

---

## ÍNDIX

<b>1. INTRODUCCIO I OBJECTIUS .....</b>	<b>6</b>
1.1 ANTECEDENTS.....	7
1.2 OBJECTIUS.....	9
1.3 EINES UTILITZADES .....	10
1.4 PLA DE TREBALL .....	11
1.5 METODOLOGIA .....	12
1.6 PLANIFICACIÓ.....	14
<b>2. FASE D'ESTUDI PREVI .....</b>	<b>15</b>
2.1 DIAGNÒSTIC I TRACTAMENT DEL CÀNCER COLORRECTAL .....	16
2.1.1 Anatomia del colon i el recte .....	16
2.1.2 El càncer de colon.....	16
2.1.3 Mètodes de diagnòstic.....	17
2.2 EL PROCÉS DE VISUALITZACIÓ.....	21
2.3 ADQUISICIÓ DE DADES I FORMAT DICOM.....	23
2.4 MODELATGE.....	27
2.5 RECONSTRUCCIÓ.....	28
2.6 VISUALITZACIÓ .....	30
2.7 PROBLEMES A RESOLDRE .....	31
2.8 LA PLATAFORMA STARVIEWER .....	33
2.8.1 Eines utilitzades.....	33
C++ .....	33
Llibreries QT .....	33
QtDesigner .....	34
Llibreries ITK.....	34
Llibreries VTK.....	35
StarUML - The Open Source UML/MDA Platform.....	37
Doxygen .....	38
2.8.2 Starviewer.....	39
<b>3. FASE D'IMPLEMENTACIÓ.....</b>	<b>41</b>
3.1 DISSENY I IMPLEMENTACIÓ D'UN SISTEMA DE GENERACIÓ DE MODELS SINTÈTICS .....	43
3.1.1 Generació del model sintètic .....	45
Definició de la trajectòria.....	45
Construcció del budell virtual .....	46
Generació del volum .....	50
Guardar el Volum .....	51
3.1.2 Diagrames generació model sintètic .....	53
Definició del budell .....	54
Generar el budell a partir de les dades introduïdes .....	59
Guardar/Carregar el model sintètic .....	65
3.1.3 Classes del mòdul de generació model sintètic .....	69
3.1.4 Resultats i testeig .....	73

<b>3.2</b>	<b>AVALUAR LES TÈCNiques DE PROCESSAMENT D'IMATGE QUE CALEN PER DIAGNOSTICAR ALGUNES DE LES LESIONS DE BUDELL.....</b>	<b>74</b>
3.2.1	Tècniques de segmentació.....	74
3.2.2	Region Growing més Thresholding .....	75
3.2.3	Diagrames segmentació.....	77
3.2.4	Classes del mòdul de segmentació.....	78
3.2.5	Resultats i testeig .....	79
<b>3.3</b>	<b>DISSENYAR I IMPLEMENTAR UN SISTEMA D'EXPLORACIÓ DEL BUDELL .....</b>	<b>81</b>
3.3.1	Tècniques utilitzades en el sistema d'exploració.....	82
	Cercar el centre del budell .....	82
	Visualització adaptada al sistema d'exploració del budell .....	92
	Recórrer Trajectòria .....	95
3.3.2	Diagrames del mòdul sistema d'exploració del budell.....	98
	Cercar trajectòria.....	99
	Visualització del volum .....	108
	Viatjar pel centre del budell.....	110
	Funcionalitat visualització trajectòries .....	113
3.3.3	Classes del mòdul implementar un sistema d'exploració del budell .....	118
3.3.4	Resultats i testeig .....	123
	Resultats dels mètodes de cerca del centre .....	123
	Resultats del sistema visualització i exploració .....	126
<b>3.4</b>	<b>INTEGRACIÓ DE TOTS ELS MÒDULS IMPLEMENTATS EN LA PLATAFORMA STARVIEWER. ....</b>	<b>128</b>
<b>3.5</b>	<b>DIAGRAMA DE CLASSES FINAL.....</b>	<b>134</b>
<b>4.</b>	<b>CONCLUSIONS I TREBALL FUTUR.....</b>	<b>135</b>
<b>4.1</b>	<b>MILLORES SOBRE EL GENERADOR DE MODELS SINTÈTICS .....</b>	<b>135</b>
<b>4.2</b>	<b>MILLORES SOBRE ELS MÈTODES DE CERCA DEL CENTRE.....</b>	<b>135</b>
	4.2.1 Millorant la cerca del centre.....	135
	4.2.2 Altres mètodes per cercar el centre del budell.....	136
<b>4.3</b>	<b>MILLORES SOBRE EL SISTEMA D'EXPLORACIÓ DEL BUDELL .....</b>	<b>136</b>
<b>5.</b>	<b>VALORACIÓ PERSONAL.....</b>	<b>137</b>
<b>6.</b>	<b>BIBLIOGRAFIA .....</b>	<b>138</b>
<b>6.1</b>	<b>LLIBRES .....</b>	<b>139</b>
<b>6.2</b>	<b>ARTICLES.....</b>	<b>140</b>
<b>6.3</b>	<b>REFERÈNCIES ELECTRÒNIQUES.....</b>	<b>141</b>
<b>7.</b>	<b>ANEXES .....</b>	<b>142</b>
<b>7.1</b>	<b>CONTINGUT DE FITXER DE META DADES DE SPATIALS OBJECTS .....</b>	<b>142</b>
<b>8.</b>	<b>ÍNDIX D'IMATGES .....</b>	<b>143</b>

---

*A la meva tutora per la paciència  
a la meva família per la comprensió  
i als amics de veritat perquè ho són*

*A tots ells moltes gràcies*

## **1. INTRODUCCIO I OBJECTIUS**

## 1.1 Antecedents

El diagnòstic mitjançant la imatge mèdica, s'ha convertit en una eina fonamental en la pràctica clínica diària ja que permet, entre 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. El projecte que es presenta forma part del desenvolupament de la plataforma informàtica de visualització i tractament de dades mèdiques, anomenada Starviewer, que desenvolupen conjuntament el laboratori de Gràfics i Imatge (GiLab) de la Universitat de Girona i l' Institut de Diagnòstic per la Imatge (IDI) de l'Hospital Josep Trueta de Girona. En particular, en aquest projecte ens centrarem en el diagnòstic del càncer de colon.

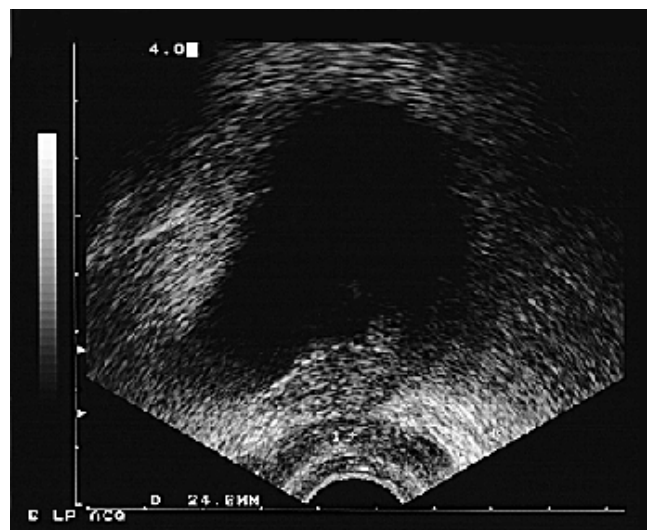


fig 1.1-1: Ecografia

Actualment, el diagnòstic de càncer colorectal es fa a partir de l'ecogràfica transrectal. Aquesta tècnica presenta diferents limitacions, entre elles el fet que és operador-dependent, la rigidesa del transductor dificulta la valoració de lesions, etc. Aquestes limitacions es poden resoldre amb l'ús complementari de la ressonància magnètica (RM) pèlvica. Aquest tipus d'exploració resol la majoria d'inconvenients que presenta l'ecografia transrectal.



fig 1.1-2: Ressonància magnètica

Tot i que la informació que proporcionen les imatges de RM és molt precisa, la valoració d'una resposta tumoral, al tractament determinat, es converteix en un judici subjectiu del radiòleg. Ja que no es disposa d'eines per avaluar el volum, la forma, la localització, entre altres característiques de la lesió que són les que indiquen l'estat de la malaltia a l'especialista.

La solució a aquests problemes passaria per l'automatització de les diferents tasques que ha de realitzar l'especialista en el moment de diagnosticar. El punt clau seria la reconstrucció del budell del pacient, a partir de les dades de RM. Per altre banda s'hauria de dotar de mètodes que permetessin calcular el volum, millorar la localització i l'extensió de la lesió d'una forma més acurada. El que permetria identificar l'estat de lesió d'una forma més objectiva i com a conseqüència determinar el tractament adient pel pacient.

## 1.2 Objectius

L'objectiu principal d'aquest projecte és implementar i integrar en la plataforma *Starviewer* les tècniques de visualització i processament de dades necessàries per donar suport als especialistes en el diagnòstic de les lesions del colon. Donada la dificultat en el processament de les dades reals del budell ens proposem:

### **DISSENYAR I IMPLEMENTAR UN SISTEMA PER CREAR MODELS SINTÈTICS DEL BUDELL.**

S'estudiarà el format de dades reals i es crearà un mòdul que, a partir de les restriccions fixades per l'usuari, permeti crear un model sintètic amb les mateixes característiques que el model real de budell. Entre les restriccions fixades per l'usuari tindrem: la trajectòria del budell, l'amplada, posició de les lesions, etc.

### **ESTUDIAR I AVALUAR LES TÈCNiques DE PROCESSAMENT D'IMATGE QUE CALEN PER SEGMENTAR LESIONS DE BUDELL.**

El punt clau en el diagnòstic del càncer de colon està en el càlcul del volum i en la localització de la lesió. Estudiarem les tècniques de segmentació que ens permetin localitzar i calcular aquest volum.

### **DISSENYAR I IMPLEMENTAR UN SISTEMA D'EXPLORACIÓ DEL BUDELL.**

Un cop tractades les dades haurem de simular un viatge virtual a través de la trajectòria descrita pel budell. En aquest cas, haurem d'implementar alguna eina que ens permeti calcular la trajectòria i navegar-hi a través visitant virtualment l'interior del tub. Ens interessa simular una colonoscopia virtual.

### **INTEGRACIÓ DE TOTS ELS MÒDULS IMPLEMENTATS EN LA PLATAFORMA STARVIEWER**

Tots els mòduls que s'implementin s'integraran en la plataforma *Starviewer* dotant a la plataforma de les noves funcionalitats desenvolupades en el projecte.

## 1.3 Eines utilitzades

El fet de partir de la plataforma *Starviewer* ens obliga a treballar amb un conjunt d'eines predeterminades. En particular, les eines utilitzades per la implementació són:

- ***El llenguatge C++***

C++ és un llenguatge de programació d'alt nivell multiparadigma amb el que es pot escriure qualsevol tipus de programa. Una de les avantatges més significatives del C++ és que suporta diferents estils de programació, com son la programació estructurada o la programació orientada a objectes.

- ***Les llibreries d'interfície gràfica Qt***

Les Qt son unes llibreries multiplataforma, molt completes i robustes, per desenvolupar interfícies gràfiques d'usuari. Utilitza el llenguatge de programació C++ de forma nativa apart de molts altres. I proporciona una multitud d'eines per el maneig d'estructures de dades tradicionals, tractament de fitxers i accessos a dades, entre d'altres.

- ***QtDesigner***

Juntament amb les llibreries Qt utilitzarem el QtDesigner. Una plataforma de programació visual de interfícies que ens en facilitarà la generació amb les Qt.

- ***Les llibreries de tractament d'imatges i volums ITK's***

ITK és un software lliure orientat a objectes desenvolupat per donar suport al processament, segmentació i enregistrament d'imatges. Les llibreries son multiplataforma i estan desenvolupades en C++. A més compten amb tot una comunitat d'usuaris i dissenyadors que per mitjà "foro-log" poden donar suport i informar sobre errors, noves classes i tests.

- ***Les llibreries de visualització de imatges VTK's.***

Les VTK (Visualization Toolkit) són unes llibreries *de codi lliure*, enfocades a gràfics 3D, processat d'imatges i visualització. Formen un sistema per processar imatges i visualització 3D disponible per una gran quantitat de plataformes. És un paquet clarament orientat a les tècniques de POO compost per més de 700 classes C++.



## 1.4 Pla de treball

Per assolir tots els objectius proposats pel projecte s'ha seguit el següent pla de treball dividit en dues fases:

- **Estudi previ**

L'objectiu d'aquesta fase és el de descriure la feina preliminar que hem hagut de fer abans de començar el disseny i el desenvolupament d'aquest projecte.

Es realitzarà un estudi de l'entorn mèdic per comprendre la patologia del càncer de colon així com les tècniques i operacions relacionades amb el diagnòstic per la imatge.

S'estudiarà tant l'Starviewer com les llibreries utilitzades en el seu desenvolupament, per tenir els coneixements bàsics per implementar mòduls adaptables sobre la plataforma. A més de estudia i conèixer algunes tècniques de visualització i renderitzat 3D implementades en l'aplicació.

- **Fase d'implementació.**

Una vegada assolits els conceptes teòrics es realitzarà el disseny, implementació i validació dels mètodes a desenvolupar. En aquesta fase es farà l'anàlisi de requeriments i el disseny dels diferents mòduls que formen l'aplicació. S'explicarà la part que fa referència al disseny del sistema informàtic. Mitjançant diagrames de classes, diagrames de cas d'ús i diagrames de seqüència definirem l'estructura del projecte.

Al final del document parlarem de les conclusions extretes una vegada finalitzada la implementació de les funcionalitats i proposarem millores que es poden fer per ampliar i millorar l'aplicació. Moltes d'aquestes millores, sorgeixen a l'hora de provar les funcionalitats implementades i veure que l'aplicació podria millorar amb ampliacions futures.

## 1.5 Metodologia

La programació d'aquest projecte és orientada a objectes (OO), per tant l'anàlisi i disseny de l'aplicació està basat en aquesta tècnica. Per fer l'anàlisi i els diagrames d'aquesta aplicació utilitzem la notació/nomenclatura *Unified Modeling Language* (UML). Tot i utilitzar la notació UML i fer-ho des d'una perspectiva orientada a objectes, la metodologia utilitzada ha estat *l'Extreme Programming* (XP) i no la *Rational Unified Process* (RUP). Al llarg de la carrera en diverses assignatures ens han insistit que ens acostumem a treballar amb la metodologia RUP, però en aquest cas i degut a la complexitat de dissenyar l'aplicació amb antelació a la implementació, hem decidit utilitzar una metodologia com l'XP. Desenvolupar el projecte utilitzant aquesta metodologia vol dir dir a terme un procés de disseny, implementació i *testing* iteratiu i simultani. Així doncs els diagrames de l'anàlisi i el disseny de l'aplicació s'han anat fent i modificant a mida que anava evolucionant el codi. Tot seguit comentem amb una mica més de profunditat la metodologia *d'Extreme Programming*.

---

### Extreme Programming

---

Moltes metodologies defineixen una sèrie de regles de construcció a partir d'uns requeriments ben definits a l'inici del projecte. Els canvis d'aquests requeriments poden variar al llarg del desenvolupament del projecte, i els costos d'adaptació als nous canvis poden disparar el cost inicialment previst. En canvi *l'extreme programming* (XP) consta d'una sèrie de regles senzilles a seguir per obtenir un software fàcilment adaptable a entorns evolutius i millora la reutilització.

L'Extreme Programming 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.

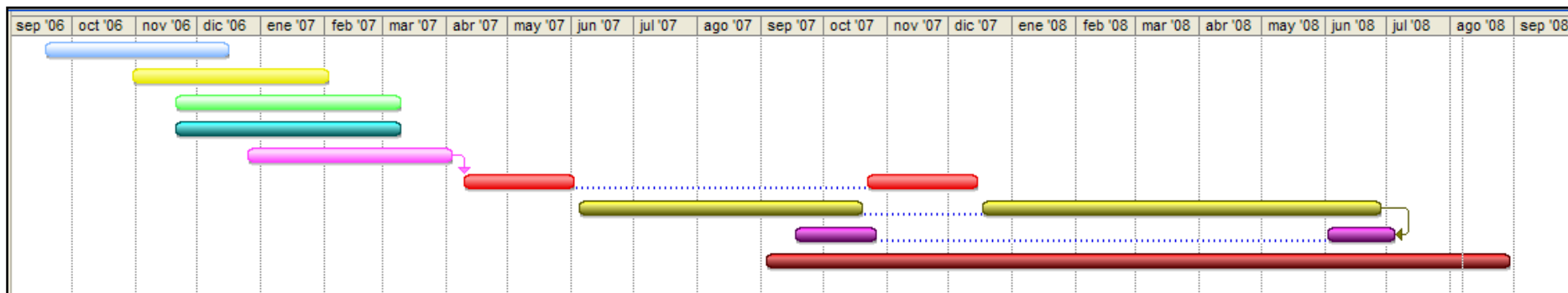
En la implementació d'aquest projecte s'intentarà seguir totes les premisses esmentades, no essent possible el compliment d'alguna d'elles, com seria, la programació per parelles, per un motiu obvi, ja que el projecte s'ha fet de forma individual. També, val a dir, que la figura del client serà substituïda 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 dades sintètics, validant que els resultats obtinguts es corresponguin a les dades del model, i que no se n'alterin el contingut al fer el tractament. En un futur es farà el testeig sobre models reals.


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

## 1.6 Planificació

En l'esquema de planificació que es mostra a continuació detalla la temporalització seguida de les tasques i les reunions amb els usuaris que hauran d'interactuar amb l'aplicació desenvolupada.



### Llegenda

- |   |                                 |   |                      |
|---|---------------------------------|---|----------------------|
|  | Fase d'estudi mèdic             |  | Fase de Disseny      |
|  | Fase d'estudi de llibreries QT  |  | Fase d'Implementació |
|  | Fase d'estudi de llibreries VTK |  | Fase de Tasteig      |
|  | Fase d'estudi de llibreries ITK |  | Documentació         |
|  | Fase d'estudi Starviewer        |   |                      |

## 2. FASE D'ESTUDI PREVI

L'estudi previ inclou una primera fase en la que s'estudiarà el problema a tractar des d'un punt de vista mèdic i les diferents tècniques que poden aplicar-se per donar suport als especialistes i una segona fase que presentarà l'entorn de treball des del punt de vista informàtic. Veurem la plataforma *Starviewer* i les diferents eines que s'usaran per programar.

## 2.1 Diagnòstic i tractament del càncer colorrectal

Per descriure el diagnòstic i tractament del càncer de colon començarem amb una breu descripció de l'anatomia del colon i el recte ja que és la zona en la que es situa la patologia. A continuació presentarem el càncer de colon i les diferents exploracions que poden realitzar-se per adquirir informació dels pacients afectats.

### 2.1.1 Anatomia del colon i el recte

El colon i el recte constitueixen l'intestí gruixut, que té un paper important en el processament dels excrements.



fig. 2.1-1 Anatomia del colon

Els primers 150 a 180 centímetres de l'intestí gruixut corresponen al colon i els últims 15 al recte que acaba a l'anus.

El colon es divideix en quatre seccions:

- El colon ascendent es la porció del colon que connecta amb el final de l'intestí prim des del cec, situat a la dreta de l'abdomen, adjacent a l'apèndix.
- El colon transvers que passa per la part superior de l'abdomen.
- El colon descendent a la part esquerra de l'abdomen.
- Finalment el colon sigmoide, situat a la part final, que connecta amb el recte.

### 2.1.2 El càncer de colon

El càncer colorectal és una malaltia que provoca un canvi en les cèl·lules normals de la paret del colon o recte creixent de forma descontrolada i sense morir. Aquests canvis habitualment tarden anys a desenvolupar-se però, com en alguns casos de malalties hereditàries, els canvis poden produir-se en mesos o anys. En un principi el creixement de les cèl·lules es mostra com un pòlip benigne (no cancerós), que amb el temps, pot esdevenir un tumor cancerigen. Si els pòlips no

són tractats o extirpats, poden progressar i desenvolupar càncers potencialment mortals. Els tumors es poden localitzar tant a la altura del colon com al recte.

