub. L'operació típica et dóna una posició en 3D i buscar el seu punt més proper. Es pot observar el seu diagrama de col·laboració a la figura 37.

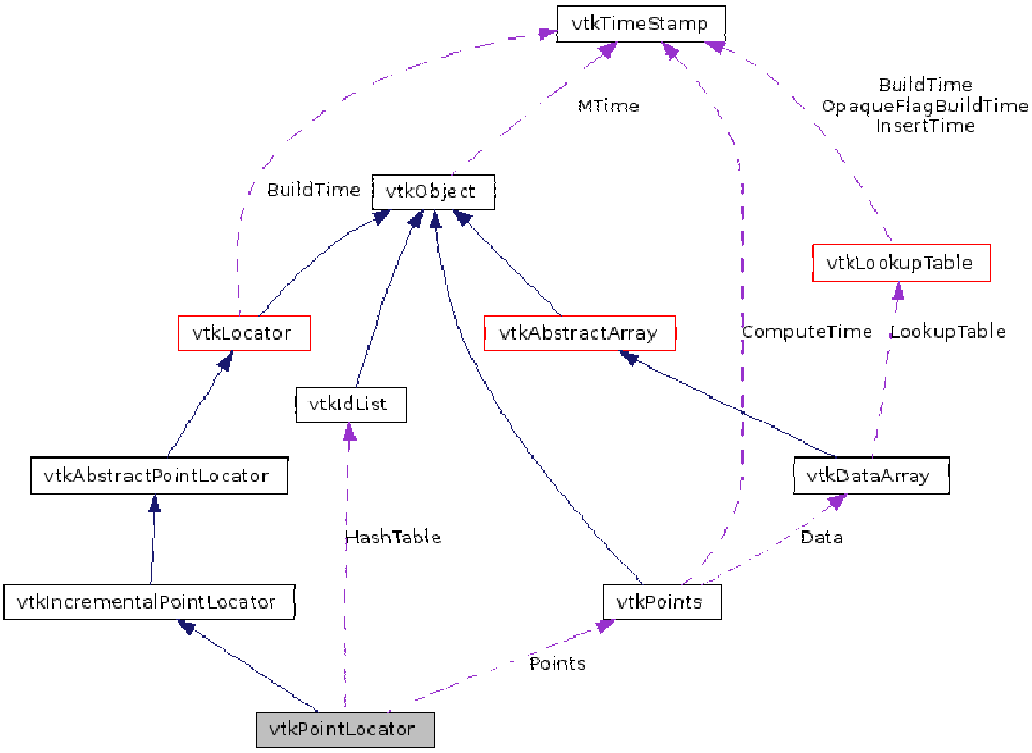


Figura 37. Diagrama de col·laboració de vtkPointLocator

vtkCellPicker

Serveix per a seleccionar una cel·la a partir de disparar un raig a la finestra dels gràfics i intersectant amb la geometria definida de l'actor (específicament a les seves cel·les). A més de retornar les ***coordenades***, ***actor*** i ***mapper***, també retorna la ***id*** de la cel·la més propera segons la tolerància del raig. Podem veure els seus diagrames a la figura 38.

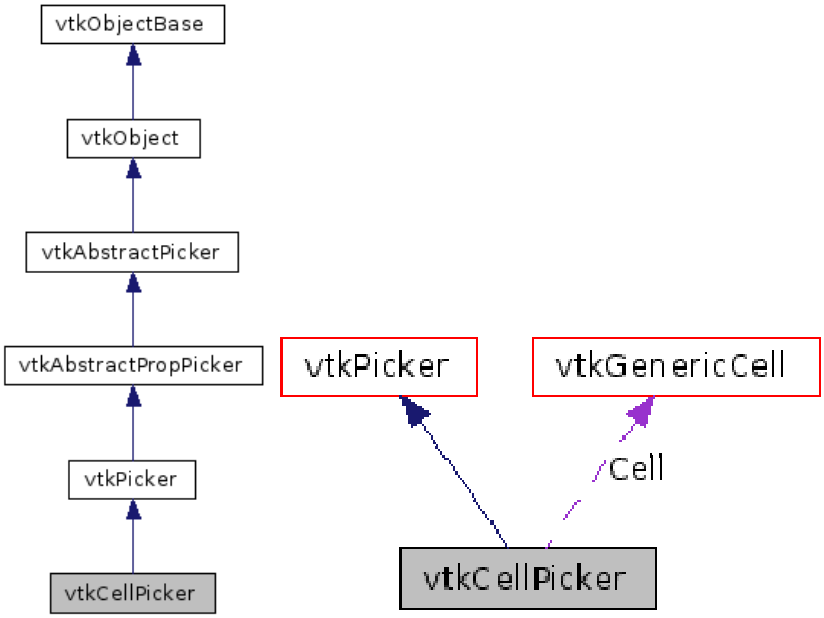


Figura 38. Diagrama d'herències i de col·laboració de vtkCellPicker

vtkGlyph3D

És un filtre que copia una representación geomètrica, anomenada ***glyph***, a cada punt de la información d'entrada.

El ***glyph*** és definit amb información poligonal a partir d'un filtre font entrat amb una funció. El ***glyph*** pot estar orientat al llarg dels vectors entrats o de normals, i pot ser escalat segons la informació d'escala o vector de magnitud. Poden usar-se més d'un ***glyph*** a si es creen una taula d'objectes fonts i a cadascun es defineixen un ***glyph*** diferent. Si la taula de ***glyphs*** són definits, llavors la taula pot ser indexada a partir d'un valor escalar o un vector de magnitud. La figura 39 són els seus diagrames de col·laboració i d'herencies.