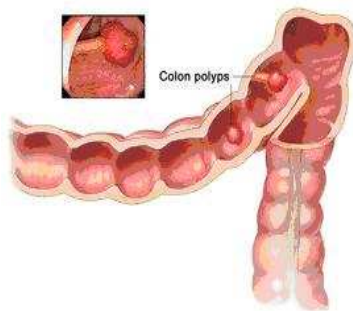


fig. 2.1-3 Pòlips en el colon

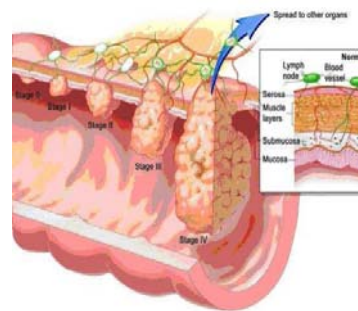


fig. 2.1-2 Fases dels pòlips

El colon és la part del sistema digestiu on s'emmagatzema els residus. El recte és l'extrem del colon adjacent a l'anus. Junts, formen un tub llarg, muscular anomenat l'intestí gros o gruixut.

El tumor del colon i recte sorgeixen de la paret interior de l'intestí gros. Els tumors benignes de l'intestí gros s'anomenen pòlips. Els pòlips benignes no envaeixen teixit pròxim o extensió a unes altres parts del cos. Aquests es poden treure fàcilment i de forma definitiva mitjançant una colonoscòpia, la qual redueix en risc de càncer. Si els pòlips benignes no s'extirpen de l'intestí, poden esdevenir malignes (cancerós) gradualment. Es considera que gran part dels càncers d'intestí gros s'han desenvolupat a partir de pòlips. El càncer de colon i recte pot envair i fer malbé teixits adjacents així com òrgans. Les cèl·lules de càncer també es poden distanciar i estendre a unes altres parts del cos (com fetge i pulmó) on formen els tumors nous. L'extensió de càncer de colon s'anomena metàstasi del càncer de colon. Una vegada que la metàstasi ha ocorregut en el càncer colorectal, una cura completa del càncer és improbable.

El càncer de colon és la tercera causa de mort tant en dones com en homes. És comú anomenar-lo càncer d'intestí. El risc de càncer de còlon augmenta amb l'edat, és estrany veure'l abans dels 40 anys, excepte en les formes hereditàries de la malaltia, que de fet són menys comuns, que la forma esporàdica.

Afortunadament, avui en dia podem comptar amb diferents mètodes per detectar de manera precoç el càncer de colon i lesions premalignes, fent possible la cura definitiva. Entre les diferents proves s'inclou l'examen d'excrements per detectar sang oculta, la colonoscòpia, el tacte rectal, etc.

### 2.1.3 Mètodes de diagnòstic

En el moment de fer el diagnòstic existeixen diferents tècniques de captació d'informació del pacient. El fet d'usar una tècnica o una altre depèn de l'estat del pacient i de les funcionalitats de les que disposi el centre de diagnòstic en el que es fan les exploracions.

### COLONOSCÒPIA

És un examen intern del colon, utilitzant un aparell anomenat colonoscopi, que consisteix en una petita càmera adherida a un tub flexible amb el qual s'explora el colon en tota la seva extensió. Com es mostra en la figura 2.1-4.

El colonoscopi es connecta a una càmera de vídeo i a un monitor perquè el metge pugui examinar el colon. També es pot extirpar els pòlips o recollir material per una biòpsia. Aquesta tècnica és, ara com ara, el millor mètode de diagnòstic.

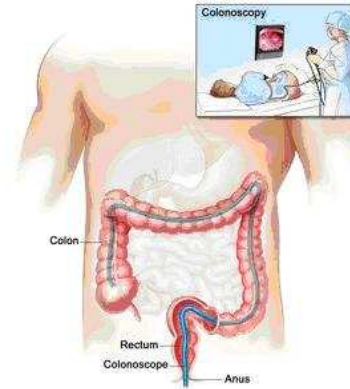


fig. 2.1-4 Colonoscòpia

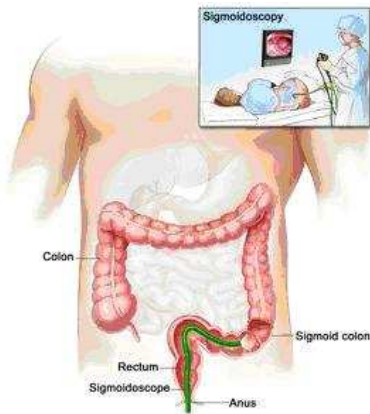


fig. 2.1-5 Sigmoidoscòpia flexible

### SIGMOIDOSCÒPIA FLEXIBLE

Aquesta tècnica consisteix en introduir un tub prim amb llum a través del recta fins uns 60 centímetres en busca de pòlips, àrees anormals i càncer. La sigmoidoscòpia flexible permet diagnosticar més de la meitat de tots els càncers de colon i biopsiant els teixits en casos que es necessiti.

### ENEMA OPAC

Es una tècnica radiològica que permet veure la silueta del colon després d'haver introduït un contrast bàric per el recte, fins omplir la meitat del colon. El pacient es mourà per distribuir el bari per tot el còlon. Després s'introdueix aire per expandir el còlon. Amb això s'aconsegueix visualitzar bé el còlon i al fer una radiografia es veuran defectes als plects o contorns irregulars. L'ènema de bari és una alternativa útil, tot i que menys eficaç, si no es disposa de colonoscòpia o si el pacient no pot ser sotmès a sedació.

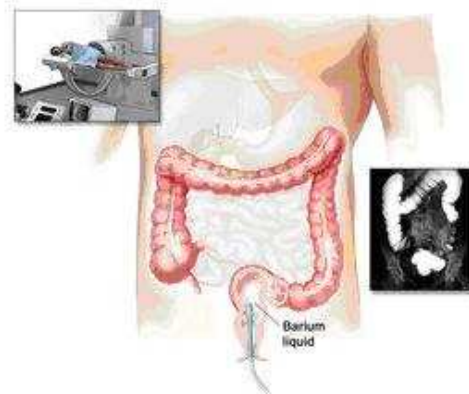


fig. 2.1-6 Enema Opac



### TACTE RECTAL

El tacte rectal consisteix en un examen del recte. El doctor introdueix un dit, protegit amb un guant lubricat a través del recte per trobar masses o àrees anormals.

Aquesta tècnica s'utilitza de manera complementària ja que com a mètode de diagnòstic no és gaire útil ja que la majoria dels tumors són més enllà dels 11cm.

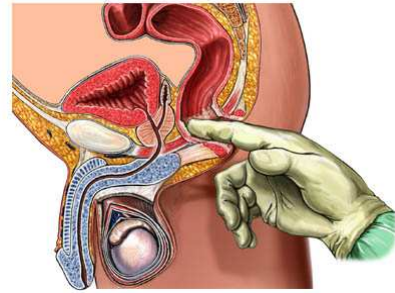


fig. 2.1-7 Tacte rectal

### ECOGRAFIA TRANSRECTAL

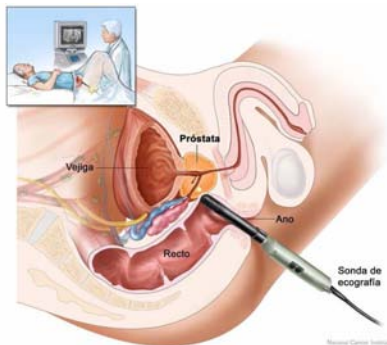


fig. 2.1-8 Ecografia tansrectal

Procediment en el qual s'insereix al recte una sonda que emet ones sonores d'alta energia. Les ones sonores reboten en teixits o òrgans interns i produeixen ecos. Els ecos formen una imatge del teixit del cos que es diu ecografia. L'ecografia transrectal s'usa per analitzar anomalies al recte i en les estructures properes, fins i tot la pròstata. També s'anomena ETR, ecografia endorrectal i EER.

Un fet rellevant de l'ecografia és la possibilitat d'efectuar estudis en temps real, permetent efectuar valoracions comparatives d'alta significació per demostrar la presència d'anomalies o canvis focals que poden ser els indicis de lesions incipients.

### COLONOSCÒPIA VIRTUAL

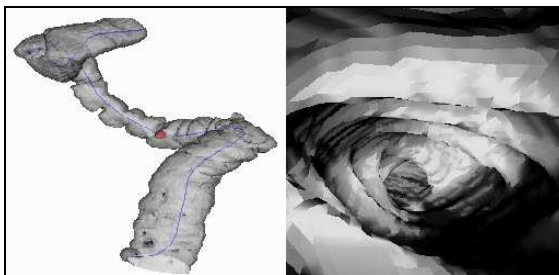


fig. 2.1-9 Colonoscòpia virtual

La colonoscòpia és el mètode de diagnòstic amb el que ens basarem en el nostre projecte.

A partir de les dades obtingudes amb el TC pèlvic, reconstruirem tridimensionalment el budell i en farem un recorregut virtual.

La colonoscòpia virtual és un tipus avançat d'examen per radiografia que utilitza tomografia computaritzada (TC o "TAC") per obtenir una vista interior del colon que normalment només es pot veure mitjançant endoscopi introduït per el recte. Es tracta d'un examen mínimament invasiu. Proporciona imatges tridimensionals que poden mostrar molts pòlips i altres lesions amb la mateixa claredat que amb la colonoscòpia òptica, però a més es podem visualitzar la paret externa de l'intestí a més de la resta d'òrgans abdominals. Aquesta tècnica permet localitzar amb exactitud la posició de les lesions observades per una posterior actuació endoscòpia o quirúrgica.

Les dades que rebem i que posteriorment tractarem son obtingudes mitjançant un TAC abdominal, seguidament expliquem el procés de captació , transformació i visualització de les dades ja que aquesta tècnica és la que nosaltres considerarem.

Volem remarcar que el nostre objectiu no és fer una colonoscòpia virtual sinó dissenyar una tècnica que s'hi approximi.

## 2.2 El procés de visualització

Des de que el pacient arriba a un centre de diagnòstic fins que les imatges surten per la pantalla de l'ordinador, passen una sèrie de processos que formen el que es coneix com el pipeline de visualització. A continuació veurem cada una de les etapes d'aquest pipeline:

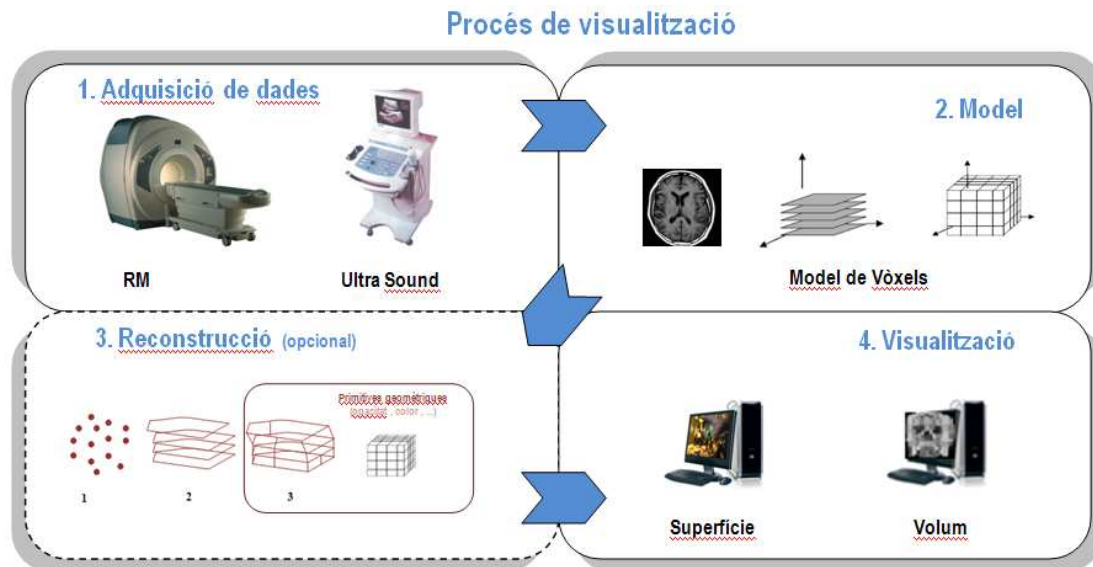


fig. 2.2-1 Pipeline del procés de visualització

### ▪ **Adquisició de dades**

La primera etapa del procés de visualització és la **captació de dades**. L'objectiu principal és l'obtenció del conjunt de dades que cal interpretar. Aquestes dades s'obtenen mitjançant simulacions o bé a través de dispositius especialitzats de captació.

El procés de captació, secciona el cos en plans de tall paral·lels on es representa la secció del volum estudiat, en el cas de les imatges mèdiques, el cos del pacient.

Una de les característiques que tenen en comú el tipus de dades amb els quals treballen, és que sempre segueixen una distribució espacial regular i que sempre ens venen distribuïdes sobre plans. Per aquest motiu, les imatges es guarden generalment en el format DICOM.

### ▪ **Modelatge**

Per comprendre i tractar les dades del pacient, recollides amb els diferents dispositius al computador, cal definir un esquema de representació que en faciliti l'accés.

El model de representació és la forma d'emmagatzemar i organitzar la informació obtinguda del pacient. En el nostre cas ens centrarem en els **model de vòxels** com a model de representació.

- **Reconstrucció**

L'etapa de **reconstrucció** té lloc un cop obtingudes les dades s'han de llegir i tractar per obtenir-ne una representació que ens permeti treballar amb facilitat sobre el volum, traduint les dades recollides al model de representació escollit.

Aquesta fase és opcional i s'aplicarà en funció dels interessos de l'usuari. Un dels mètodes de reconstrucció més usats és el *marching cubes* a partir del qual es poden reconstruir estructures i representar-les com a malles de trinagles.

- **Visualització**

La darrera etapa és la de **visualització**. En aquesta fase s'aplica alguna tècnica de visualització sobre el model de dades (o el model simplificat) per tal d'obtenir la imatge final que representa les dades.

El mètode de visualització variarà en funció del model de representació. La tècnica de visualització haurà de determinar quins atributs gràfics s'assignen als diferents valors de propietat representats en el model per tal que es pugui obtenir una representació gràfica del mateix. Evidentment el tipus d'atributs i les tècniques a aplicar variaran en funció del tipus d'informació que volem visualitzar i del tipus d'informació que tinguem representada en el model.

A continuació presentarem la descripció de cadascuna d'aquestes etapes amb més detall.

## 2.3 Adquisició de dades i format DICOM

L'objectiu principal de la captació de dades és l'obtenció del conjunt de dades que cal interpretar. Aquestes dades es poden obtenir mitjançant simulacions o bé a través de dispositius especialitzats de captació.

En el cas de les aplicacions mèdiques, aquesta etapa es realitza mitjançant un dispositiu de captació, com per exemple les CT (*Computerized Tomography*), que permet mesurar un o més paràmetres determinats en diferents punts del cos del pacient. Aquests paràmetres es representen en forma d'imatges, de manera que cada nivell d'intensitat de la imatge representa un valor. La fiabilitat de les dades obtingudes dependrà de la precisió de l'aparell utilitzat en la captació de les mateixes.

Generalment, aquestes imatges es guarden en el format DICOM. Una de les característiques que tenen en comú el tipus de dades amb els quals treballen, és que sempre segueixen una distribució espacial regular i que sempre ens venen distribuïdes sobre plans.

Aquest procés de captació secciona el cos en "llesques" o, per entendre-ho millor, en imatges planes 2D, definides per plans paral·lels a algun dels plans de projecció (frontal, coronals o axial). Aquest procés és representat a la figura 2-3.1 on es veuen les llesques resultants d'un TAC cerebral.

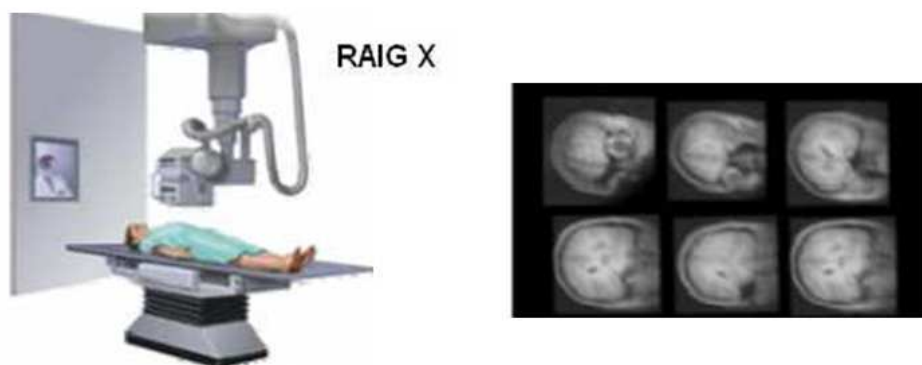


fig. 2.3-1 Obtenció de llesques del tac

Cal esmentar que els conjunts de dades proporcionats pels diferents dispositius de captació poden tenir resolucions diferents. Cada volum de dades tindrà la seva pròpia resolució, depenent de la prova que s'hagi fet i la màquina que s'hagi fet servir.

D'altra banda s'ha de tenir en compte que l'extensió espacial dels conjunts de dades també pot ser diferent. Podem trobar-nos per exemple, que en un model hi ha una exploració de tot el cap i en l'altre només una part. Tota aquesta informació ens ve donada en la capçalera del fitxer DICOM i serà la que determinarà les característiques del model de vòxels.

Per entendre, amb més detall, com s'obtenen les imatges a partir de dispositius de captació com les CT, mostrem les imatges de la figura 2.3-2. El pacient es col·loca dins de l'aparell, el qual té un tub que llança raigs X. Aquest tub va donant voltes al voltant del pacient per captar dades des de tots els angles. A part d'aquest moviment, l'aparell de captació es desplaça longitudinalment a través de la zona d'estudi del pacient.

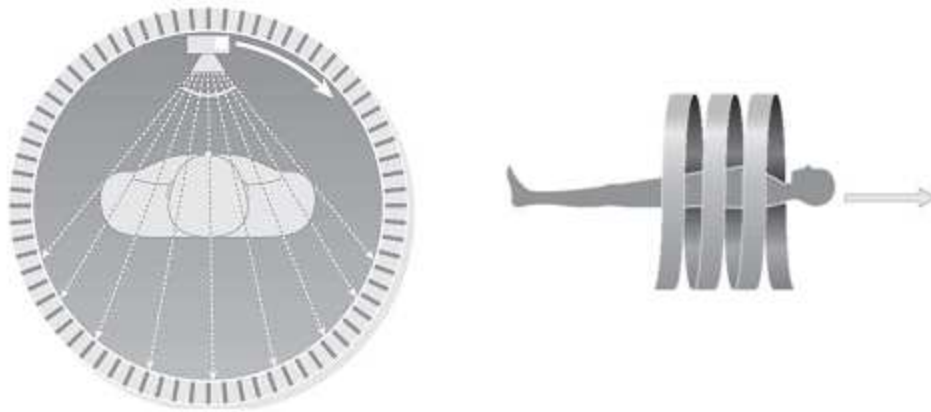


fig. 2.3-2 Procés de captació de dades per CT.

A la figura 2.3-3, veiem un exemple de llesca obtinguda a partir d'una CT. Els diferents valors de gris representen determinats teixits del cos.

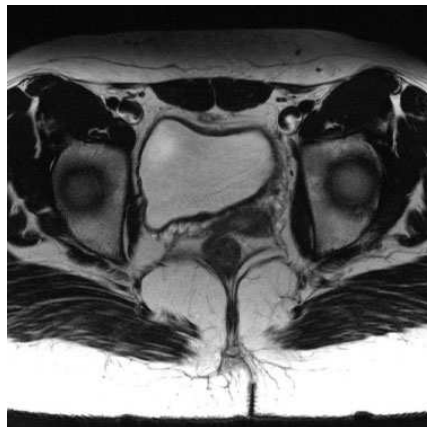


fig. 2.3-3 Tac pèlvic

---

## DICOM

El **DICOM** (Digital Imaging and Communications in Medicine) és una estàndard per manipular, emmagatzemar i transmetre informació d'imatges mèdiques. La norma inclou una definició del format del fitxer DICOM i dels protocols de comunicació, basats en TCP/IP, que permeten intercanviar informació entre dos entitats que tinguin la capacitat d'enviar i rebre informació en aquest format.

El DICOM va ser desenvolupat per permetre la integració i comunicació d'escàners, aparells radiogràfics, impressores, servidors, estacions de treball i múltiple *hardware* de xarxa de proveïdors dins d'un sistema d'emmagatzematge i comunicació d'imatges.

Els mecanismes diferents vénen amb declaracions de conformitat DICOM (*conformance statements*) que estableixen clarament les classes de DICOM a què donen suport. DICOM ha estat àmpliament adoptat per hospitals i s'està fent ús en aplicacions més petites com les consultes de dentistes i metges.

Aquest format és el que més s'usa actualment.

## ESTRUCTURA DEL FITXER DICOM

Els fitxers DICOM corresponen a la part 10 de l'estàndard DICOM. Descriu com emmagatzemar la informació de les imatges mèdiques en un mitjà extraïble. Generalment és obligatori incloure també les *metadades* de la imatge. DICOM restringeix els noms dels fitxers a 8 caràcters. Del nom del fitxer no s'ha d'extreure cap informació. Els protocols usats pels aparells de captació solen dividir les dades captades en dos fitxers, el fitxer de capçalera i el fitxer de dades, aquest format és part del protocol DICOM.

- **Fitxer capçalera (.mhd)**

Aquest fitxer indica la modalitat de la imatge i ens serveix per saber com s'ha de llegir i interpretar la informació del fitxer de dades. Un cop haguem obtingut els valors correctament, sabrem les característiques i el mètode a aplicar per poder veure la informació que conté.

El fitxer de la figura 2.3-5 ens informa que el fitxer de dades és *.dt-helix.raw* les dades que trobarem són del tipus *float*, estan guardades en *big endian* i les imatges són de 38x39px i en tenim 40.

```
NRRD0001
type: float
dimension: 4
sizes: 7 38 39 40
axis mins: NaN -2 -2 -2
axis maxs: NaN 2 2 2
data file: ./dt-helix.raw
endian: big
encoding: raw
```

fig. 2.3-4 Exemple fitxer capçalera .mhd

- **Fitxer de dades (.raw)**

Fitxer que conté les dades. Mitjançant la capçalera sabem cada dada que llegim a quina imatge es correspon i a quin punt de l'espai (x,y) pertany.

En l'exemple de la figura 2.3-6 mostrem la informació del fitxer .raw i és el que conté les dades de l'estudi realitzat. És a dir, el conjunt d'imatges del pacient. Hem de tenir en compte que les dades que es representen en aquest fitxer es poden representar com imatges. Només cal assignar un valor d'intensitat al valor de la mostra.

```
00010000000001000101011
01001110100101010001001
01001001001010010100001
01011100001110000110101
101001010100000.....
```

fig. 2.3-5 Exemple de fitxer de dades d'un DICOM



## 2.4 Modelatge

En aquesta etapa, es defineix el model de representació que s'usarà per representar les dades obtingudes en l'etapa anterior. En el nostre cas ens centrarem en els models volumètrics.

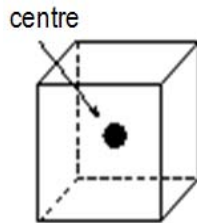


fig. 2.4-1 Representació de vòxel

Si volem tractar les dades del pacient al ordinador cal definir un esquema de representació que ens permeti accedir fàcilment a elles. El model més usat en el camp de la medicina és el model de vòxels proposat per *Kaufman*.

Aquest model subdivideix l'espai en un conjunt de cubs o paral·lelepípedes de les mateixes dimensions, seguint una malla regular. Cadascun d'aquests cubs, o paral·lelepípedes, rep el nom de vòxel.

Seguidament descrivim el tractament que es fa de les dades obtingudes del pacient. En les figures següents veiem com en cada vòxel es guarda una descripció gràfica de les dades del volum.

Partim de les dades obtingudes a partir dels TAC o TC guardades en format DICOM. Per tant tenim una representació de la part del pacient mitjançant plans de tall que contenen informació referent a les parts incloses en la extensió d'estudi. Els diferents plans s'apilen un sobre l'altre reconstruint el volum en 3D.

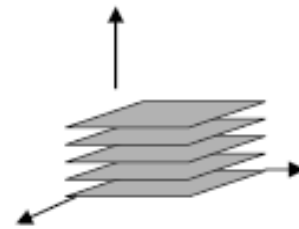


fig. 2.4-2 Plans de representació del model

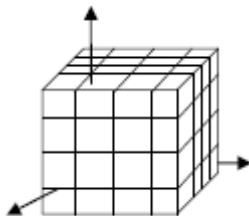


fig. 2.4-3 Representació volumètrica de les dades

D'aquesta manera obtenim una malla regular que representa la informació de cada pla de tall volumètricament. Aquest model es coneix com a model de vòxels.

## 2.5 Reconstrucció

Una vegada definit el model de vòxels, la següent fase del procés de visualització és la selecció de les primitives geomètriques que s'utilitzaran per generar la imatge final.

Els algoritmes de visualització variaran en funció de si es treballa sobre el model de vòxels inicial o si es treballa sobre un model simplificat, com són les primitives geomètriques, construïnt a partir de les dades inicials representades en el model. En el primer cas es tracta de la representació directe del volum, mentre que el segon és una visualització del model reconstruït.

Els tres tipus de reconstruccions que es poden realitzar a partir de models de vòxels són les següents:

- **RECONSTRUCCIONS 0D**

El primer cas correspon a les reconstruccions 0D. En aquest cas la representació simplificada del model de vòxels s'obté prenent punts del model distribuïts en determinades posicions. En el moment de generar la imatge final només es tenen en compte aquests punts. Perden la resta d'informació que teníem representada en el model. Per altra banda, treballem amb un model simplificat permetent que les visualitzacions siguin més ràpides ja que es disminueix el volum de dades a tractar.

- **RECONSTRUCCIONS 1D**

El segon cas es correspon a les reconstruccions 1D. En aquest cas el model de vòxels es processa per llesques. Per cada llesca s'extreu un contorn, en particular el contorn que té cert valor de propietat (per exemple la frontera d'un ós). Aquest procés es repeteix per totes les llesques i la imatge final s'obté simplement visualitzant els diferents contorns. Existeixen variants d'aquests algorismes en els quals a més d'extreure els contorns, es realitza un procés d'unió entre els contorns. Igual que en l'opció anterior es treballa amb un model simplificat a canvi de perdre informació del model inicial.

- **RECONSTRUCCIONS 2D**

El tercer cas és el de la reconstrucció de superfícies. Aquesta tècnica juntament amb la visualització directa de volums és la més aplicada.

En el cas de la reconstrucció que es portarà a terme amb les dades que rebrem a partir del TAC, farem una reconstrucció 2D. I a partir d'aquesta reconstrucció farem el tractament i visualització segons les necessitats determinades pels especialistes.

A partir de la informació obtinguda pels dispositius específics es pot crear, mitjançant les imatges, un model de volum. Aquest model s'obté col·locant les imatges una sobre l'altre i representant la informació amb el model escollit com es mostra en la figura 2.5-1.

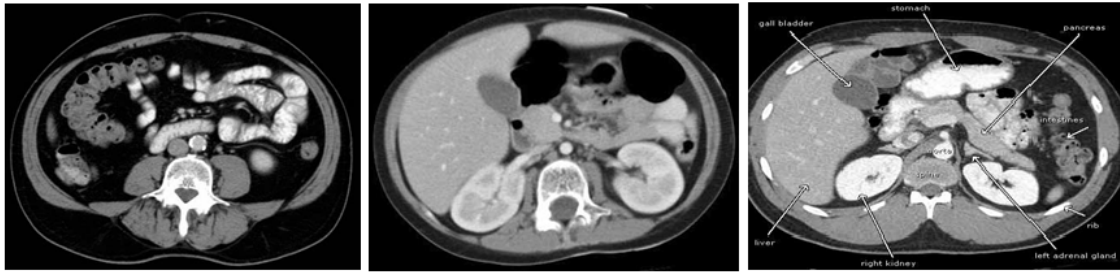


fig. 2.5-1 Exemple reconstrucció 3D

## 2.6 Visualització

La darrera etapa és la de **visualització**. En aquesta fase s'aplica alguna tècnica de visualització sobre el model de dades (o el model simplificat) per tal d'obtenir la imatge final. Aquest mètode de visualització variarà en funció del model de representació. La tècnica de visualització haurà de determinar quins atributs gràfics s'assignen als diferents valors de propietat representats en el model per tal que es pugui obtenir una representació gràfica del mateix. Evidentment el tipus d'atributs i les tècniques a aplicar variaran en funció del tipus d'informació que volem visualitzar i del tipus d'informació que tinguem representada en el model. No és el mateix visualitzar el model de vòxels que visualitzar un model reconstruït. Ja que la informació del model es pot representar a partir de punts, de contorns, de superfícies o bé a través de vòxels. La visualització en cadascun dels casos serà diferent ja que el contingut de dades es diferent.

Bàsicament, es consideren dues famílies d'algorismes de visualització: els que visualitzen superfícies i els que visualitzen tot el model de vòxels. Ambdues tècniques es descriuen en detall en l'apartat de *3.3.1 Tècniques utilitzades en el sistema d'exploració* dins el subapartat *Visualització adaptada al sistema d'exploració del budell*.

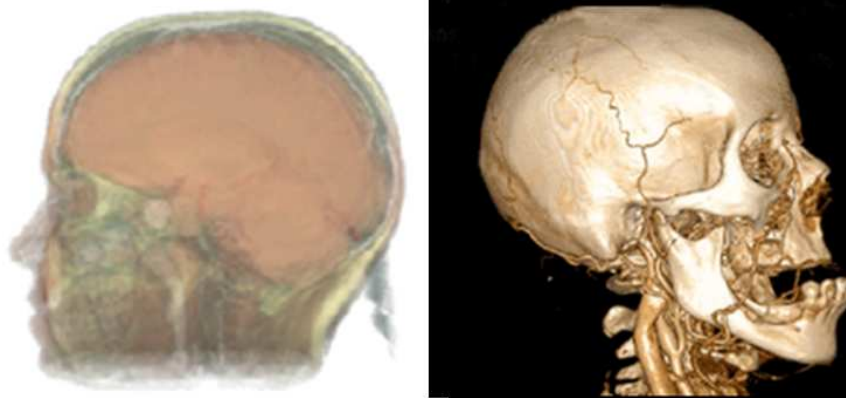


fig. 2.6-1 Visualització de volums i Visualització de superfícies

## 2.7 Problemes a resoldre

Després d'haver fet la introducció i les explicacions preliminars, cal proposar-nos les preguntes més importants per assolir els nostres objectius i explicar la manera de resoldre-les. Seguidament explicarem cadascun d'aquests requeriments que necessitem complir.

### DISSENYAR I IMPLEMENTAR UN SISTEMA PER CREAR MODELS SINTÈTICS DEL BUDELL

Els models reals que ens interessin per a desenvolupar el nostre projecte parteixen d'un format DICOM obtingut mitjançant dispositius específics per captació d'imatges mèdiques. En el nostre cas en concret, el TAC.

- **Els models sintètics**

En alguns casos, al no tenir els aparells necessaris, no és possible obtenir els fitxers DICOM amb dades reals. En aquests casos es reconstrueixen simulacions dels cossos reals a estudiar. Aquestes simulacions s'anomenen *phantoms* o models sintètics.

Amb les simulacions es poden realitzar proves i testejar els algoritmes per comprovar-ne la fiabilitat. En aquests casos es poden comprovar directament els resultats finals amb l'estructura i la informació del model sintètic.

En el nostre cas d'estudi, degut a la dificultat per obtenir dades inicials, s'ha d'implementar un mòdul que generi un model sintètic, amb les mateixes característiques que els models reals.

Per poder generar el volum artificialment necessitarem crear una estructura similar al contingut real de les dades. Per tant, haurem de crear una estructura virtual que representi un colon. Per fer-ne la representació crearem un tub, podent assignar-li si ens interessa tumors que seran representats a partir de esferes o el·lipses. D'aquesta forma obtenim un volum simple que s'aproxima al model real i ens permetrà fer un testeig dels algoritmes a implementar.

### CERCAR EL CENTERLINE D'UN VOLUM DE VÒXELS DE BUDELL

En el cas estudiat, la reconstrucció del budell, la visualització i modelatge es complica. Degut a que el TAC pèlvic a part del budell també conté altres òrgans a la zona observada que es converteixen en soroll del nostre fitxer. A més ens trobem en una zona on el soroll dins de les dades augmenta ja que són àrees on els òrgans els afecte un canvi o que contenen variacions en l'interior. En el cas del budell, el contingut de gas o excrements afegeix un punt de dificultat en la segmentació.

Per altra banda la forma del budell també en dificulta el modelatge. Per tant tenim una imatge amb molt de soroll i una dificultat afegida en la modelització del volum.

Aquests problemes també compliquen la cerca de la trajectòria del budell. El fet d'obtenir aquest format d'imatge amb tant soroll i amb una forma poc homogènia complica la cerca del centre.

S'haurà de desenvolupar un mètode que permeti trobar el centre del budell tot i la complexitat de les dades.

#### **DISSENYAR I IMPLEMENTAR UN SISTEMA D'EXPLORACIÓ DEL BUDELL**

La colonoscòpia és el mètode de diagnòstic amb el qual treballarem. A partir de les dades obtingudes amb el TC pèlvic, reconstruirem tridimensionalment el budell i en farem un recorregut virtual a través de la trajectòria.

El nostre objectiu no és obtenir una eina que permeti fer colonoscòpies sinó desenvolupar una tècnica que s'aproximi a la colonoscòpia.

Un cop s'hagi trobat la trajectòria implantarem mètodes que permetin a l'usuari viatjar a través del budell, visualitzant les zones afectades per pòlips, i si fos el cas, combinant aquest apartat amb el de segmentació.

#### **AVALUAR LES TÈCNiques DE PROCESSAMENT D'IMATGE QUE CALEN PER DIAGNOSTICAR ALGUNES DE LES LESIONS DE BUDELL.**

La segmentació del volum ens permetrà identificar diferents zones del model, identificant per separat el budell i els pòlips. D'aquesta forma podrem marcar a nivell visual les zones lesionades de forma que puguin ser identificades.

Estudiarem les tècniques avaluades per el grup de desenvolupadors de l'Starviewer, a partir de les opcions que tinguem haurem adaptar-les per saber identificar quines zones són les que interessin als especialistes per poder segmentar i separar correctament les parts que ens interessin.

#### **INTEGRAR TOTS ELS MÒDULS IMPLEMENTATS EN LA PLATAFORMA STARVIEWER**

Aquest apartat s'anirà realitzant a mesura que el projecte avanci, i segons les necessitats de cada mòdul. La forma d'integrar les aplicacions seran afegint noves extensions per cada funcionalitat, tal com esta desenvolupat fins al moment.

## 2.8 La plataforma starviewer

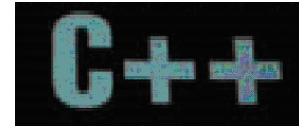
### 2.8.1 Eines utilitzades

Les eines utilitzades per la implementació són: el llenguatge C++, les llibreries d'interfície gràfica Qt, i les de tractament d'imatges i volums ITK's i VTK's, respectivament. L'ús d'aquestes eines i no d'altres, ve donat pel fet de que l'aplicació de la qual partim i en la que haurem d'integrar els mòduls, és a dir, l'Starviewer, està implementada amb aquestes mateixes eines de programació.

#### C++

---

Es tracta d'un llenguatge de programació basat en C, estandaritzat (ISO/IEC 14882 1998-International Organization for Standardization/International Electrotechnical Commission) i extensament utilitzat. Gràcies a aquesta estandarització i a la llibreria estàndard, C++ s'ha convertit en un llenguatge potent, eficient i segur, característiques que l'han convertit en un llenguatge altament extes tant en l'àmbit professional com l'educatiu, i competitiu davant altres llenguatges com C# de Microsoft o Java de Sun Microsystems.



Les característiques que fan del C++ un llenguatge tant utilitzat són:

- Programació Orientada a Obejectes (POO). Permet generar codi altament reutilitzable d'una forma lògica i productiva.
- Portabilitat. Pràcticament es pot compilar el mateix codi C++ en la gairebé totalitat d'ordenadors i sistemes operatius.
- Brevetat. El codi implementat en C++ es molt curt en comparació amb altres llenguatges degut a la facilitat amb la que es poden anidar expressions i a la gran quantitat d'operadors.
- Programació modular. El cos d'una aplicació en C++ es pot construir a partir de diferents fitxers font que es podran compilar per separat.
- Velocitat. El codi resultant d'una compilació en C++ és molt eficient degut a la seva dualitat com llenguatge d'alt i baix nivell i el reduït tamany del mateix.

#### Llibreries QT

---

Qt és una biblioteca multiplataforma per desenvolupar interfícies gràfiques d'usuari, és utilitzada en KDE, un entorn d'escriptori per sistemes com GNU/Linux o FreeBSD, entre altres. Utilitza el llenguatge de programació C++ de forma nativa *bindings* per C, Python (PyQt), Java (Qt Jambi), Perl (PerlQt) i Ruby (QtRuby) entre altres.



L'API de la biblioteca compta en mètodes per accedir a bases de dades de dades mitjançant SQL, Així com l'ús de XML i una multitud d'altres eines pel maneig de fitxers, a més d'estructures de dades tradicionals.

Les Qt són unes llibreries molt usades ja que són molt completes i robustes, com ho demostra que l'entorn gràfic de Unix KDE les utilitza.

### Widgets

Qt disposa d'un ampli nombre de *widgets*. Un *widget* és qualsevol element visual (botons, caixes de text, pestanyes, etiquetes, *scroll bars*, menús, etc.) que formi part de la interfície gràfica. Cada tipus de *widget* és una subclasse de *QWidget*. A més, es poden crear fàcilment nous *widgets* que també seran subclasses de *QWidget*. Només cal crear una subclasse de qualsevol *widget* de Qt ja existent. Aquesta és una altra de les raons pel que s'han escollit aquestes llibreries, per la seva flexibilitat.

### Signals&Slots

El *signals&slots* és un sistema que serveix perquè els objectes de qualsevol tipus es comuniquin entre ells. Un *signal* és un avís que dona un objecte quan ha passat algun succés (*event*), per exemple un botó pot emetre un *signal* "clicked" quan se'l prem. I un *slot* és aquella rutina que està esperant un senyal per executar-se, per exemple finalitzar l'execució d'una aplicació. De fet, la implementació d'un *slot* és com la d'un mètode qualsevol. Per connectar un *signal* amb un *slot* (o un altre *signal*) es crida la funció *connect()*.

Aquest sistema de comunicació és probablement la característica que més diferencia les Qt de la resta de llibreries d'interfícies gràfiques.

### QtDesigner

---

Cal destacar el QtDesigner, una eina visual que permet dissenyar aquestes interfícies gràfiques. Tota la interfície gràfica que hem dissenyat ha estat creada amb les llibreries Qt i utilitzant el QtDesigner.



### Llibreries ITK

---

ITK és un software lliure orientat a objectes per el processament, segmentació i registrament de imatges. Tot i la seva magnitud ITK estan dissenyades per facilitar l'ús de les llibreries.



**NLM Insight**  
Segmentation & Registration Toolkit

A més, els usuaris i dissenyadors de ITK poden aportar una gran contribució en el desenvolupament del software ja que mitjançant el "foro-log" dels usuaris o dissenyadors es poden proporcionar informes de *bug*, *bug* fixats, tests, i noves classes que podran utilitzar els demés usuaris i/o dissenyadors.



Les llibreries ITK estan desenvolupades en C++. ITK es “*cross-platform*”, utilitza el CMake per controlar la compilació i linkatge de les classes. Així permet utilitzar el nucli de llibreries desenvolupades en C++ conjuntament amb altres llenguatges amb els que es poden utilitzar les ITK, com son TCL, Java i Python.