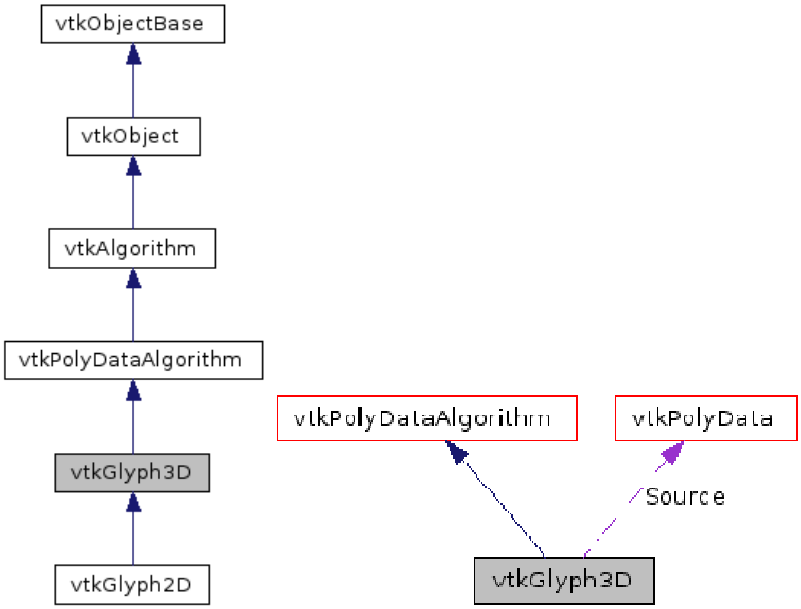


Figura 39. Diagrama d'herencies i de col·laboracó de la classe vtkGlyph3D.

vtkSphereSource

Aquesta classe el que fa es crear una esfera representada per polígons d'un determinat radi centrat a l'origen. La resolució, és a dir, la discretització poligonal, en les direccions de latitud i longitud poden ser especificades. Per defecte la superfície de tessel de les esferes utilitza triangles, no obstant es pot canviar a quadrilàters. Aquesta esfera serà la nostra esfera investigadora que apuntàvem a l'apartat 2.3 del capítol 2. Podem veure el diagrama de col·laboració d'aquesta classe a la figura 40.

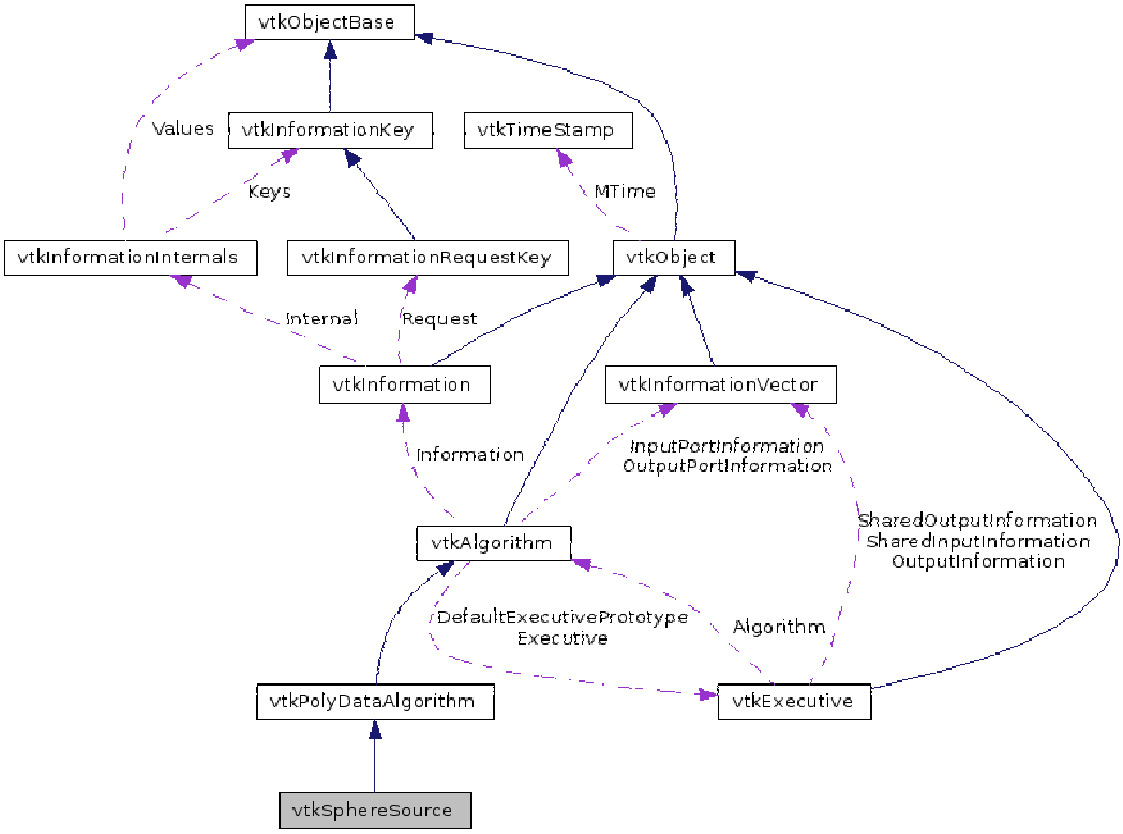


Figura 40. Diagrama de col·laboració de vtkSphereSource

vtkPolyDataMapper

vtkPolyDataMapper és una classe que mapeja informació poligonal com per exemple vtkPolyData a gràfics primitius. Serveix com una superclasse que mapeja renders/gràfics. Tenim el seu diagrama a la figura 41.