Característiques tècniques de NLM's Insight ITK toolkit:

- El toolkit proporciona representació de dades i algoritmes per a la segmentació i registra. El focus és sobre aplicacions mèdiques; encara que és capaç de processar altres tipus de dades. Proporciona representacions de dades en format genèric per a imatges (dimensió arbitrària) i (desestructurat) xarxes.
- El toolkit no permet generar interfície de visualització d'usuari o gràfica. Aquesta propietat es delega a altres eines específiques de visualització i interfície gràfica (com VTK, VisPack, 3DViewnix, Metalimage, etc.)
- El toolkit proporciona un mínim d'utilitats per la gestió d'arxius. Aquesta feina també queda relegada a altres toolkits/libreries específics per aquesta gestió.
- Suporta l'execució paral·lela amb fils d'execució.
- El desenvolupament del toolkit es basa en Extrem Programming

Característiques d'Estructura de NLM's Insight ITK toolkit:

- El toolkit s'organitza al voltant d'una arquitectura de flux de dades. És a dir, les dades es representen utilitzant objectes de dades que són processats per filtres. Els objectes de dades i objectes de procés estan connectats mitjançant pipelines. Les pipelines són capaces de processar les dades per parts segons un límit de memòria especificat per l'usuari.
- S'utilitzen objectes Factory per instanciar els objectes.
- Un patró de disseny de comand/observer s'utilitza per al processament d'events.

### ***Llibreries VTK***

---

VTK és un sistema per processar imatges i visualització 3D de codi obert disponible per una gran quantitat de plataformes. És un paquet clarament orientat a les tècniques de POO compost per més de 700 classes en C++, encara que també disposa d'interfícies perquè pugui ser utilitzat des de Tcl/Tk, Java i Python. Disposada d'algoritmes molt avançats de modelat, visualització, mètodes volumètrics, etc.



Les VTK (Visualization Toolkit) són unes llibreries de codi lliure, enfocades a gràfics 3D, processat d'imatges i visualització utilitzades mundialment.

### - **Filosofia d'implementació**

Aquest kit d'eines està estructurat en dos subsistemes: un nucli que és una llibreria de classes implementades i compilades en C++ i un "embolcall" amb varis llenguatges interpretats que permeten utilitzar la llibreria mitjançant llenguatges que no siguin necessàriament C++ com poden ser Tcl, Java o Python. La implementació de les llibreries VTK ha estat seguint els principis de l'orientació a objectes i són multiplataforma (han estat instal·lades i testejades en tot tipus de plataformes existents actualment: Unix, Windows i MacOSX).

### - **Arquitectura de les VTK**

Les VTKs es poden dividir en dos grans mòduls, el mòdul de gràfics i el mòdul de visualització:

El **Graphics Model** consisteix en una sèrie d'objectes bàsics que combinats creen una escena. Aquests objectes bàsics són:

- **Actor**
- **Light**
- **Camera**
- **Property**
- **Mapper**
- **Transform**
- **Renderer**
- **RenderWindow**

Un *actor* representa allò que "veiem" en l'escena, els *lights* il·luminen els *actors* en una escena i la *càmera* defineix la posició de l'observador, el punt cap a on mira i les altres característiques d'una càmera, determinant com la geometria tridimensional es projecte en imatges bidimensionals. Els actors estan definits per objectes de tipus *mapper*, *property* i *transform*. Un *mapper* representa la definició geomètrica de l'*actor*, un *property* representa els atributs de *rendering* de l'*actor* (com color, textura, especularitat, *shading*) i un objecte de tipus *transform* especifica la posició i orientació d'un *actor*, *camera* o *light*. Finalment, un *renderer* com el seu nom indica coordina el *rendering* dels llums, càmeres i objectes de l'escena i un *renderWindow* no és res més que la finestra que engloba una escena (en una mateixa escena s'hi poden "dibuixar" varis *renderers*).

El **Visualization Model** és l'encarregat de construir la representació geomètrica que després es *renderitza* en el model de gràfics. Aquest model és molt semblant a les ITKs, també està basat en el paradigma *data-flow* i els seus dos tipus d'objectes bàsics també són *data-object* i *process-object*. Com ja hem dit en l'apartat de les ITK, els primers guarden les dades a tractar i els segons els algoritmes que es poden aplicar als primers. Evidentment, també tenen els mètodes *SetInput()* i *GetOutput()* per poder-se connectar formant *pipelines* per on passa el flux de dades (*data-flow*).

## ***StarUML - The Open Source UML/MDA Platform***

---

*Unified Modeling Language* (UML) és un llenguatge que va ser dissenyat per reduir la complexitat del disseny del sistema de software. La notació específica de la llengua i del procés s'ha substituït per una notació visual universal que facilita la comunicació dins de tots els nivells del desenvolupament d'analistes del negoci amb coneixement de programació limitat fins al final als arquitectes majors del programari.



StarUML és un projecte de codi lliure de disseny UML que pot suplir perfectament una plataforma comercial.

UML 2.0 *Unified Modeling Language* (Llenguatge de Modelat Unificat) és un llenguatge per especificar, dissenyar, construir i documentar sistemes, inicialment de programari orientat a objectes.

L'UML intenta definir un llenguatge homogeni (un llenguatge gràfic) per modelar totes les fases del desenvolupament d'una aplicació. Des de les especificacions del programa per part del client fins al disseny detallat per al programador. L'UML és un llenguatge per especificar, i no un mètode o un procés. Ofereix un estàndard per descriure un model de sistema, incloent aspectes conceptuals com ara processos de negocis i funcions del sistema, i aspectes concrets com expressions de llenguatges de programació, esquemes de bases de dades i components de programari reutilitzables. L'UML es pot usar en una gran varietat de formes per suportar una metodologia de desenvolupament de programari (tal com el Procés Unificat de Rational), però no especifica en si mateix quina metodologia o procés usar.

Va ser ideat, segons la definició de *Rational*, per a especificar, construir, visualitzar i documentar els diversos aspectes d'un sistema basat en programari. Té l'avantatge de ser un llenguatge gràfic i fàcil d'entendre.

És el llenguatge d'aquest tipus més conegut i utilitzat en l'actualitat; tot i que encara no és un estàndard oficial, està recolzat en gran manera pel OMG (Object Management Group).

*MDA (Model-Driven Architecture o MDA)*: La arquitectura dirigida per models és una aproximació al disseny de software, proposat i patrocinat per el *Object Management Group*. MDA s'ha concebut per donar suport a l'enginyeria dirigida a models dels sistemes de software. MDA és una arquitectura que proporciona un conjunt de guies per estructurar especificacions expressades amb models. El principi bàsic de l'MDA és l'elaboració de models independents de la plataforma (Platform Independent Model, PIM) i la seva transformació a models dependents de la plataforma (Platform Specific Model, PSM) per obtenir la implementació pròpia del sistema.

*Plug-in Architecture*: Degut a la variabilitat i necessitat dels usuaris a adaptar-se a diferents llenguatges, StarUML proporciona una potent i compacte arquitectura de mòduls que poden ser desenvolupats en diferents llenguatges (com C++, Delphi, C#, VB, ...) permetent-ne la compatibilitat.

*Usabilitat*: La usabilitat és molt important en un software de desenvolupament. StarUML està implementat de forma que faciliti la feina al desenvolupar un interfície que agilitza la interacció

del usuari amb interaccions amb el teclat, manipulació del diagrama per teclat, previsualitzacions de diagrames, etc.

StarUML està majoritàriament desenvolupat en Delphi. Però és un projecte multilingatge, per tant es pot utilitzar qualsevol llenguatge per desenvolupar-lo (per exemple, C/C++, Java, Visual Basic, Delphi, Jscript, VBScript, C#, VB.NET,..)

### ***Doxygen***

---

*Doxygen* és un generador de la documentació a partir de codi per C++, C, *Objetivo-c*, *Python*, IDL, entre altres llenguatges. Funciona en la majoria dels sistemes del Unix així com en OS X, de Windows i del Mac.



El Doxygen és un sistema de documentació per C++, C, Java i altres llenguatges i és multi-plataforma. Aquesta eina ens permet generar documentació a partir del codi i els comentaris que hi posem. Al comentar tot el codi s'aconsegueix un document força extens amb la documentació de totes les classes i mètodes i què fa cadascuna d'elles.

Això s'aconsegueix comentant cada classe i mètode just abans de definir-los.

## 2.8.2 Starviewer

Actualment la plataforma permet visualitzar volums extraient les dades a partir de fitxers .mhd, o a partir de fitxers DICOM. A més permet realitzar les operacions més bàsiques per interactuar amb el model, fer rotacions i zooms per aproximar-se al volum reconstruït.

Les interfícies de l'aplicació funcionen mitjançant extensions que s'afegeixen per les noves funcionalitats, construint una pantalla de les finestres organitzades en diferents pestanyes per cada mòdul. Permetent obtenir diferents funcionalitats sobre un mateix volum.

Tot seguit mostrem un diagrama de classes que mostra algunes de les principals classes del projecte Starviewer tal i com estava inicialment i amb les que mes s'ha treballat durant el desenvolupament i implementació del projecte.

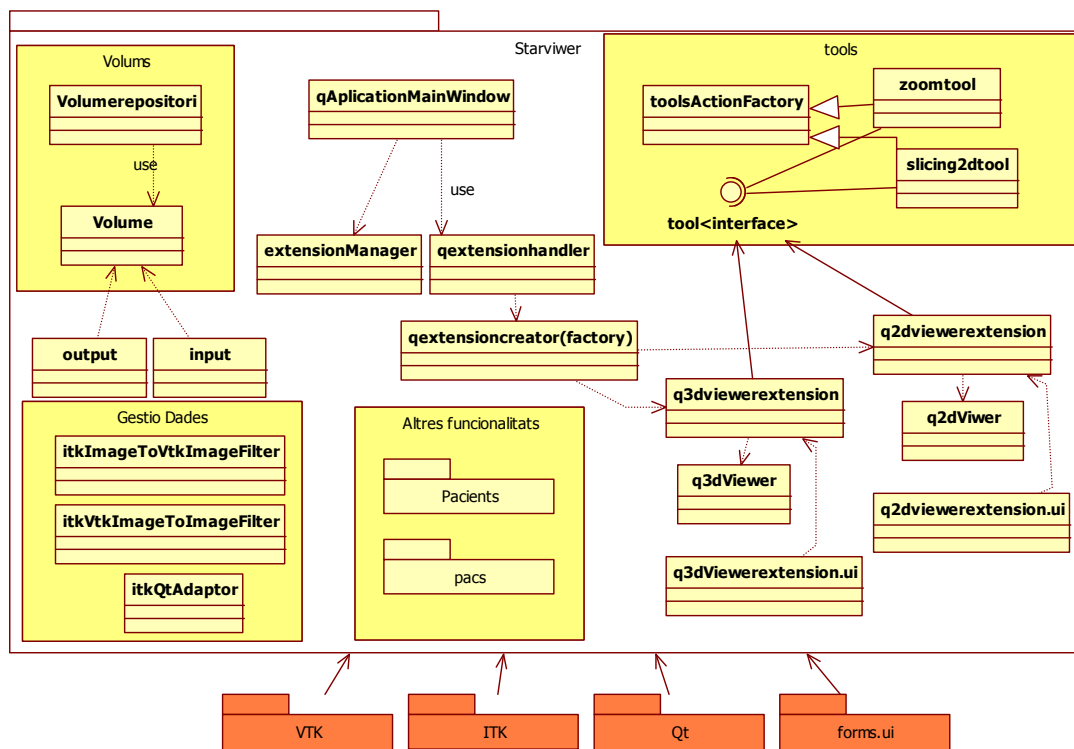


fig. 2.8-1 Diagrama inicial de classes Starviewer

Tal com veiem en la imatge anterior, l'Starviewer està format per un conjunt de classes, programades a partir de llibreries externes com són, itk i vtk, pel tractament de les dades mèdiques i imatges, i Qt i els formularis .ui per a programar la part de la interfície gràfica de l'aplicació. Tota la interfície gira al voltant de la classe que podríem anomenar "principal", la classe QapplicationMainWindow, que ens implementa el que és la interfície en sí. Tenim classes per a capturar paràmetres d'entrada, per a poder executar les funcionalitats i a banda també tenim les conegudes com a classes Director, que s'encarreguen d'interactuar entre la part gràfica de l'aplicació i els procediments interns que s'executen. Per acabar la descripció sobre les classes de l'aplicació, cal destacar, les classes que componen el repositori de volums. Aquestes classes serveixen per a gestionar, de manera eficient, l'obtenció i manipulació dels volums, ja que són dades que ocupen molta memòria i cal gestionar-los degudament.

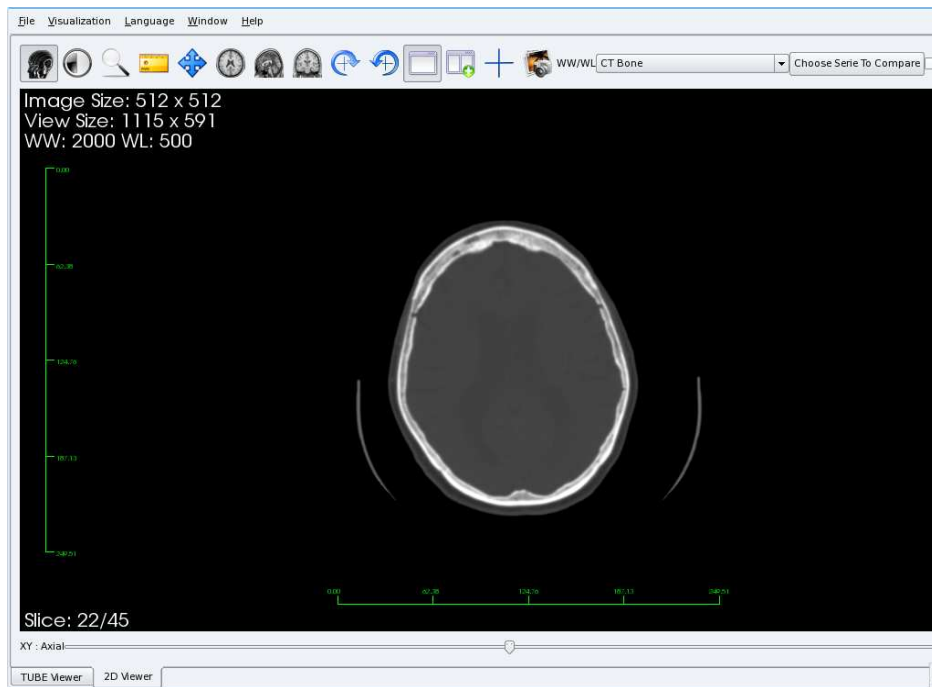


fig. 2.8-2 Interfície Starviewer

En la figura 2.7-2 que mostrem es pot veure la interfície de l'Starviewer. Format per el que seria l'extensió, interfície d'usuari, la barra de tools, el menú principal del programa i la barra de pestanyes per les diferent extensions que es tinguin obertes.



fig. 2.8-3 : Barra de Tools

La zona del de les tools és la barra superior on es pot selecciona la tool desitjada per aplicar-la en el visor que es vulgui. Els visors mitjançant una classe toolmanager gestionen els efectes de les tools sobre ells. Cada visor permet diferents accions sobre ell mateix, per tant si es té una tool activada però el visor sobre el que actuem no té permesa la funcionalitat simplement no actuarà.

Cada extensió que es genera mitjançant la classe factoryextensioncreator. Cada nova extensió creada es sobreposa com pestanyes per damunt les anteriors finestres.



fig. 2.8-4 : Pestanyes, permeten canvia d'extensió

### **3. FASE D'IMPLEMENTACIÓ**

En aquesta fase es farà l'anàlisi de requeriments, el disseny i implementació de l'aplicació. Avaluarem els requeriments funcionals del projecte, els quals es plasmaran utilitzant el llenguatge UML. També explicarem la part que fa referència al disseny del sistema informàtic. Mitjançant diagrames de classes, diagrames de seqüència i diagrames d'activitat, definirem l'estructura del projecte.

En la fase d'implementació diferenciarem les següents etapes :

- ***Disseny i implementació d'un sistema de generació de models sintètics***

On definirem l'estructura virtual que representarà el model sintètic, i desenvoluparem una funcionalitat per generar un volum que compleix els estàndards DICOM de l'estructura virtual.

- ***Cerca del centerline d'un volum de vòxels de colon***

Implementarem les classes necessàries per trobar el centre del budell.

- ***Estudiar i avaluar les tècniques de processament d'imatge que calen per diagnosticar algunes de les lesions de budell.***

Estudiarem les diferents tècniques de segmentació i visualització que s'han aplicat en altres mòduls de l'Starviewer i escollirem les que millor s'adapten a les nostres necessitats i que permetran diferenciar les dues parts d'interès del budell. S'ha fet conjuntament amb un altre projectista amb l'ajuda del desenvolupadors de l'starviewer que m'ha indicat quina segmentació em donaria mes bon resultat

- ***Dissenyar i implementar un sistema d'exploració del budell***

Desenvoluparem un mòdul que dotarà a l'usuari d'una utilitat per viatjar a través del budell, contemplant l'interior d'aquest a partir de la visualització del volum.

- ***Integració de tots els mòduls implementats en la plataforma Starviewer***

A mesura que es vagin desenvolupant les diferents funcionalitats o prestacions anteriorment esmentades, s'aniran complementant i afegint a l'Starviewer de forma homogènia.



### 3.1 Disseny i implementació d'un sistema de generació de models sintètics

Per poder generar els models sintètics el primer que s'ha de fer és definir una estructura de dades que representi una estructura tridimensional similar al volum real. En el nostre cas la representació més adient és un tub que simularà el budell i el·lipses de diferent mida per representar els pòlips.

Per poder fer la reconstrucció 3D del model sintètic, com una representació simplificada de l'estructura del budell. Ens caldrà definir l'estructura del tub.

El budell sintètic vindrà determinat per un conjunt de punts que definiran una trajectòria, i un radi que determinarà el diàmetre del budell en cada punt de la trajectòria. Tal com es mostra a continuació en la figura 3.1-1.

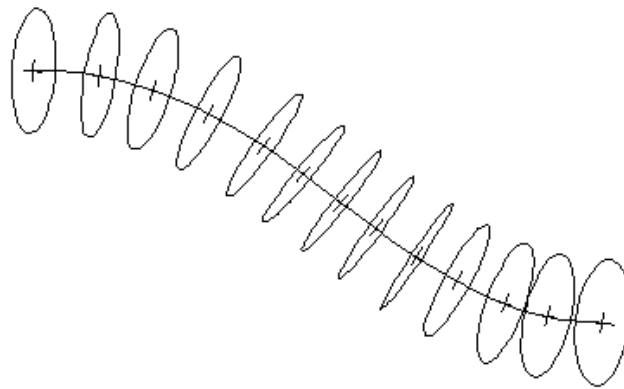


fig. 3.1-1 Representació del model sintètic

El procés que hem de seguir és una reconstrucció 3D a partir d'una estructura virtual. Hem de simular una reconstrucció d'un volum.

En imatges planes 2D, definides per plans de tall paral·lels, respecte a algun dels plans de projecció (frontal, coronals o axial) del volum, en el nostre cas l'estructura virtual. Hem de connectar els plans de forma que pla a pla anem reconstruint l'estructura definida. D'aquesta manera obtindrem el volum virtual.

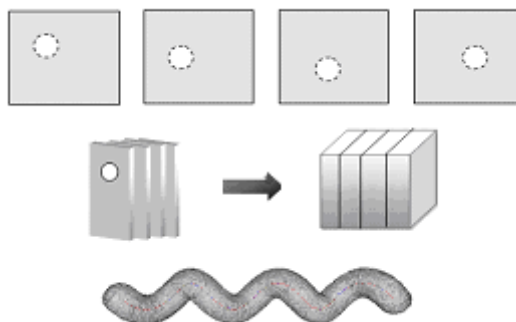


fig. 3.1-2 Exemple de reconstrucció d'un tub a partir els pans de tall

Per reconstruir tridimensionalment el volum utilitzarem les llibreries ITK que ens proporcionen un suport idoni per aquesta feina. En concret, utilitzarem la classe `SpatialObjects` i herències d'aquesta classe de itk que permeten generar objectes simples.

### *Spatial Objects*

Els `Spatial Object` proporcionen un suport de consulta, manipulació i interconnexió entre objectes en espais físics. Mitjançant `spatial objects` es pot representar un volum real i en facilitar la feina a l'hora de accedir, calcular o transformar valors de l'objecte.

Els `spatials` permeten definir un conjunt de objectes que mantenen una relació de pares/fills generant un arbre n-ari. On els fills de cada node seran objectes que dependran de l'anterior. Aquesta estructura ens permetrà fer un càlcul global sobre el conjunt d'objectes, o obtenir la informació referent a tot el grup com ara l'àrea ocupada, entre altres.

Una de les característiques principals és que a partir d'un objecte `spatial object` i un punt determinat es pot saber fàcilment si el punt en concret cau dins l'àrea del volum definit per l'objecte. Aquesta propietat també s'utilitza sobre el conjunt d'objectes fills de l'objecte consultat. Definint fins a quin nivell de l'arbre creat ens interessa la informació podem saber si un punt es troba dins l'àrea definida pel pare i els seus predecessors fins al nivell de l'arbre definit.

També aporten un suport d'escriptura i lectura de l'estructura, d'objectes o conjunts de `spatial objects`, en formats de fitxers plans de text. El que permet guardar la informació amb un fitxer de poc tamany per posteriorment carregar-la i generar de nou l'estructura.

En el nostre cas els `spatial objects` ens serviran per definir l'estructura virtual, i per guardar-la en un format de fitxers de text pla. De manera que podem guardar la informació del volum pregenerat, i el podem tornar a generar a partir del fitxer de text.

A més, les classes `SpatialObject` ens permetran interactuar-hi afegint, eliminant o modificant punts de l'estructura. I finalment generarem el volum amb l'ajuda dels filtres que proporciona itk per convertir `spatialobjects` a imatges de n dimensions.

### **DEFINICIÓ DE L'ESTRUCTURA VIRTUAL**

Tal com hem dit anteriorment per recrear un colon sintètic definim un tub que simularà el budell i el·lipses per simular els pòlips o tumors. L'estructura del tub la generarem a partir d'esferes amb menor o major radi per definir l'amplada del tub en cada punt de la trajectòria.

Les classes de itk que utilitzarem a l'hora de generar l'estructura seran la itkSpatialGroup per agrupar tots els objectes que formaran l'estructura i la itkEllipseSpatial per definir les esferes i/o el·lipses del budell. Els el·lipses tant ens serviran per representar les esferes que formaran el cos del tub, com els que simularan els pòlips. Per tant l'estructura final quedarà definida per un conjunt de el·lipses que sobreposades les unes amb les altres formaran un budell amb o sense tumors.

#### **3.1.1 Generació del model sintètic**

Els passos a seguir a l'hora de definir un model sintètic i generar-ne un volum són els següents:

1. Recollir dades de la trajectòria definida per l'usuari.
2. Construir l'estructura del budell sintètic a partir de la trajectòria definida per l'usuari.
3. Generar un volum a partir de l'estructura definida.
4. Guardar el volum generat.

A continuació explicarem pas a pas el procés que seguim al generar el budell:

#### **Definició de la trajectòria**

Definir tots els punts que formen la trajectòria d'un budell manualment pot resultar una feina molt costosa. Per facilitar la feina a l'usuari, a l'hora de definir un budell, permetrem introduir alguns punts significatius de la trajectòria separats entre ells per una certa distància. Els punts intermedis entre ells es generaran automàticament.

A través de la interfície permetrem definir l'estructura del budell. Es podran introduir els punts assignant una posició i un radi per cadascun d'ells. A més es podrà decidir si el budell sintètic contindrà tumors o no, els quals seran generats de forma completament transparent a l'usuari.

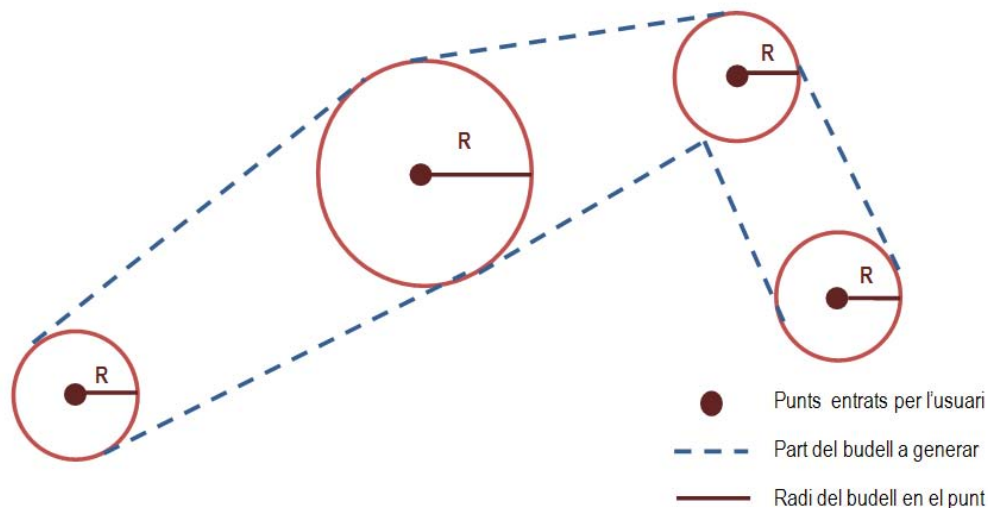


fig. 3.1-3 Informació introduïda per l'usuari a l'hora de definir un budell sintètic.

Per altra banda també es permetrà definir alguns aspectes referents a la imatge com ara l'spacing o si s'ha d'aplicar soroll sobre la imatge per simular una distorsió en la recollida de les dades del volum.

Un valor important a l'hora de definir una imatge és l'spacing donat, ja que indica l'espai entre vòxels. Un spacing diferent sobre una mateixa estructura pot donar un aspecte completament diferent a un mateix volum.

### ***Construcció del budell virtual***

Després d'haver introduït tots les dades procedirem a construir l'estructura virtual que definirà el budell.

Com hem comentat al punt anterior, per agilitzar la generació de models sintètics hem automatitzat la generació de la trajectòria a partir d'alguns punts característics que la defineixen. Que són els punts que haurà d'entrar a l'usuari manualment.

El nostre propòsit és unir dos punts entrats per l'usuari dibuixant els punts intermedis entre ells que passen per sobre el segment que els uneix. Per trobar aquests punts intermedis utilitzarem la fórmula de la distància entre els dos punts.

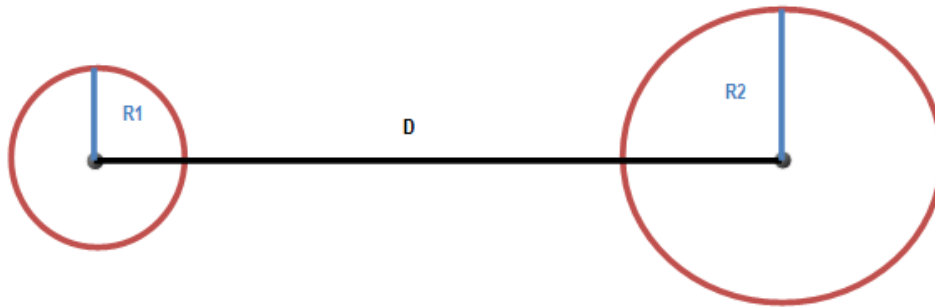


fig. 3.1-4 Representació de dos punts entrats per l'usuari amb radis R1 i R2

Donats dos punts  $P(X_1, Y_1, Z_1)$  i  $Q(X_2, Y_2, Z_2)$  la distància  $d_{PQ}$  entre els dos punts és la longitud del segment que uneix els punts. Que es correspon a la fórmula següent:

$$d = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2}$$

Per saber el nombre d'esferes que hi ha entre els dos punts ho calculem a partir de la distància  $D$  entre els 2 centres i el radi més petit de les dues esferes. Dividim la distància entre el radi obtenint així el nombre de esferes, de radi més petit, que necessitem per unir les esferes. Tal com mostrem a continuació:

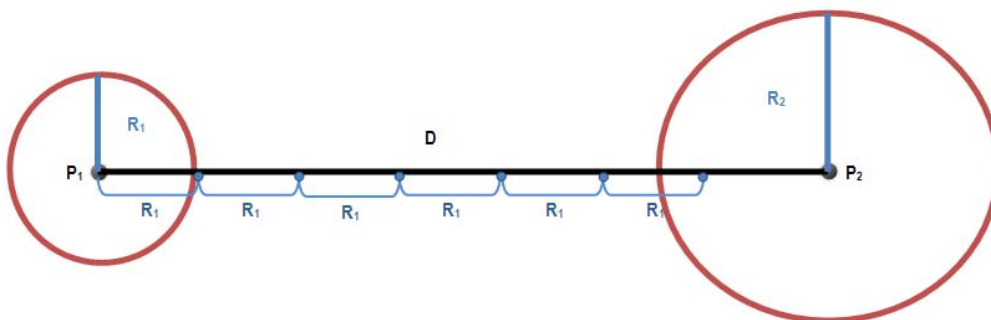


fig. 3.1-5 Càlcul dels punts intermedis a partir del radi

En l'exemple de la figura anterior per saber el nombre d'esferes necessàries, per unir els dos punts donats  $P_1$  i  $P_2$ , dividirem la distància  $D$  per  $R_1$ . Així obtenim el nombre d'esferes amb radi  $R_1$  que ens calen per unir les dues esferes amb centres  $P_1$  i  $P_2$ .

Quan ja sabem el numero d'esferes en calculem les posicions i el radi per cada nova esfera que haguem de generar. Podem calcular l'increment  $x$ ,  $y$  i  $z$  i el valor d'increment del radi a partir del valor de les components i el radi del punt inicial i les components i el radi del punt final.

Per calcular l'increment aplicable a cada variable que definirà les esferes intermèdies ho calculem de la següent manera:

$$\text{Increment} = \frac{\text{Valor Inicial} - \text{Valor Final}}{N_{\text{puntIntermedis}}}$$

Amb aquesta fórmula obtenim l'increment que hem d'aplicar a les variables de posició i al radi, per cada nova esfera creada. D'aquesta manera les esferes generades son connexes entre sí i acaben convertint-se en un tub que connecta els punts entrats per l'usuari.

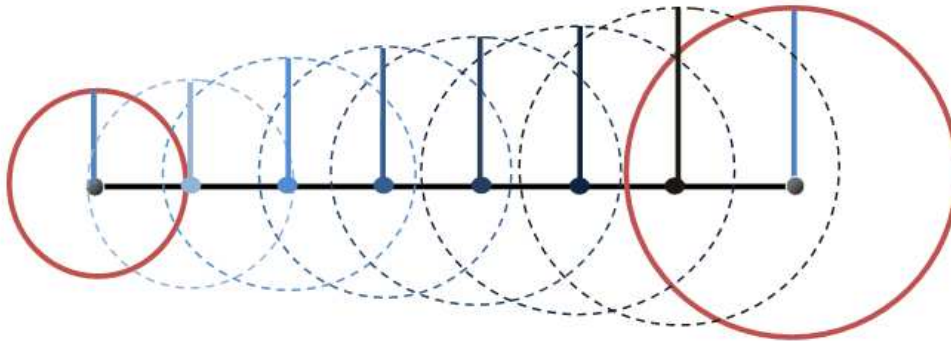


fig. 3.1-6 Generació d'esferes intermèdies amb increment de radi

A part de calcular les esferes intermèdies entre els punts entrats per l'usuari, també es genera automàticament i aleatòriament els el·lipses que representen els tumors. Cada tumor se li assigna una posició i un radi de forma que el tumor quedi sobre la paret del budell sintètic.

Permetrem generar 1 tumor com a màxim per cada punt generat. Escollim aleatòriament els punts sobre els que hi situarem un tumor amb un percentatge determinat.

Al definir la posició del tumor ho farem a partir del centre de l'el·lipse sobre el que es generarà el tumor. Les posicions possibles del tumor estaran definides per l'àrea interior a l'esfera de radi igual al diàmetre de l'esfera a la que li posarem el tumor. Tal i com es mostra a la figura 3.1-7.

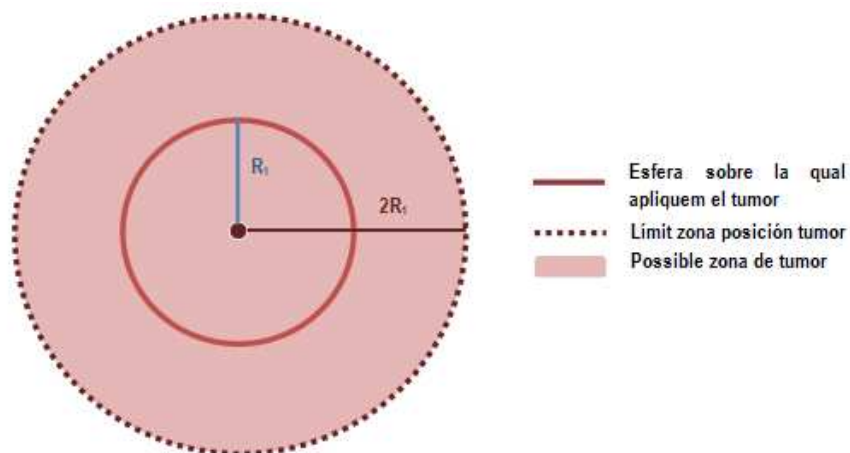


fig. 3.1-7 Zona sobre la que es pot generar un tumor

Per calcular la posició del tumor aplicarem a cada component de l'espai  $x$ ,  $y$  i  $z$  del centre de l'esfera un increment, calculat aleatòriament entre 0 i  $2R$ , que ens definirà una posició del centre del tumor.

Amb això aconseguim que el centre del tumor sigui el centre de l'esfera més un increment sobre aquest. Si entenem els centres dels el·lipses com a els vectors que els defineixen obtenim la següent fórmula:

$$V_t ( X_t, y_t, Z_t ) = V_e ( X_e, y_e, Z_e ) + V_i ( X_i, y_i, Z_i )$$

En la que intervenen els vectors  $v_t$ , que representa el centre del tumor,  $v_e$ , vector que representa el centre de l'esfera, més  $v_i$ , un vector increment, les components del qual són valors aleatoris entre  $-2R_1$  i  $2R_1$ .

$$x_i = ( 2R \times \text{aleatori}(-1, 1) )$$

$$y_i = ( 2R \times \text{aleatori}(-1, 1) )$$

$$z_i = ( 2R \times \text{aleatori}(-1, 1) )$$

Sumant els vectors  $v_i$  i  $v_e$  obtindrem un nou vector que indicarà la posició que donarem al tumor.

Calculada la posició on col·locarem el tumor en definim els radis. S'haurà de tenir en compte la posició del tumor ja que la paret de l'esfera i la del tumor han d'interceptar per simular correctament un tumor del budell.

Calcularem la distància a la que es troba el centre del tumor amb el centre de l'esfera que el conté. A partir d'aquesta distància i el radi de l'esfera obtenim la distància del centre del tumor a la paret del budell sintètic, i aquesta distància ens defineix el radi mínim que ha de tenir el tumor.

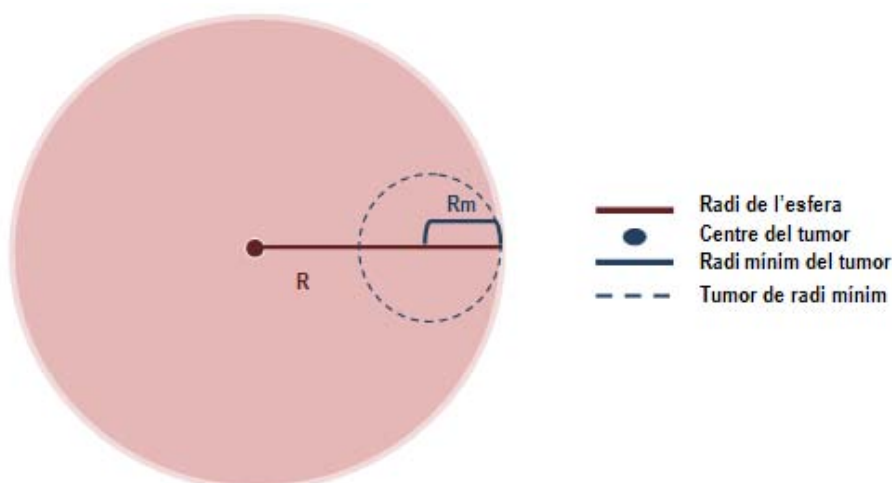


fig. 3.1-8 Càlcul del radi mínim del tumor respecte la posició

Sobre el radi mínim aplicarem uns increments variables per generar tumors de diferents formes el que evitarà que tots els tumors generats siguin iguals.

La imatge següent mostra la generació completa de l'estructura d'un budell sintètic. A partir de la trajectòria definida entre dos punts, entrats per l'usuari.

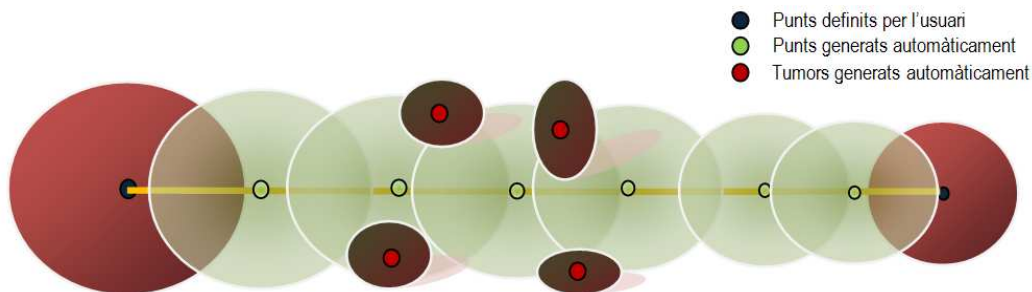


fig. 3.1-9 Generació model sintètic

### ***Generació del volum***

Un cop construïda l'estructura del budell haurem de convertir-la en una imatge 3D. Per realitzar aquest procés les llibreries itk donen un suport específic que permet convertir l'*SpatialObject* en una imatge binària.

La classe concreta per utilitzar aquesta funcionalitat es la *itkSpatialObjectToImageFilter* una classe que, a partir d'un origen i una mida que defineixen la imatge, pinta la part del *spatialobject* que cau dins dels límits de la imatge. Pintant l'interior del *spatialobject* d'un color i l'exterior d'un altre formant una imatge volumètrica que simula perfectament una imatge segmentada.

El problema que tenim és que aquesta funcionalitat només pot pintar 2 parts sobre el volum, dins i fora. I el que ens interessa és pintar dins, fora i a més els tumors d'un tercer color. Per simular una imatge segmentada d'un budell. Com la que mostrem a continuació.

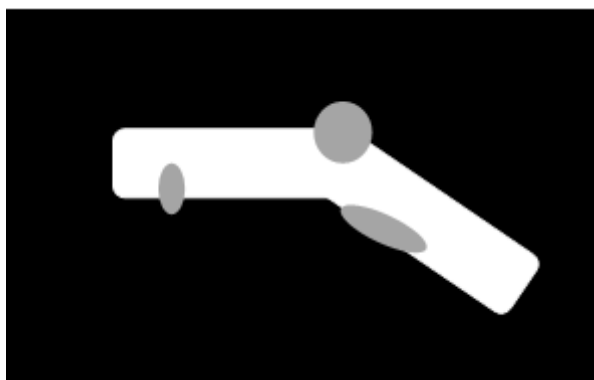


fig. 3.1-10 Segmentació budell i tumors

Per poder solventar aquest problema desenvoluparem un mètode que recorre tota la imatge i que consultant el grup d'objectes *saptial* determinarà si el punt de la imatge es troba dins algun tumor de l'objecte generat, i en el cas que s'hi trobi el pinti. La resta de la imatge no es modifica.



Seguint els consells de les llibreries hem utilitzat els `itkImageIterators`. Classes de iteradors desenvolupats especialment per les llibreries que agilitzen el procés de consulta i modificació de les dades de la imatge.

Un cop haguem obtingut el volum, utilitzant l'`SpatialObject` creat juntament amb la classe `itkSpatialObjectToImageFilter`, obtindrem una imatge binària. On tindrem pintada tota la zona que formen el tub i els tumors de color blanc. Llavors crearem un iterador sobre la imatge 3D que recorrerem vòxel a vòxel repintant les zones que estiguin dins dels tumors.

La imatge següent mostra un exemple del procés a desenvolupar.



fig. 3.1-11 Imatge obtinguda final i imatge obtinguda de la conversió dels spatial objects

### ***Guardar el Volum***

Finalment guardarem el Volum generat amb el format DICOM generant un fitxer capçalera.mhd i un altre fitxer de dades.raw.

Aquesta funcionalitat ja estava implementada en l'`starviewer`, l'únic que hem fet es utilitzar les classes que s'havien desenvolupat per dur a terme aquest propòsit. La classe en concret es la `Output`, que a partir d'un volum `Vtk` és capaç de guardar en el format DICOM que és el que ens interessa.