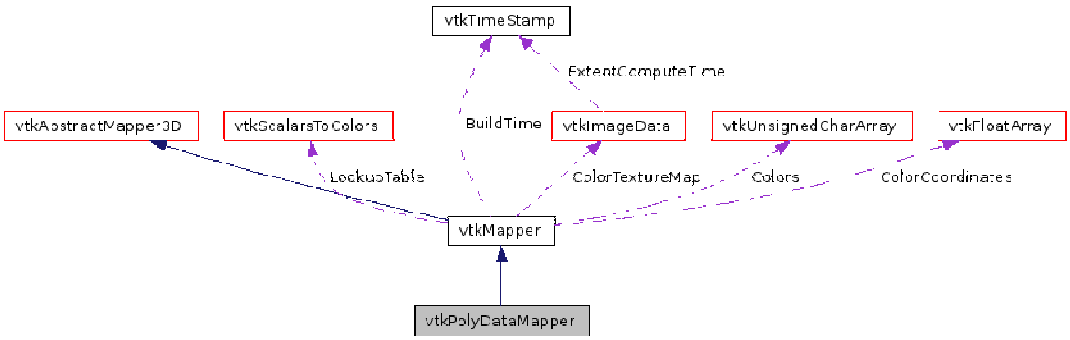


Figura 41. Diagrama de col·laboració de vtkPolyDataMapper

vtkIdList

Serveix per representar i passar informació de id's entre objectes. Es poden representar qualsevol tipus de id d'enter, però normalment representa punts i cel·les identificadores. Podem observar el seu diagrama d'herències i col·laboració a la figura 42.

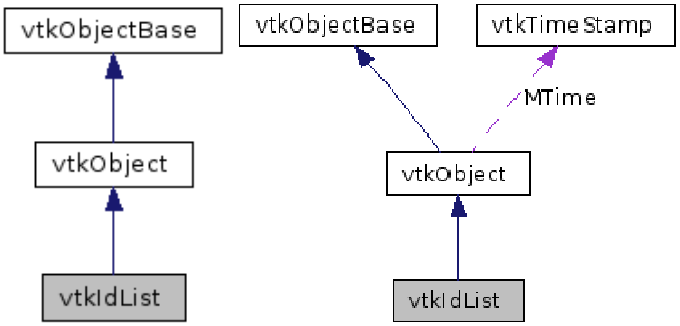


Figura 42. Diagrama d'herències i col·laboració de vtkIdList.

vtkTriangleFilter

El que fa aquesta classe és generar triangles a partir de polígons d'entrada i franges de triangles. El filtre també passarà a través de vèrtexs i línies si és vol. El seu diagrama de col·laboració el tenim a la figura 43.

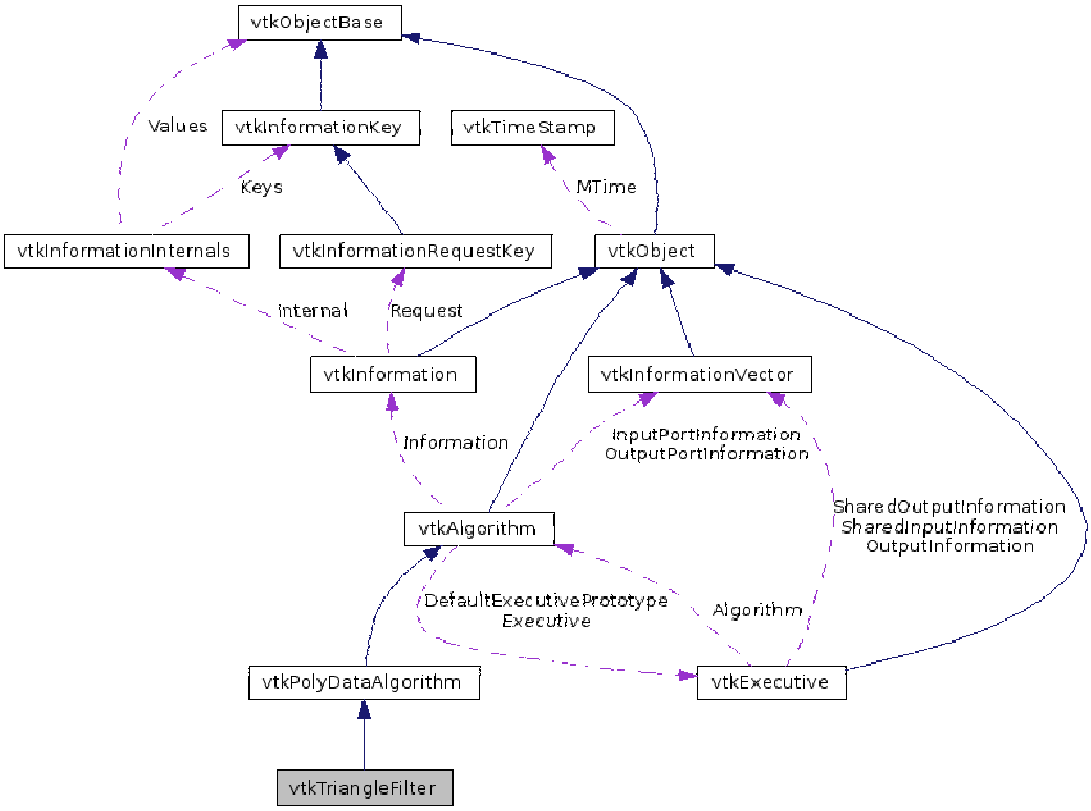


Figura 43. Diagrama de col·laboració de vtkTriangleFilter.

vtkCleanPolyData

vtkCleanPolyData és un filtre que agafa informació poligonal d'entrada i genera una informació poligonal de sortida. El que fa aquesta classe és unir punts duplicats (segons una tolerància especificada i si està activada aquesta opció), eliminar els punts que no es fan servir i si el desenvolupador vol, transformar cel·les degenerades en formes apropiades. Per exemple un triangle convertit en una línia si dos punts i un triangle són units.

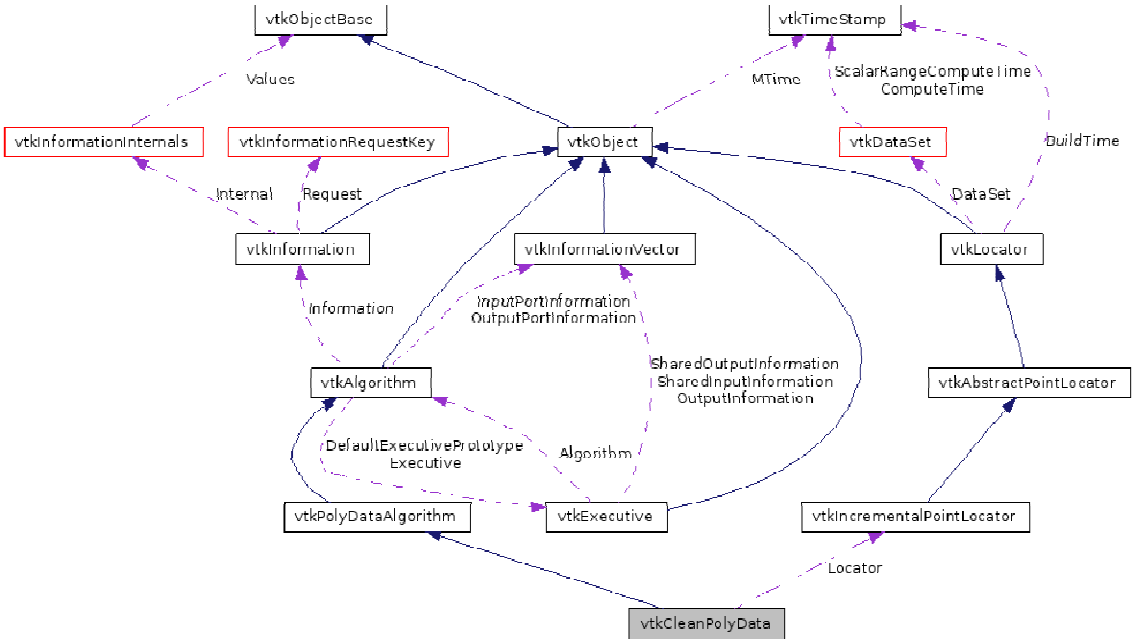


Figura 44. Diagrama de col·laboració de vtkCleanPolyData

A continuació explicarem les classes implementades en c++ de les VMTK.

vtkvmtkNeighborhoods

Aquesta classe serveix per a poder trobar punts més propers o veïns a partir d'una superfície entrada i una llista de punts. Té funcions com ***GetNumberOfPoints*** per tal de retornar la llista de tots els punts o ***GetNeighborhood(i)*** que a partir d'un punt entrat per paràmetre et retorna un altre punt veí.

vtkvmtkCapPolyData

Les VMTK utilitzen aquesta classe per afegir limitacions a les fronteres, per tal de poder fer l'algorisme de centerlines descrit a l'apartat 2.3 del capítol 2.

Podem veure un exemple del resultat:

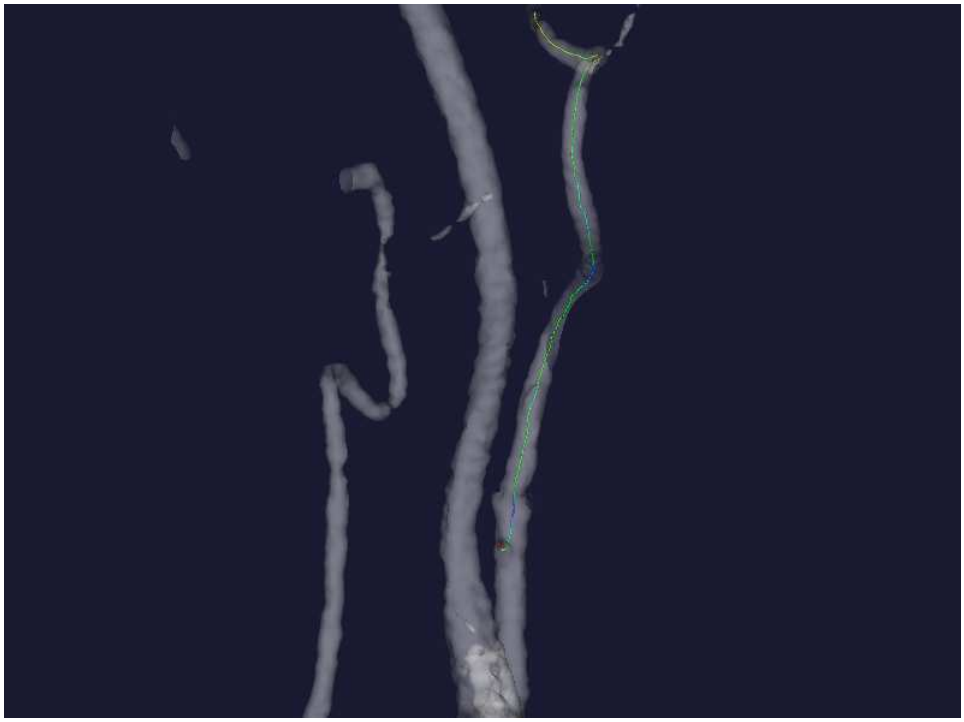


Figura 45. Exemple de l'esquelet amb el centerline.

CAPÍTOL 7

Visualització

En aquest capítol es mostra la part de la implementació de la visualització. En els primers apartats tenim els anàlisis i requeriments amb una explicació detallada a partir de diagrames de casos d'ús i seqüència. Després apuntem com és el disseny empleat per realitzar la seva implementació.

7.1 Anàlisi de requeriments

7.2 Requeriments funcionals

Un cop tenim la superfície juntament amb el model de l'esquelet, ens interessa poder mostrar per pantalla aquesta superfície en 3D.

7.3 Diagrames de casos d'ús

El bloc que segueix la reconstrucció de l'esquelet a l'esquema de la figura 24 a on hi ha la visió global del problema, és el bloc de la visualització del model. Per fer-ho aprofitarem el visor 3D ja implementat a Starviewer amb algunes modificacions.

L'usuari ha d'activar la opció de visualització de la superfície 3D amb l'esquelet.

Nom	***Visualització del sistema vascular en 3D***
Actor	Usuari
Precondició	Haver generat una superfície 3D amb l'esquelet de centerlines.
Flux Principal	1. Activar la visualització de la superfície en 3D.
Flux alternatiu	
Postcondició	Visualització pel visor 3D del sistema vascular amb el seu esquelet de centerlines.
Comentaris	

Diagrama de cas d'ús

La figura 46 mostra el diagrama de cas d'ús de la visualització del sistema vascular en 3D.