Un cop guardada la imatge la podrem tractar d'igual forma que un volum d'un budell real.

Per altra banda ens interessava implementar una forma que ens permetés guardar l'estructura del budell amb un format que no fos el del volum. De manera que poguéssim transportar la informació sense que ocupes tant com el fitxers que es generen amb el format DICOM.

Les classes `itkSpatialObjectWriter` i `itkSpatialObjectReader` estan implementades expressament per aquest objectiu. Les utilitzarem per guardar i carregar les estructures dels volums definits per l'usuari amb un fitxer menys pesat. El pròxim cop que es vulgui visualitzar el volum, s'haurà de tornar a generar, però l'usuari no haurà de tornar a definir l'estructura.

La informació estructural dels SpatialObjects es guarda en fitxers de text amb extensió .meta. Aquest fitxers contenen la informació necessària per tornar a construir l'estructura que s'havia guardat.

Seguidament mostrem una part del contingut d'un fitxer .meta generat a partir d'un volum construït a partir de punts entrats per l'aplicació desenvolupada:

```
ObjectType = Scene
NDims = 3
NObjects = 23
ObjectType = Group
NDims = 3
ID = 23
Color = 1 0 0 1
TransformMatrix = 1 0 0 0 1 0 0 0 1
Offset = 0 0 0
CenterOfRotation = 0 0 0
ElementSpacing = 1 1 1
EndGroup =
ObjectType = Ellipse
NDims = 3
Name = Ellipse 1
ID = 1
ParentID = 23
TransformMatrix = 1 0 0 0 1 0 0 0 1
Offset = 1 1 1
CenterOfRotation = 0 0 0
ElementSpacing = 1 1 1
Radius = 0.5 0.5 0.5
ObjectType = Ellipse
```

fig. 3.1-12 Definició d'un spatialObject contingut arxiu .meta

En la imatge es pot veure com el fitxer guarda el mínim d'informació sobre cada objecte per poder-lo tornar a crear al llegir-lo. En l'exemple es guarda un objecte SpatialObjectScene, que de 3 dimensions i que conté 23 SpatialObjectes. El primer és un SpatialObjectGroup del que penjarem tots els objectes creats. A partir d'aquest vindrien tots els el·lipses amb els valors de posició, radi, rotació i matriu de translació.

### 3.1.2 Diagrames generació model sintètic

A continuació podem veure el diagrama complet de casos d'ús de generació de model sintètic.

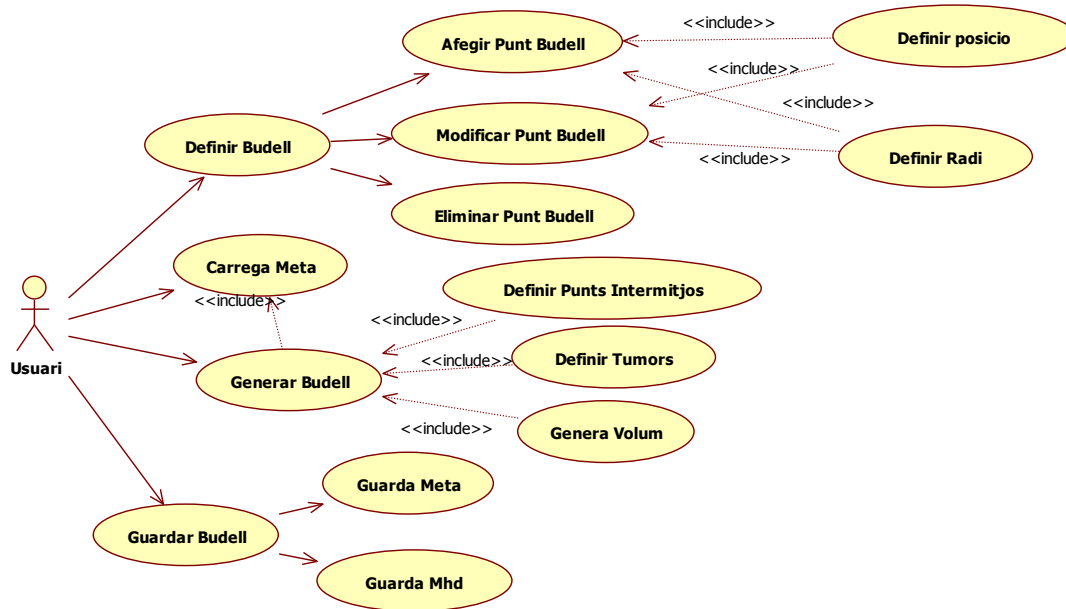


fig. 3.1-13 Diagrama complet de casos d'ús de generació de model sintètic

Descrivim les accions que es podran portar a terme amb aquest mòdul:

- » **Definir budell**
  - Afegir Punt
  - Modificar Punt
  - Eliminar Punt
- » **Generar budell**
  - Definir punts intermedis
  - Definir tumors
  - Generar el Volum
- » **Guardar/Carrega del budell**
  - Guardar en format .meta
  - Guardar en format .mhd
  - Carregar arxiu .meta

### ***Definició del budell***

---

Mostrem el diagrama de casos d'ús per l'apartat de definició del budell:

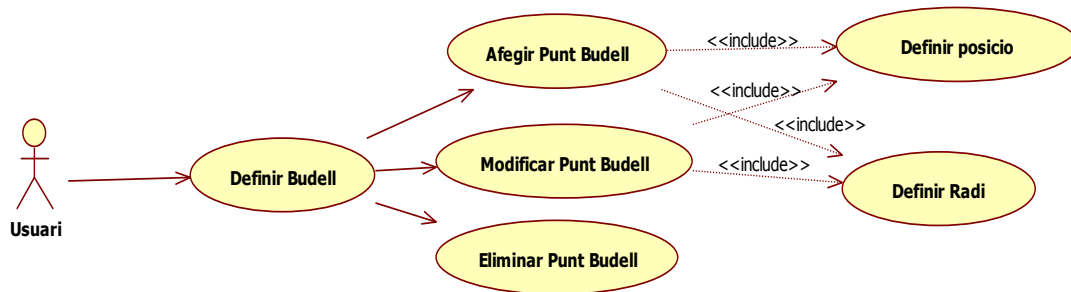


fig. 3.1-14 Diagrama cas d'us Definir Budell

Aquesta funcionalitat permetrà a l'usuari definir els punts característics del budell a partir dels quals es generarà el budell.

L'usuari podrà interactuar amb la interfície fins a definir el tub desitjat. Podent generar, modificar i eliminar punts de l'estructura. Per cada punt es definiran els valors de posició i el del radi.

**Definir budell**

**- Fitxa de cas d'ús:**

Cas d'ús	Definir Budell
Descripció	Afegeix un Punt a la llista de punts del tub
Actors	Usuari
Precondició	
Flux Principal	1.- L'usuari defineix un punt que definirà la trajectòria. 2.- Afegeix el punt del budell.
Fluxos Alternatius	1.- L'usuari tria el punt a modificar. 2.- El modificarà els valors del punt. 3.- En guarda els valors resultants.
Fluxos Alternatius	1.- L'usuari tria el punt a eliminar 2.- Eliminar punt del budell:
Postcondició	S'ha afegit un nou punt i s'ha incrementat el nombre de punts

Per definir el budell l'aplicació proporciona els funcionalitats d'afegir, modificar i eliminar punt de la llista de punts de l'estructura.

**- Diagrama de seqüència:**

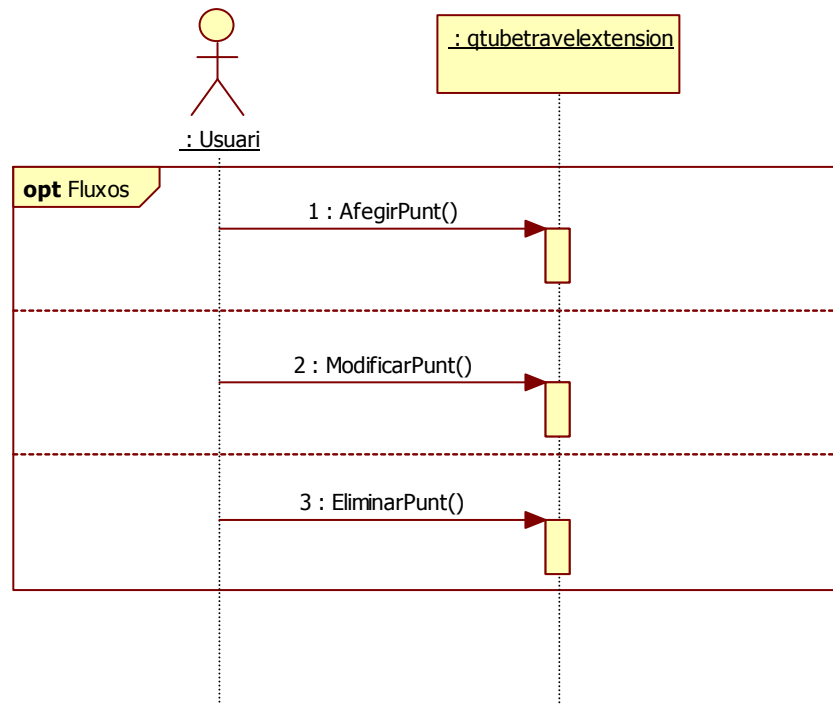


fig. 3.1-15 Diagrama de seqüència "Definir Budell"

**Afegir punt al budell:**

- **Fitxa de cas d'ús:**

Cas d'ús	Afegir punt al budell
Descripció	Afegeix un Punt a la llista de punts del tub
Actors	Sistema
Precondició	- Dades entrades per l'usuari
Flux Principal	1.-Definir components 2.-Definir radi 3.-Afegir nou punt a la llista de punts
Fluxos Alternatius	
Postcondició	S'ha afegit un nou punt i s'ha incrementat el nombre de punts

"Afegir punt el budell" interacciona amb la llista de punts, afegint nous punts al final de la llista que va generant l'usuari.

- **Diagrama de seqüència:**

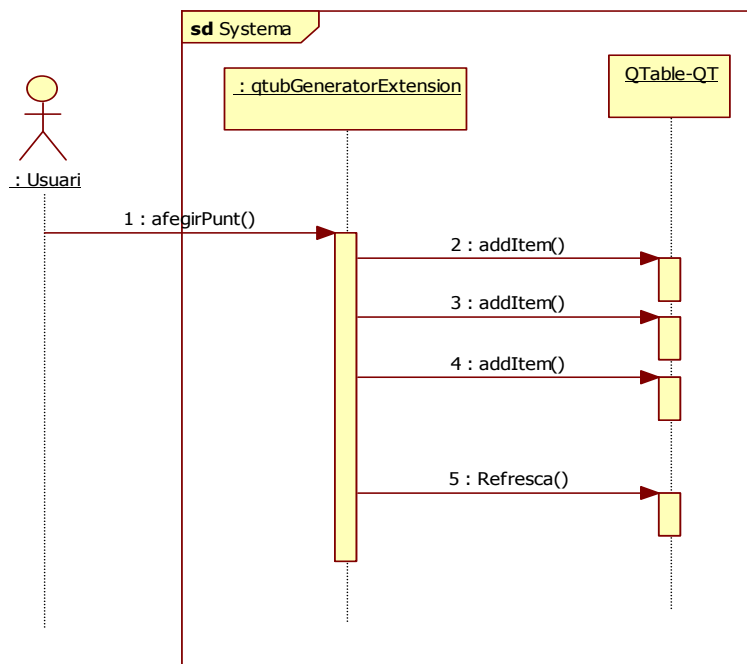


fig. 3.1-16 Diagrama seqüència afegir punt al budell

**Modificar punt del budell:**

- **Fitxa de cas d'ús:**

Cas d'ús	Modificar Punt Budell
Descripció	Modificar un punt de la llista
Actors	Sistema
Precondició	Hi ha d'haver un punt introduït com a mínim
Flux Principal	1- Selecció punt a modificar 2- Editar valors del punt 3- Guardar modificacions
Fluxos Alternatius	1- Si no hi ha cap punt, mostra missatge
Postcondició	S'ha modificat un punt de la llista

Modificar punt permet a l'usuari modificar els paràmetres dels punts que ja hagi entrat, i formin part de la llista.

- **Diagrama de seqüència:**

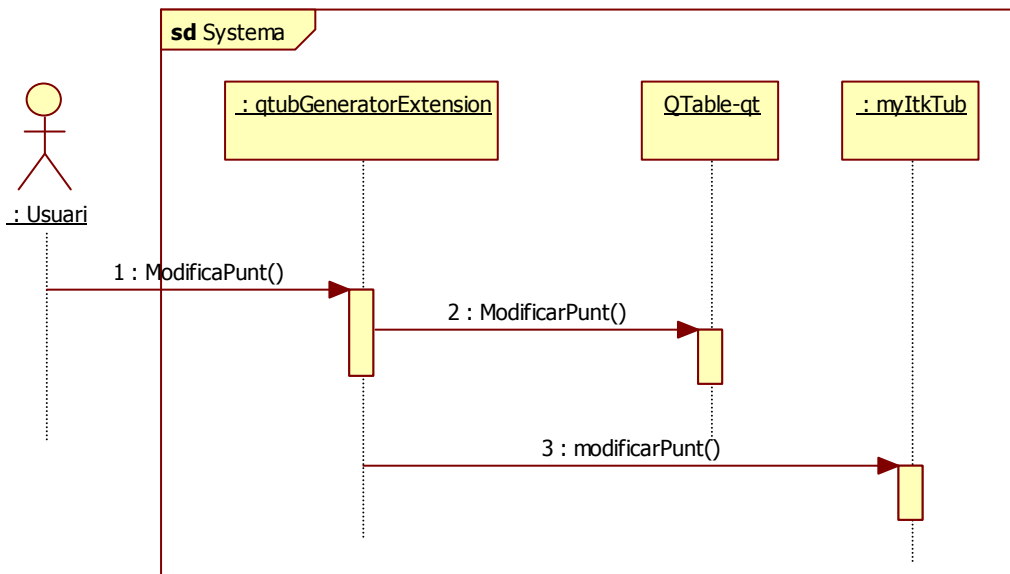


fig. 3.1-17 Diagrama seqüència "Modificar punt"

### Eliminar punt budell

- **Fitxa de cas d'ús:**

Cas d'ús	Eliminar Punt Budell
Descripció	Modificar un punt de la llista
Actors	Sistema
Precondició	Hi ha d'haver un punt introduït com a mínim
Flux Principal	1- Selecció punt a modificar 2- Eliminar el punt de la llista, si no té cap seleccionat elimina últim
Fluxos Alternatius	1- Si no hi ha cap punt, mostra missatge
Postcondició	S'ha eliminat un punt de la llista

"Eliminar punt" permet al usuari eliminar l'últim punt afegit a la llista de punts.

- **Diagrama de seqüència:**

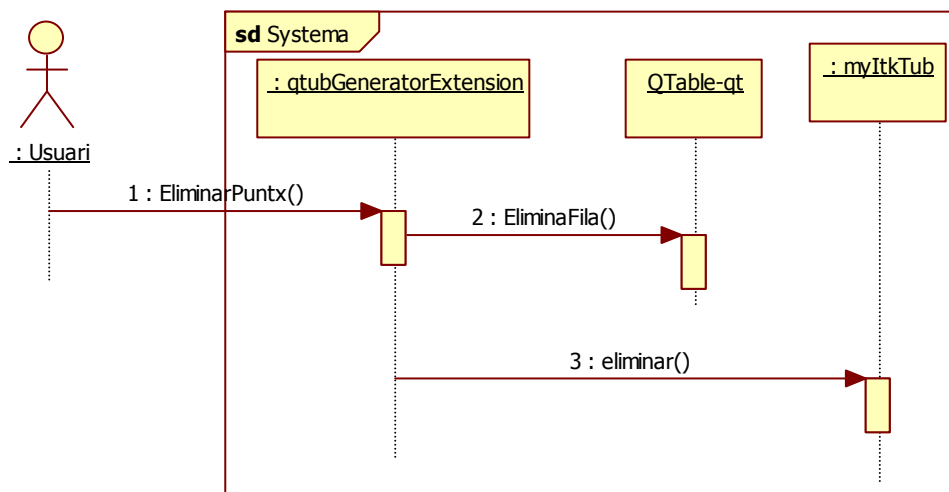


fig. 3.1-18 Diagrama de seqüència "Eliminar Punt Budell"



### ***Generar el budell a partir de les dades introduïdes***

A continuació mostrem el diagrama de casos d'ús de la generació del budell:

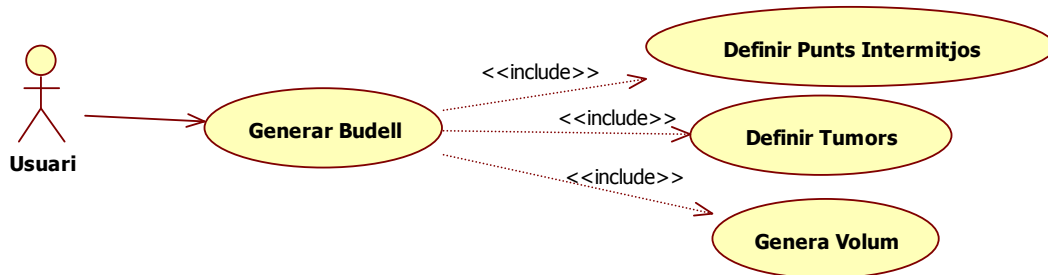


fig. 3.1-19 Diagrama de casos d'ús generació del budell

Aquesta funcionalitat ens servirà per construir l'estructura a partir de les dades entrades per l'usuari. Un cop s'hagi entrat tota la informació referent al budell sintètic es podrà generar l'estructura del budell.

Es seguiran els passos pel següent ordre, es definiran els punts intermedis per definir completament l'estructura del budell, ja que l'usuari només entre alguns punts de l'estructura, no tots. Es definiran els tumors a mida que es vagin creant les parts de l'estructura sintètica. Un cop finalitzat el procés de creació, es convertirà l'estructura en un volum de dades tridimensional, que podrem tractar de la mateixa forma que tractaríem un volum real.

**Generar Budell:**

- **Fitxa de cas d'ús:**

Cas d'ús	Generar tub
Descripció	Generar tub a partir dels punts de la llista
Actors	Sistema
Precondició	Hi ha d'haver dos punts introduït com a mínim
Flux Principal	1.- Crear tub 1.1.- Afegir punt 1.2.- Definir punts intermedis 3.- Generar Imatge a partir de l'estructura del tub generat. 4.- Assignar Imatge als Visors i Refrescar
Fluxos Alternatius	3.1- Si hi ha algun error al generar el tub, mostra missatge d'error
Fluxos Alternatius	4.1- Si hi ha algun error al generar la imatge a partir del SpatialObject mostra missatge d'error
Postcondició	S'ha generat el tub, i una imatge que representa el volum del tub

A partir dels punts entrats per l'usuari es genera el tub complet. S'aniran afegint els punts definits per l'usuari un a un a l'estructura del budell sintètic (1.1) definint entre el nou punt i l'anterior els punts intermedis (1.2). Un cop s'hagi definit l'estructura completa es generarà la imatge (3) i finalment la mostrarem per els visors de l'extensió del mòdul (4).

- **Diagrama de seqüència:**

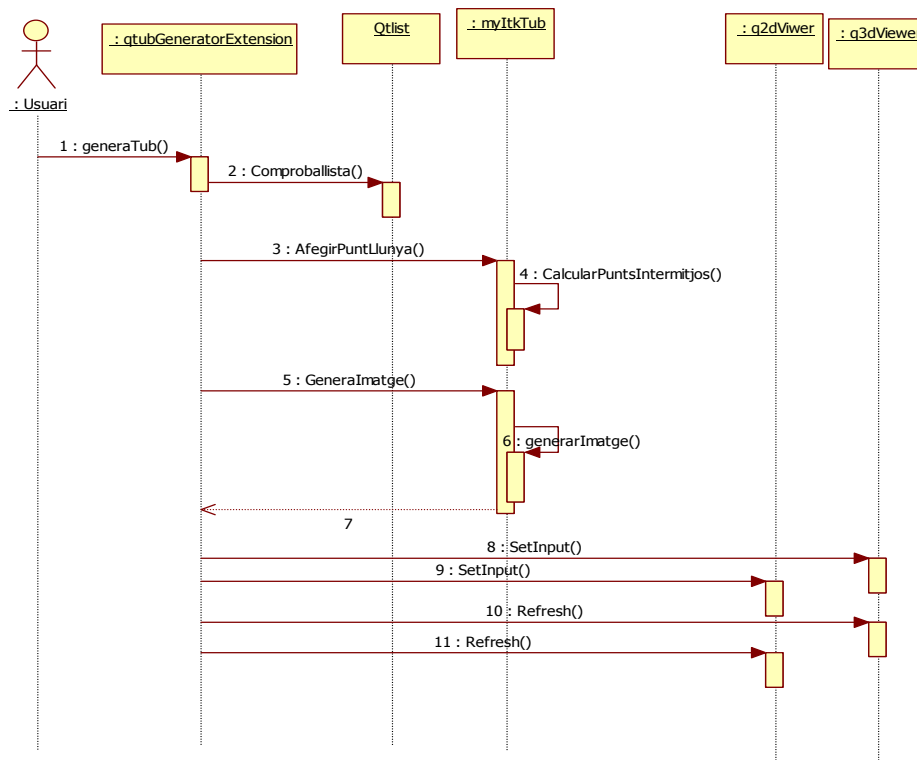


fig. 3.1-20 Diagrama de seqüència "Generar Budell"

### Definir Punts Intermèdies

---

**- Fitxa de cas d'ús:**

Cas d'ús	Definir Punts Intermedis
Descripció	Generar els punts intermedis entre dos punt entrats per l'usuari
Actors	Sistema
Precondició	Hi ha d'haver un punt anterior al actual
Flux Principal	1.- Calcular Distància 2.- Calcular n punts intermedis 3.- Calcular increments per cada component 4.- (n)Per cada n punts intermedis 4.1.- Generar Punt 4.2.- Afegir Punt
Fluxos Alternatius	En cas que s'hagin de crear tumors, el bucle de generar punts es modifica. 4.- Generar Punt 4.1.- Definir Tumor 4.2.- Afegir Punt
Postcondició	S'han definit i afegit els punts entremitjos entre tots els punts entrats per l'usuari.

Per generar els punts intermèdies entre dos punts es calcularà la distància entre ells dos (1). A partir de la distància es podrà trobar els nombre de punts que haurem de generar per connectar els punts separats (2). El nombre de punts es permetrà calcular els increments que haurem d'aplicar sobre els paràmetres de cada nou punt (3). Amb els increments trobats anirem generant nous punts i afegint-los a l'estructura (4, 4.1 i 4.2). En el cas que es pugui crearem els tumors a partir del punt generat (4.1 bis). Un cop finalitzat el procés tindrem l'estructura generada connectant els punts entrats per l'usuari i amb tumors si fos el cas.

- **Diagrama de seqüència:**

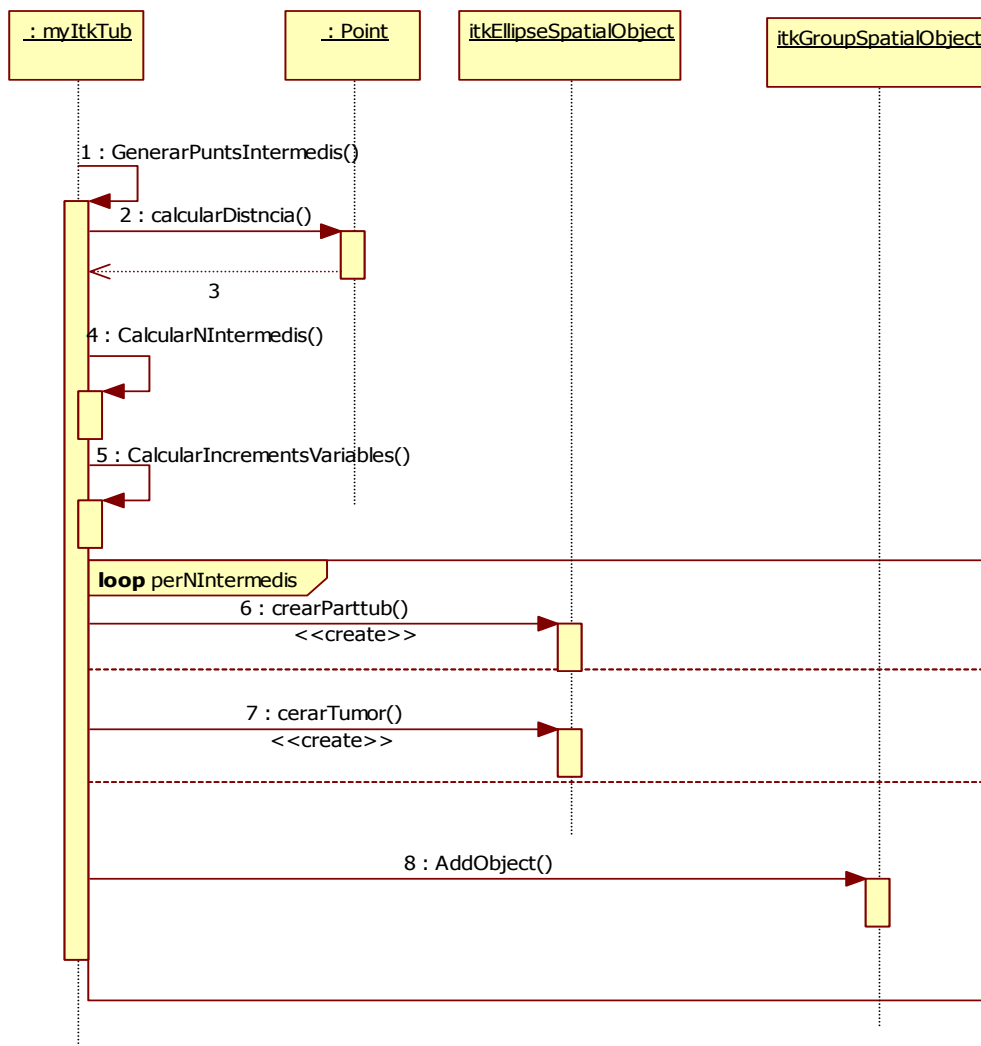


fig. 3.1-21 Diagrama seqüència "Definir Punts Intermedis"

**Definir Tumors**

**- Fitxa de cas d'ús:**

Cas d'ús	Definir Tumor
Descripció	Definir un tumor del budell
Actors	Sistema
Precondició	Esfera, que conté el tumor, generada
Flux Principal	1.- Calcular posició 2.- Calcular Radi del tumor 3.- Generar Tumor 3.1.- Generar El·lipse 3.2.- Assignar radi 3.3.- Assignar posició 4.- Afegir tumor a l'estructura
Postcondició	S'ha definit un tumor del budell

Al definir un tumor, primer de tot, calcularem la posició d'aquest respecte el centre de l'el·lipse que el conté(1). Després calcularem el radi que tindrà el tumor (2). Crearem el tumor assignant els paràmetres que el defineixen (3), i l'afegirem a l'estructura del model sintètic.

**- Diagrama de seqüència:**

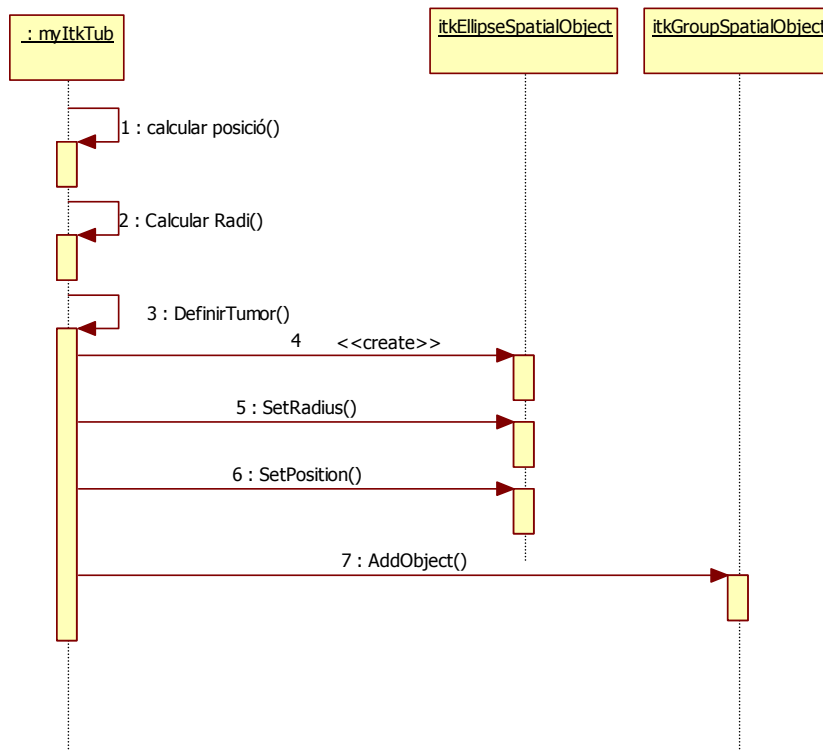


fig. 3.1-22 Diagrama seqüència "Definir tumor"

*Generar Volum*

- **Fitxa de cas d'ús:**

Cas d'ús	Generar Volum
Descripció	Generar el volum a partir de l'estructura del budell
Actors	Sistema
Precondició	Estructura del budell completament creada.
Flux Principal	1.- Generar Objecte SpatialObjectToImageFilter 2.- Assignar l'estructura del budell 3.- Assignar valors de dins i fora l'estructura 4.- Obtenir els límits de l'estructura del budell 5.- Assignar spacing 6.- Assignar mida 7.- Generar Imatge 8.- Obtenir la imatge
Fluxos Alternatius	4- Si es dona algun error a l'hora de convertir l'estructura en imatge, para el procés i mostra error.
Postcondició	S'ha generat una imatge que conté el volum definit per l'estructura creada.

Amb l'estructura definida podrem generar un volum. Per fer-ho aplicarem el filtre spatialObjectToImageFilter. Haurem de definir un objecte de la classe (1) i passar-li els paràmetres i objectes que necessiti per l'execució (2 a 6 ). Finalment executarem el filtre de itk i obtindrem la imatge binària resultant (7 i 8).

- **Diagrama seqüència:**

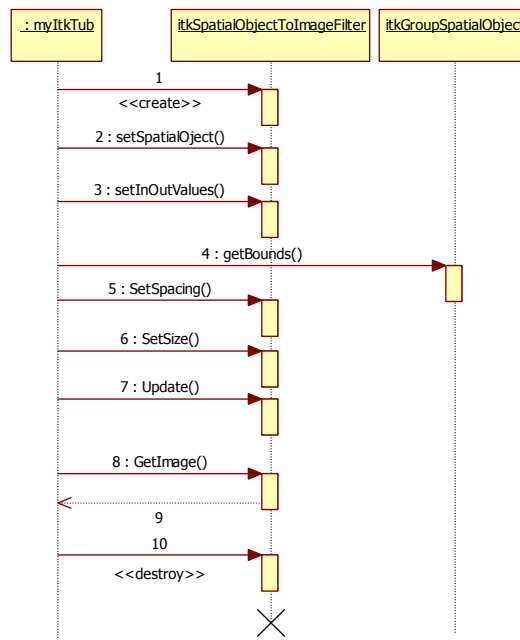


fig. 3.1-23 Diagrama de seqüència "Generar Volum"

### ***Guardar/Carregar el model sintètic***

---

Casos d'ús definits per la funcionalitat de guardar el volum generat:

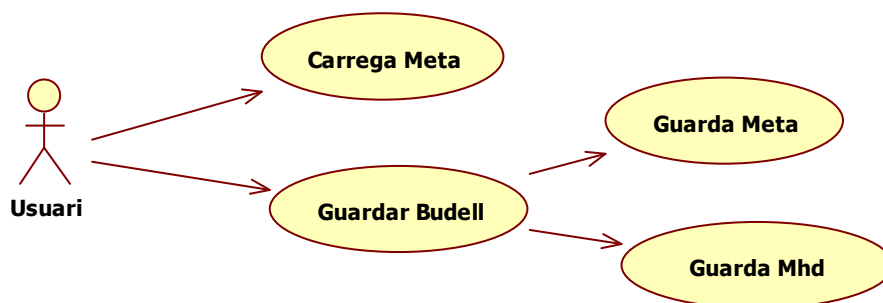


fig. 3.1-24 Cas d'ús "Guardar/Carregar el model sintètic"

Aquest apartat permet guardar el volum generat en diferents formats. En un format .mhd que simula els fitxers DICOM a partir dels dispositius mèdics. O en un format de fitxer pla que guarda l'estructura generada, en aquest cas per obtenir el volum s'haurà de tornar a generar a partir de la informació guardada en el fitxer.

### Guardar Budell

- **Fitxa de cas d'ús:**

Cas d'ús	Guardar Budell
Descripció	Generar el volum a partir de l'estructura del budell
Actors	Sistema
Precondició	Estructura del budell completament generada.
Flux Principal	1.- Guardar budell arxiu meta
Fluxos Alternatius	2.- Guardar budell arxiu mhd
Postcondició	S'ha guardat una imatge que conte el volum definit per l'estructura creada.

La interfície d'usuari permet guardar el model generat amb dos formats, com a volum mhd o com a fitxer pla amb la definició de l'estructura generada.

- **Diagrama de seqüència:**

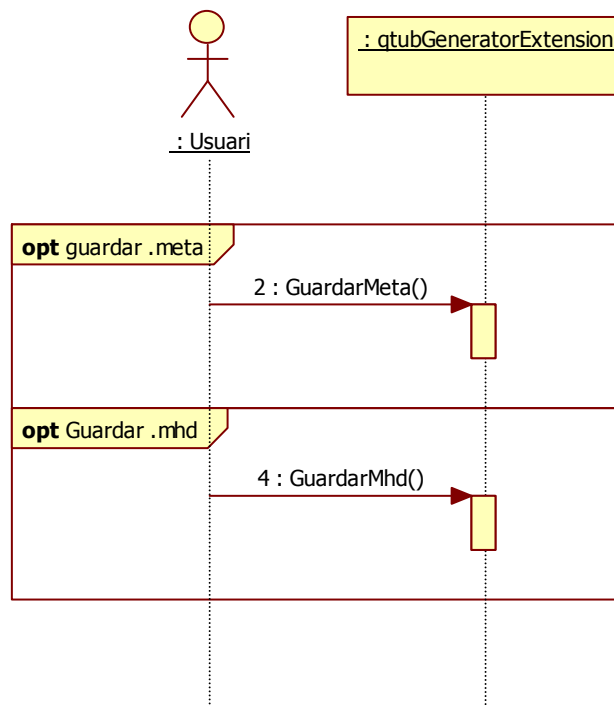


fig. 3.1-25 Diagrama seqüència "Guardar Budell"



**Guardar fitxer META:**

- **Fitxa cas d'ús:**

Cas d'ús	Guardar fitxer .meta
Descripció	Guardar la informació del tub, com a descripció en fitxers .meta. (format que es genera a partir d'un SpatialObject)
Actors	Sistema
Precondició	S'ha d'haver generat el model Sintètic
Flux Principal	1-Obtenir itkSpatialObject de myItkTub 2-Obrir FileDialog (assignar nom fitxer) 3-Guardar
Fluxos Alternatius	1- Si hi hagués un error al guardar la descripció del tub, mostra missatge d'error
Postcondició	S'ha guardat en un fitxer

Per guardar l'estructura del budell simplement crearem un spatialobjectwriter i li assignarem el grup d'objectes, que defineixen l'estructura, com a paràmetres (1) . L'usuari passarà a la classe escriptor el nom del fitxer sobre el que vol guardar l'estructura (2). S'executarà el mètode escriure de la classe escriptor generant el fitxer .meta (3).

- **Diagrama de seqüència:**

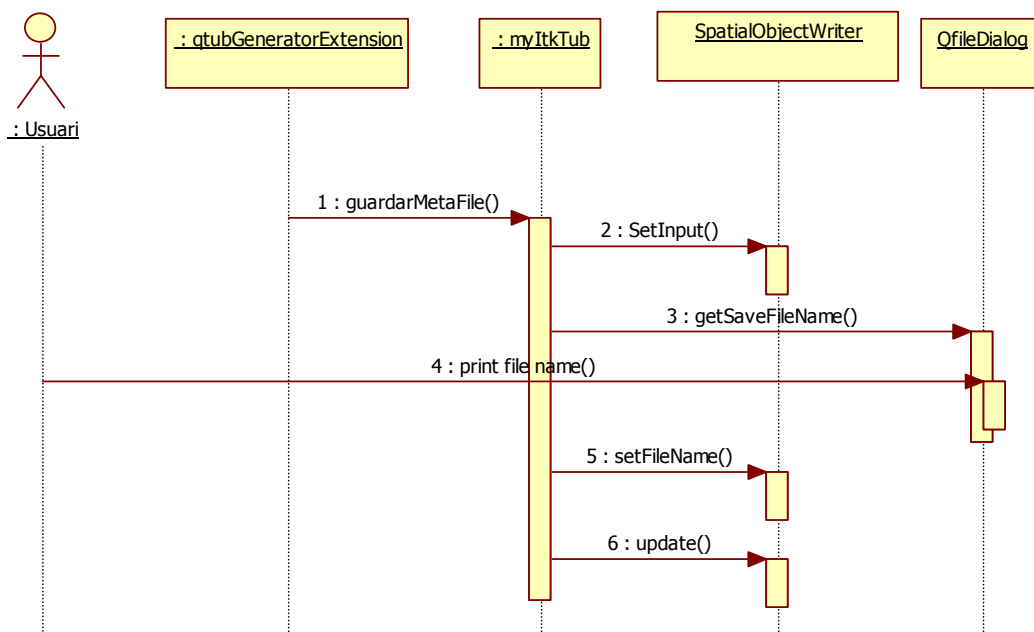


fig. 3.1-26 Diagrama de Seqüència "Guardar fitxer Meta"

### Guardar Tub MHD

- **Fitxa cas d'ús:**

Cas d'ús	Carregar fitxer .mhd
Descripció	Guardar la informació del tub, com a descripció en fitxers .meta. (format que es genera a partir d'un SpatialObject)
Actors	Sistema
Precondició	S'ha d'haver generat el model Sintètic
Flux Principal	1-Obtenir la ImatgeItk/Vtk 2-Crear Objecte Output 3-Assignar Imatge al Output 4-Obrir FileDialog (assignar nom fitxer) 5-Guardar
Fluxos Alternatius	1- Si hi hagués un error al guardar la descripció del tub, mostra missatge d'error
Postcondició	S'ha guardat en un fitxer

Per guardar el volum en format DICOM ho farem mitjançant la classe Output que està especialment dissenyada per aquest propòsit. Passarem la imatge a guardar i un nom pel fitxer que es vulgui guardar. Finalment obtindrem el fitxer .mhd que defineix el volum i un fitxer raw amb les metadades complint amb el format DICOM.

- **Diagrama de seqüència:**

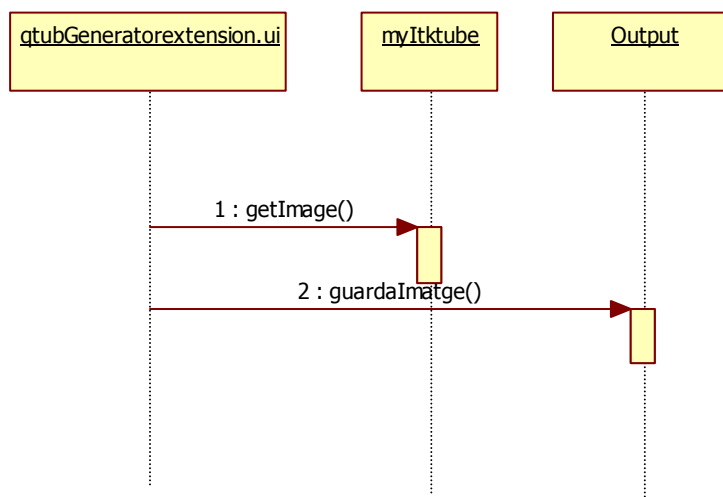


fig. 3.1-27 Diagrama seqüència "Guarda fitxer mhd"

### Carrega fitxer meta

- **Fitxa cas d'ús:**

Cas d'ús	Guardar tub .meta
Descripció	Carrega un tub a partir d'un fitxer .meta (format que es genera a partir d'un SpatialObject)
Actors	Sistema
Precondició	Existeixi el fitxer de carrega
Flux Principal	1-Obrir openFileDialog 2-Seleccionar Fitxer a Carregar 2.1-Carregar Objecte itkSpatial del fitxer 2.2-Generar Volum 3-Aplicar Contingut Filter al volum generat i mostrar
Fluxos Alternatius	1- Si hi hagués un error al carregar l'arxiu del tub, mostra missatge d'error
Postcondició	S'ha generat un tub, amb la informació del fitxer .meta

A partir d'un fitxer .meta podem carrega el contingut generant una estructura de spatialobjectes (2.1). I generarem el volum a partir de l'estructura que conté el fitxer (2.2). Finalment es mostrar el volum generat per als visors de l'extensió (3).