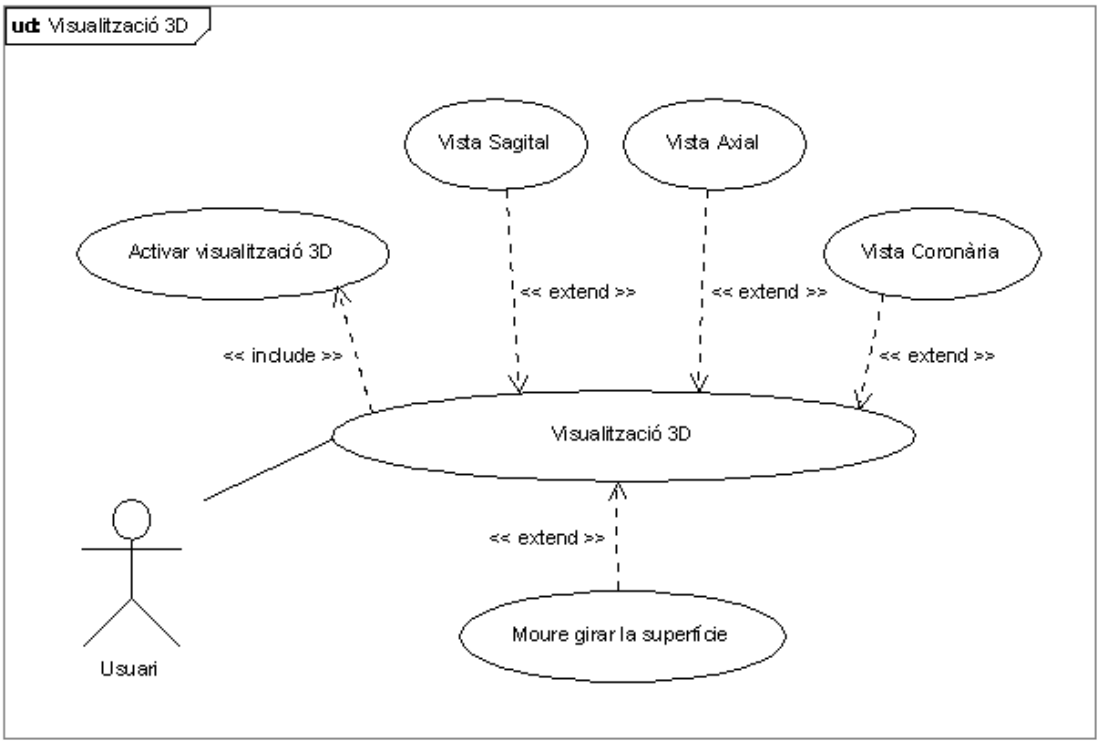


Figura 46. Diagrama de cas d'us de la visualització del sistema vascular en 3D.

7.1.1 Diagrama de seqüència

La figura 47 mostra el diagrama de seqüència del visualitzador de superfícies.

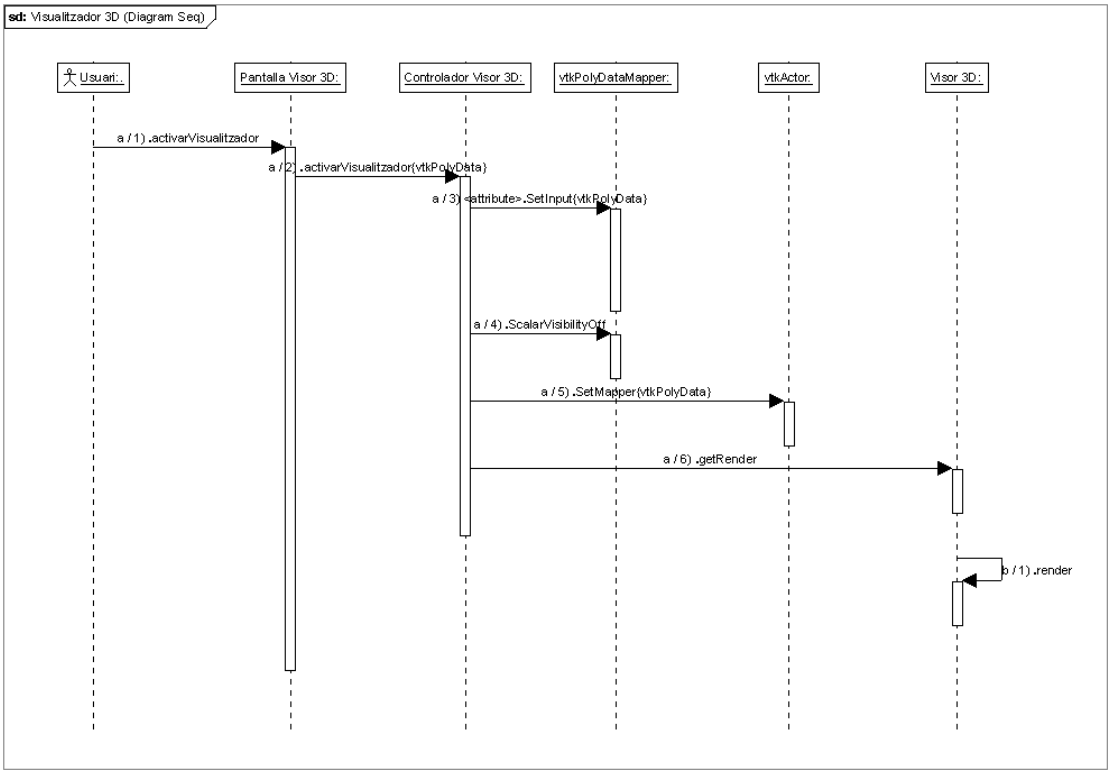


Figura 47. Diagrama de seqüència de la visualitzador de superfícies.

CAPÍTOL 8

Interacció

En aquest capítol es mostra la part de la implementació de la interacció. En els primers apartats tenim els anàlisis i requeriments amb una explicació detallada a partir de diagrames de casos d'ús i seqüència. Després apuntem com és el disseny empleat per realitzar la seva implementació.

8.1 Anàlisi de requeriments

8.2 Requeriments funcionals

Finalment tenim la visualització en 3D de l'esquelet del sistema vascular. Ara ens interessa poder mesurar el diàmetre dels elements vasals.

8.3 *Diagrames* de casos d'ús

Aquest és l'últim bloc de l'esquema de la figura 24 i és a on l'usuari interacciona amb el model per tal de poder realitzar els seus càlculs pertinents. Per a poder desenvolupar aquesta part, hem aprofitat les eines que disposa Starviewer per a poder mesurar models.

L'usuari ha d'utilitzar l'eina de mesura sobre el model en tres dimensions que hi ha al visor 3D.

Nom	***Visualització del sistema vascular en 3D***
Actor	Usuari
Precondició	Haver visualitzat una superfície 3D amb l'esquelet de centerlines.
Flux Principal	1. Prendre mesures amb l'eina Starviewer.
Flux alternatiu	
Postcondició	L'usuari pot fer els càlculs per a l'estenosi de caròtida a partir de les mesures que recull del model.
Comentaris	

Diagrama de cas d'ús

La figura 48 mostra el mateix diagrama de cas d'ús de la visualització del sistema vascular en 3D, però que a més a més l'usuari és capaç de poder prendre mesures del model en tres dimensions per a poder fer els càlculs d'estenosi de caròtida.

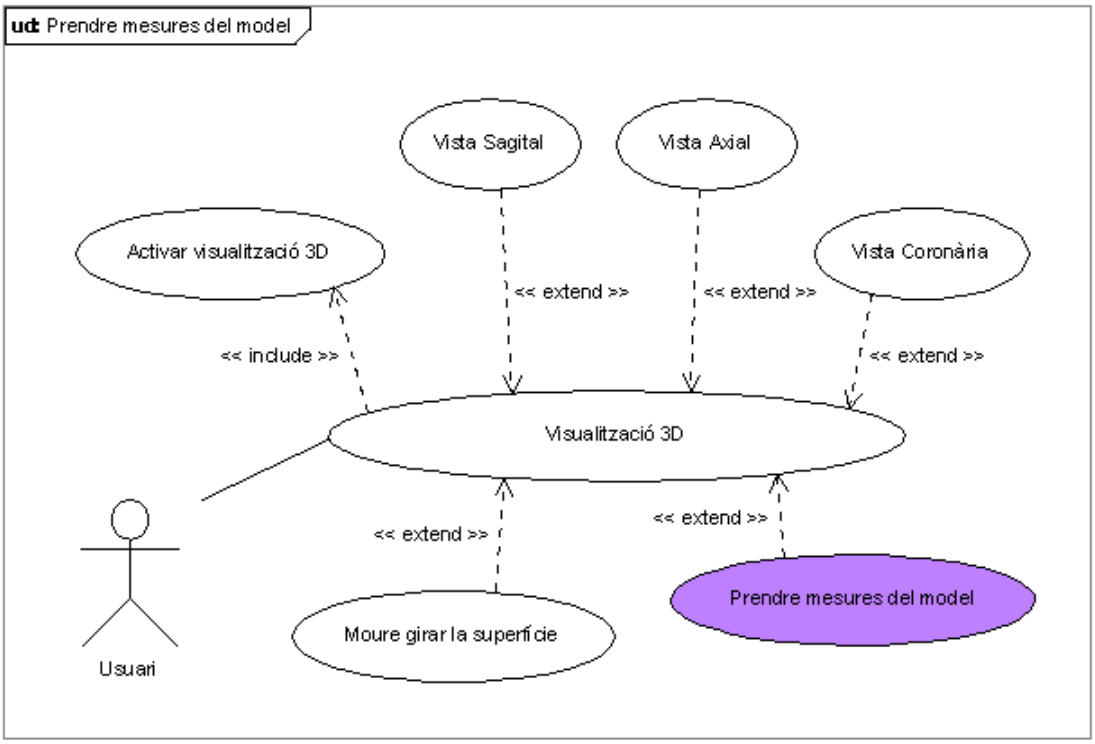


Figura 48. Diagrama de cas d'us per poder prendre mesures del model.

A causa que la integració amb les VMTK ha estat més complicada del que crèiem, ens ha ocupat molt més temps del previst i no n'hem dedicat tant al bloc de la interacció. Hem aprofitat les eines que ja ens donava la plataforma Starviewer per a la comunicació amb el model i creiem que amb això ja seria suficient per a que l'especialista pugui fer els càlculs pertinents.

Tots aquests mòduls formen part d'una mateixa extensió la qual s'anomena ***vascularmodeling***.

Per tal de poder fer més amè a la seva execució s'ha volgut procurar un disseny d'interfície a on hi hagi un visor per a cada bloc realitzat.

Un petit esquema del disseny gràfic que seria la nostra extensió seria:

<table>
<tr><td rowspan="2">Quadre d'eines que requereixen la interacció amb l'usuari</td><td>IMATGE ORIGINAL 2D</td><td>SUPERFÍCIE GENERADA PER MARCHING CUBES</td></tr>
<tr><td>SUPERFÍCIE ESQUELET AMB CENTERLINES</td><td>SUPERFICIE PREPARAT PER INTERACTUAR PER PRENDRE MESURES</td></tr>
</table>

CAPÍTOL 9

Resultats

Mostrarem les diferents etapes sobre un cas real amb les seus resultats corresponents.

Per tal de poder veure el funcionament de tots els blocs, farem un seguiment de totes les etapes amb un model de cas real.

Primerament tenim una angiografia de ressonància magnètica de substracció. A l'Starviewer obrim el DICOM i ens apareix el visor 2D amb la següent imatge:

Figura 49 Visio en 2D de l'ARM de substracció.

Llavors amb la eina d'informació de vòxel ens anem posant per sobre les zones a on hi ha vascular i s'intenta buscar el valor màxim d'intensitat.

En aquest cas concret veiem que el valor és 3000. Per tant escrivim el valor i executem el procés de segmentació. El sistema realitzarà el mètode de marching cubes amb un isovalor de 3000, i tindrem el resultat següent:

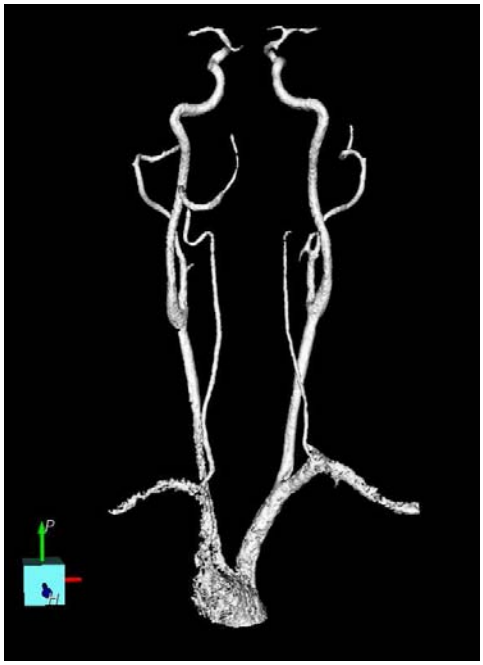


Figura 50 Segmentació del sistema vascular realitzat amb marching cubes.

Ara ja tenim el sistema vascular identificat del model. Ara s'haurà de marcar sobre la superfície els punts a on es voldrà que comenci la línia de centerline i els punts a on es voldrà que s'acabi l'eix central.

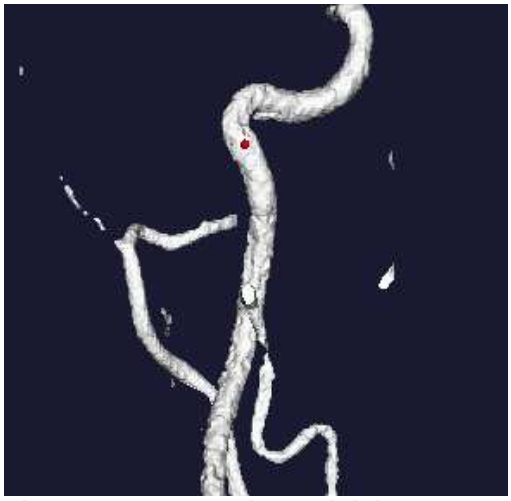


Figura 51. Marcar el punt inici

Si realitzem correctament aquestes pautes obtindrem la reconstrucció de l'esquelet amb el centerlines sobre l'estructura vascular.

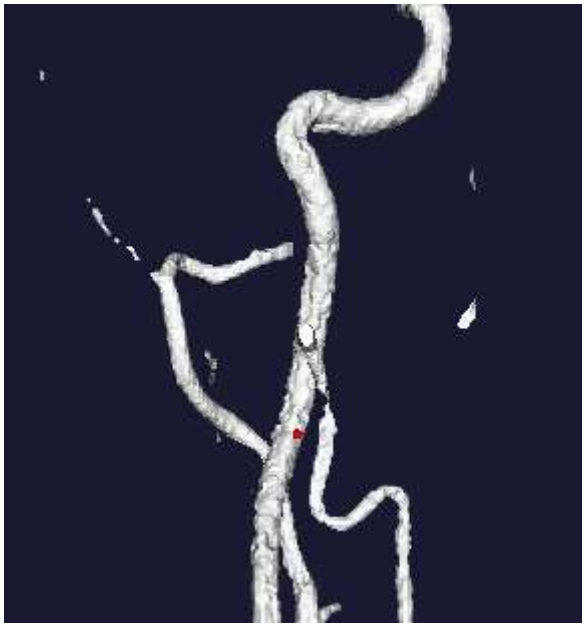


Figura 52. Marcar punt final

Gràcies al visor en tres dimensions i a les eines que disposa Starviewer, l'usuari ja pot fer les mesures per a realitzar els seus càlculs per diagnosticar l'estenosi de caròtida.

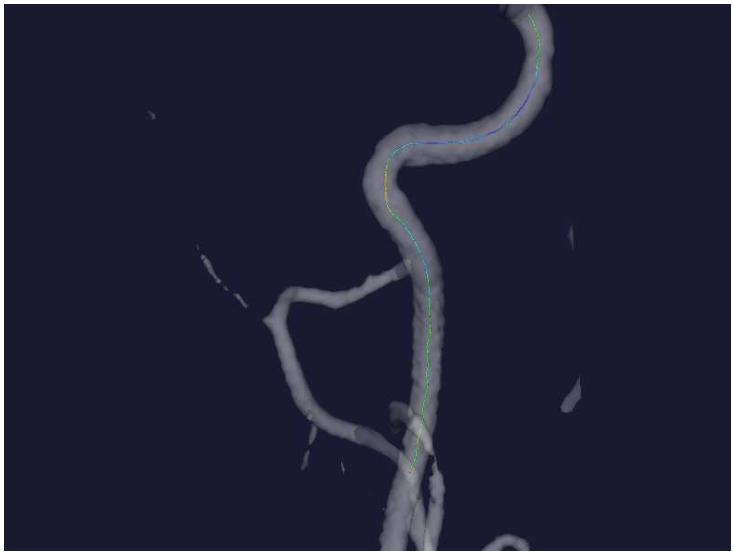


Figura 53. Reconstrucció del sistema vascular

Tenint en compte aquestes fórmules l'especialista podrà saber si s'ha d'intervenir quirúrgicament al pacient o no.

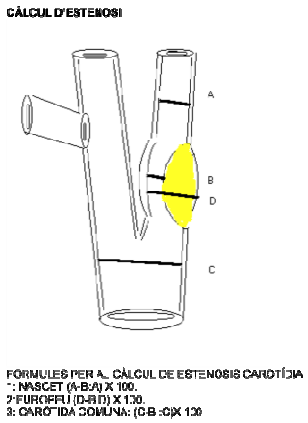


Figura 54 Fórmules per al diagnòstic d'estenosi de caròtida.

CAPÍTOL 10

Conclusions i Treball Futur

En aquest capítol es mostra la part de la implementació de la visualització. En els primers apartats tenim els anàlisis i requeriments amb una explicació detallada a partir de diagrames de casos d'ús i seqüència. Després apuntem com és el disseny empleat per realitzar la seva implementació.

L'objectiu principal d'aquest projecte era ampliar la plataforma Starviewer integrant els mòduls necessaris per donar suport al diagnòstic de l'estenosi de caròtida permetent interpretar de forma més fàcil les imatges de Angiografia per Ressonància Magnètica (ARM)

Els resultats són positius ja que s'han assolit tots els objectius proposats en començar el projecte

- **Ens hem familiaritzat amb la plataforma Starviewer i hem après el funcionament de les llibreries que s'usen en el seu desenvolupament**. Ens hem integrat en l'equip i hem adoptat la seva metodologia de treball per tal de realitzar el nostre projecte.

- **Hem estudiat les imatges de ARM per analitzar les seves característiques i identificar-ne la seva problemàtica**. També hem après l tipus d'operacions que cal realitzar per poder satisfer les necessitats dels especialistes. Hem après la funcionalitat de les imatges ARM i les diferències amb les altres tecnologies de diagnòstic. Hem vist que el cas de les imatges RMA les estructures vasculars no coincideixen amb cap altre valor d'intensitat de la imatge, una característica que ens ajuda a determinar com seran els blocs de segmentació i reconstrucció.

- Gràcies a la característica del valor d'intensitat de les ARM que ens assegura que les estructures òssies tenen valors diferents a les artèries, **hem pogut escollir d'entre els molts mètodes de segmentació que existeixen, un de reconstrucció de superfícies per tal de detectar i reconstruir automàticament el sistema vascular**. El mètode triat és el de marching cubes, el qual s'ha estudiat i s'ha dissenyat a partir de les llibreries noves VMTK amb el resultat d'obtenir una superfície amb el sistema vascular identificat.

- Hem après que una vegada s'han identificat les venes de les diferents imatges 2D per poder generar el sistema vascular cal aplicar un mètode de reconstrucció que sigui capaç de connectar les diferents parts. Nosaltres hem implementat la segmentació a partir de superfícies i ja disposem d'una reconstrucció 3D de tot el model vascular però tot i això ens cal identificar quin és l'esquelet d'aquest sistema. **Hem triat un sistema de reconstrucció amb l'objectiu d'obtenir l'esquelet del sistema vascular**. Com que hem estudiat les noves llibreries VMTK les quals han desenvolupat molt el mètode de reconstrucció amb centerlines, hem adoptat aquest com a nostre i l'hem implementat per a que ens faci el bloc de representació.

- **Hem implementat un mètode de visualització que ens ha permès visualitzat la informació en la pantalla.**

- **Finalment, hem implementat les eines bàsiques per poder prendre mesures sobre el nostre model per poder determinar el grau d'estenosi**.

Treball futur

Ens proposem validar tot l'entorn desenvolupat en l'entorn mèdic per tal de poder modificar i ajustar-lo a les necessitats reals. Som conscients de que aquesta fase ens implicarà moltes modificacions.

S'hauria de poder afegir un altre bloc addicional entre la segmentació i la reconstrucció a on l'estructura vascular passi per una següent segmentació com podria ser el level sets, per tal d'acondicionar més l'estructura per a que faci un càlcul més òptim de l'esquelet amb els centerlines.

Les VMTK ja disposen de scripts que fan aquesta tasca, però són força complexos ja que és un mètode de segmentació molt complert.

Conclusions personals

Com a apunt personal encara que el projecte hagi sigut força complex de realitzar, m'ha semblat molt interessant ja que el tema de la medicina sempre l'he trobat fascinant. El fet de pensar que gràcies a una cosa desenvolupada per nosaltres pugui ser de gran ajuda per a persones que realment ho necessiten en moments tan importants i durs com la incertesa de tenir alguna afecció patològica, és molt gratificant. Estic molt satisfeta que el nostre producte tingui una sortida útil per a professionals.

D'altra banda he après a utilitzar moltes llibreries que no havia tractat mai al llarg de la carrera i el fet d'haver-me implicat tant en les VMTK, unes de molt noves que quasi no tenen documentació, m'ha suposat quasi un repte personal per a poder lligar-ho tot per a aprofitar al màxim les eines les quals disposàvem. Sempre m'ha agradat el camp de la investigació de noves tecnologies i estudiar coses que encara no són gaire conegudes.

També m'ha servit per aprendre i quasi descobrir el llenguatge de programació Phyton. Té les seves particularitats i m'ha suposat una fase d'estudi addicional el qual no comptàvem.

Per acabar voldria afegir que estem contents amb el que hem realitzat però som conscients que es podria millorar molt per tal que es pugui adaptar més a les necessitats dels especialistes a la hora de realitzar el diagnòstic d'estenosi de caròtida.

Bibliografia

A continuació apuntem les fonts d'informació.

[VMTK] http://www.vmtk.org/

[VTK] Schroeder W., Martin K., Lorensen B. (2002-2004). *The Visualization Toolkit (3rd ed.).* Kitware

[VTK] VTK Mailing List. http://public.kitware.com/pipermail/vtkusers/

[ITK] Ibáñez L., SchroederW. (2003). *The ITK Software Guide.* Kitware

[ITK] ITK Mailing List. http://public.kitware.com/pipermail/insight-users/

[QT] QT Referente Documentation. http://doc.trolltech.com/4.0/index.html

[Wikipedia] http://es.wikipedia.org/wiki/Wikipedia:Portada