- **Diagrama de seqüència:**

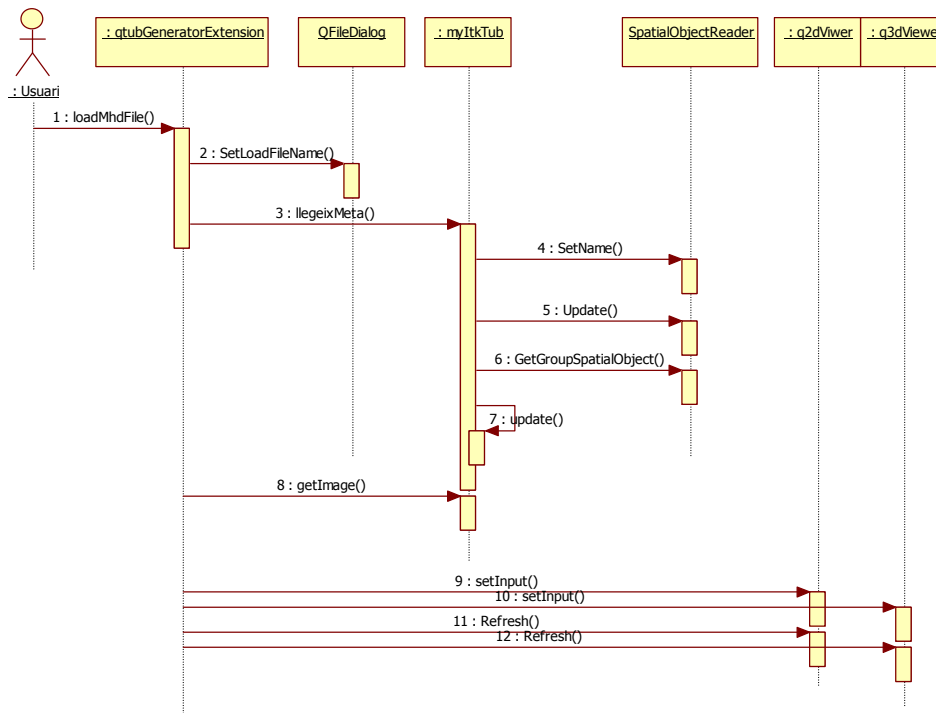


fig. 3.1-28 Diagrama de seqüència "Carregar fitxer .meta"

**3.1.3 Classes del mòdul de generació model sintètic**

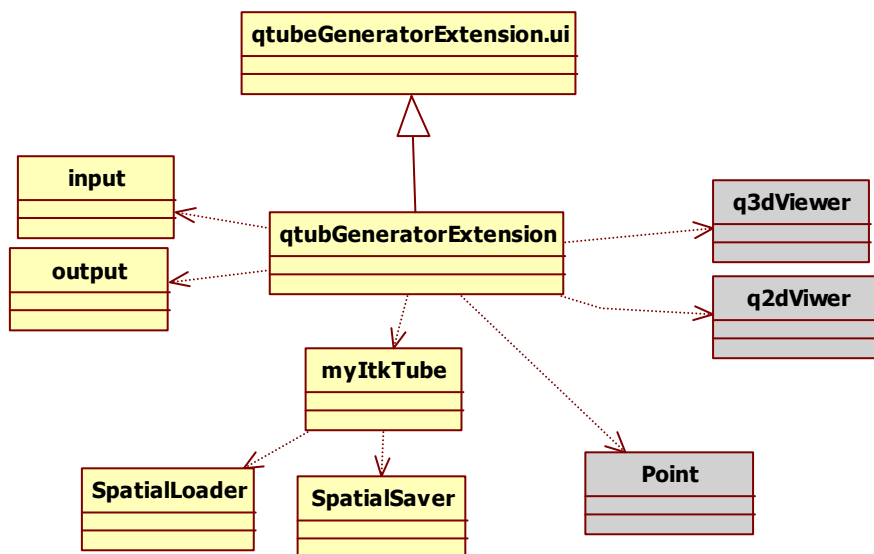


fig. 3.1-29 Diagrama de classes que intervenen en la generació del model sintètic

Per dotar l'aplicació de les funcionalitats del mòdul descrit s'han implementat les classes que es mostren en el diagrama de classes anterior. Les classes de color gris ja estaven implementades en l'aplicació però s'han hagut de modificar o ampliar degut als requeriments del mòdul.

### Myitktube

Aquesta classe conté els algorismes i mètodes de generació de models sintètics. Gestiona la creació d'un model sintètic, afegint i traien punts que definiran la trajectòria final. Crear un volum a partir d'una trajectòria definida en spatialobjects.

Alguns mètodes d'interès:

- **afegirPuntLlunya( x,y,z,radi ) :**

Genera els punts entremitjos entre l'últim punt que forma part de la trajectòria i el punt i el radi entrats. Cridant el mètode generarInterPoints( punt , punt )

- **generarTub():**

Un cop s'han definit tots els el·lipses es genera el tub convertint la informació estructural en objetes SpatialObject, i finalment generant la imatge a partir d'aquests spatialobjects obtinguts.

- **generalInterPoints( punt, punt ):**

A partir dels punts entrats i del nou radi es generaran els punts intermèdies per connectar els dos punts. Primer calcula els el·lipses que caben entre el dos punts, tinguent en compte l'augment o disminució del radi respecte l'últim punt. Es van generant els el·lipses variant el radi en cas que fos necessari.

### *SpatialSaver*

---

Guardar estructura model sintètic en un arxiu .meta guardant informació estructural.

Alguns mètodes d'interès:

- ***GuardaSpatialObject(string nomfitxer):***

Genera un fitxer amb la informació estructural del spatialobject que té la classe.

### *Spatialloader*

---

Carregar estructura model sintètic d'un arxiu .meta que conté la informació d'un spatialobject.

Alguns mètodes d'interès:

- ***LlegeixSpatialObject(string nomfitxer):***

Llegeix la informació del fitxer entrat i reconstrueix l'estructura generant un spatialObject.

### *Modificació classe Point*

---

Representació d'un punt 3D.

Mètodes afegits:

- ***Operator ==(Point, Point)***

Retorna boolea, amb valor cert, indicant si els punts tenen el mateix valor en les components, en cas contrari retorna falç.

- ***Operator != (Point, Point)***

Retorna un boolea, amb valor cert, indicant si els punts no tenen el mateix valor en les components, en cas contrari retorna falç.

- ***Distance( Point p) :***

Retorna la distància entre dos punts.

### *Modificació classe 3DViewer*

---

Visualitzador 3D amb diferents tècniques de renderitzat i visualització.

Mètodes afegits:

- ***renderConturing()***:

A partir del volum principal fa una reconstrucció/visualització del volum pintat només el contorn d'aquest.

### *qtubeGeneratorExtension*

---

Aquesta classe connecta la interfície del mòdul amb les classes desenvolupades. S'han connectat les parts de la interfície, amb la tècnica de signal & slot, amb diferents mètodes que interactuen amb les classes del mòdul.

### *qtubeGeneratorExtension.ui*

---

Es la interfície desenvolupada amb el qtDesigner. Conté un visor 3d per veure el volum renderitzat. Un altre visor 2D per veure les seccions del volum generat. I un conjunt de widgets (spinbox, slicers, botons i textbox) perquè l'usuari interactuï amb l'extensió.

### 3.1.4 Resultats i testeig

Mostrem alguns exemples resultants de tubs generats amb el generador implementat:

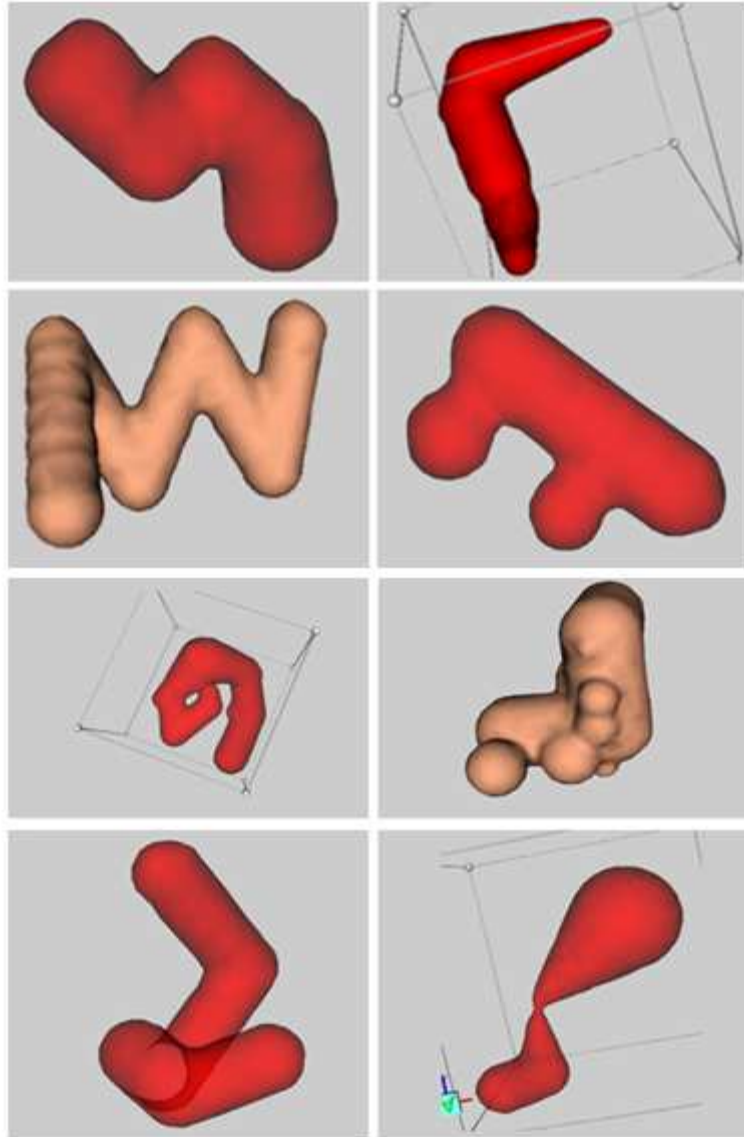


fig. 3.1-30 Diferents tubs generats a partir del mòdul implementat

Per guardar els models també els podem guardar en arxius amb metadades de l'estructura que defineix el budell. Al annex hem afegit una arxíu .meta complet per veure el format dels tipus d'arxius generats.

En la generació dels models sintètics el testeig s'ha fet sobre la visualització resultant, s'ha comprovat que el volum tingues la forma desitjada. Sabent els punts entrats podem valorar el resultat generat.

## 3.2 AVALUAR LES TÈCNiques DE PROCESSAMENT D'IMATGE QUE CALEN PER DIAGNOSTICAR ALGUNES DE LES LESIONS DE BUDELL.

L'objectiu del mòdul dissenyat per la segmentació i visualització de lesions de colon és facilitar el diagnòstic del càncer colorectal. Les funcionalitats principals seran la segmentació i la visualització. La segmentació ens permetrà aïllar la zona patològica de la tota la informació del volum. Mentre que l'objectiu de la visualització és la representació en pantalla del budell i les zones lesionades.

El mètode de segmentació ens ha de permetre obtenir dues zones d'interès del volum, el budell i els tumors. Un cop haguem separat les dues parts haurem d'implementar un mètode de visualització que satisfaci els requeriments del mòdul.

Amb tot això els objectius d'aquest apartat son la segmentació del volum, i la visualització d'aquest, de forma que permetin visualitzar els tumors i el contorn del budell simultàniament.

Per aquest mòdul s'ha reutilitzat l'estudi i implementació prèvies que han portat a terme el grup de programadors de l'Starviewer. A partir del mètodes escollits es farà un estudi i s'aplicarà sobre els models sintètics per comprovar el resultat, sobre els volum generat, és correcte.

Tot i que el mètode de segmentació no ha estat implementat per nosaltres si que ens ha calgut estudiar-lo per poder entendre el seu funcionament i aplicar-lo a les nostres dades.

### 3.2.1 Tècniques de segmentació

Entenem per segmentació del budell, sintètic o real, la separació dels vòxels que corresponen a les parets del colon. Es a dir, volem aïllar aquesta informació de la resta d'informació representada en model de vòxels.

Hi ha diferents mètodes de segmentació, tot seguit enumerarem alguns dels mes coneguts. Entre els esmentats escollirem el que millor s'adapti a les nostres necessitats.

- **Segmentació segons l' histograma de la imatge**

A partir de les dades dels píxels com intensitat, color o altres es generen uns histogrames. A partir dels màxims i mínims es poden identificar característiques o zones d'interès de la imatge.

- **Segmentació per thresholding (llindars)**

Es defineixen uns llindars sobre característiques de la imatge entre els quals hi ha d'haver la zona que es desitja segmentar per tal de ser acceptada. Amb aquest algoritme obtenim imatges binàries que marquen amb 1 la zona de interès i amb 0 la part que queda fora. Aquesta tècnica ens serà útil per separa les lesions ja que solen tenir un nivell de gris determinat.



- ***Split and Merge***

Consisteix en dividir la imatge en regions uniformes. Igual que la segmentació per histograma es útil per imatges que segueixen regions uniformes. Per el que no ens serveix en el nostre cas.

- ***Region Growing***

Consisteix en escollir un punt inicial arbitrari, anomenat llavor ( "seed" ), a partir del qual s'aniran annexant punts veïns, sempre que compleixin les propietats adequades, incrementant la regió seleccionada.

Dels mètodes descrits optarem per una combinació dels mètodes "Region Growing" al que li haurem de determinar una llavor, i el thresholding que ens permetrà avaluar si una vòxel forma part de la zona a segmentar o no.

### 3.2.2 ***Region Growing més Thresholding***

Conjuntament amb els desenvolupadors de l'Starviewer s'ha decidit que dels mètodes que tenien implementats els de thresholding i region growing són els que s'adapten millor a les nostres necessitats.

Els mètodes escollits necessiten uns determinats paràmetres, que l'usuari pot especificar, però se'n donen uns per defecte amb els quals es solen obtenir bons resultats. Tot seguit comentarem aquests paràmetres:

- ***Llindars (superior i inferior):***

Valors entre els quals ha d'estar la intensitat de gris del nou píxel que es vol afegir.

- ***Localització de la llavor:***

Punt a partir del qual començarà a créixer la regió que volem segmentar. La selecció de la llavor inicial es farà automàticament cercant un punt de la imatge que compleixi els llindars definits.

El mètode escollit ens permetrà segmentar les parts d'interès que en el nostre cas seran el budell i els tumors. Volem poder-les segmentar per separat aconseguint així poder identificar les dues parts. Farem servir el mètode que utilitzen actualment a la plataforma Starviewer.

La segmentació també ens permetrà considerar-ne el volum del pòlip, entre altres tècniques que ja estan utilitzant l'equip de desenvolupament de l'Starviewer.

Per poder definir les zones a segmentar s'ha d'haver estudiat les formes que interessin de les imatges obtingudes. En la imatge següent veiem com les zones que ens interessin de les imatges son les del budell que son les parts més fosques que es veuen en la figura 3.2-1.

## FASE D'IMPLEMENTACIÓ

avaluar les tècniques de processament d'imatge que calen per diagnosticar algunes de les lesions de budell.

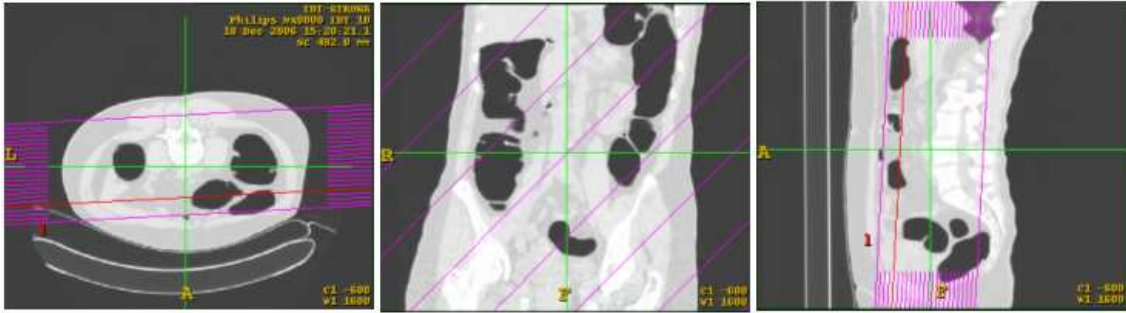


fig. 3.2-1 Seccions de plans sagital axial i coaxial Resultats tac

Com veiem en les imatges, la zona que ens interessa del TAC és la zona més negra. La zona negra és el budell. És una zona buida, ja que per realitzar el TAC inflen el budell amb la finalitat de millorar la visualització del budell al realitzar la prova.

Definint correctament els paràmetres de segmentació es poden obtenir resultats de segmentació que permeten visualitzar volums com el que mostrem a continuació.

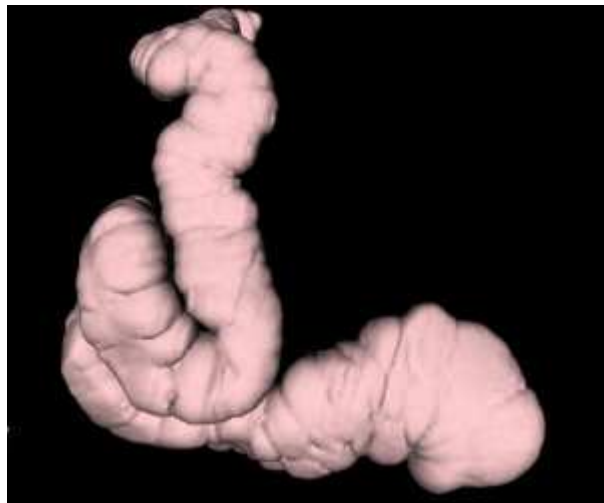


fig. 3.2-2 Segmentació de colon renderitzat en 3D

### 3.2.3 Diagrames segmentació

Diagrames casos d'ús del mòdul de segmentació:

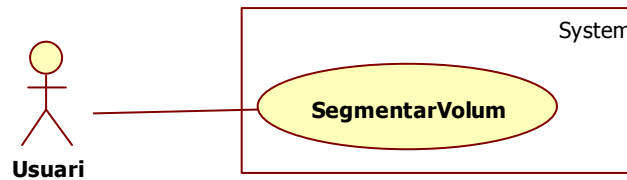


fig. 3.2-3 Diagrama de cas d'ús Segmentació

Aquest mòdul s'encarrega de segmentar la imatge obtinguda, separant les regions que contenen els tumors i el cos del budell utilitzant les tècniques descrites en l'apartat anterior.

#### Segmentar Volum

- **Fitxa cas d'us:**

Cas d'ús	Segmentar volum
Descripció	Segmentar un volum introduint els rangs i nivells de grisos que determinen les zones d'interès.
Actors	Sistema,Usuari
Precondició	S'ha de tenir una imatge generada
Flux Principal	1-Assignar Imatge procés Segmentació 2- Inicia procés de segmentació 2.1- Assignar nivells de gris a zones interès 3- Mostra la imatge per els visors corresponents
Fluxos Alternatius	1- Si hi ha algun error en el procés, mostra missatge d'error
Postcondició	S'ha obtingut una imatge segmentada del volum inicial.

Aquesta classe es l'encarregada de executar el mètode de segmentació. Aquest mètode internament utilitza classes de itk que ajuden segmentar la imatge.

- **Diagrama de seqüència:**

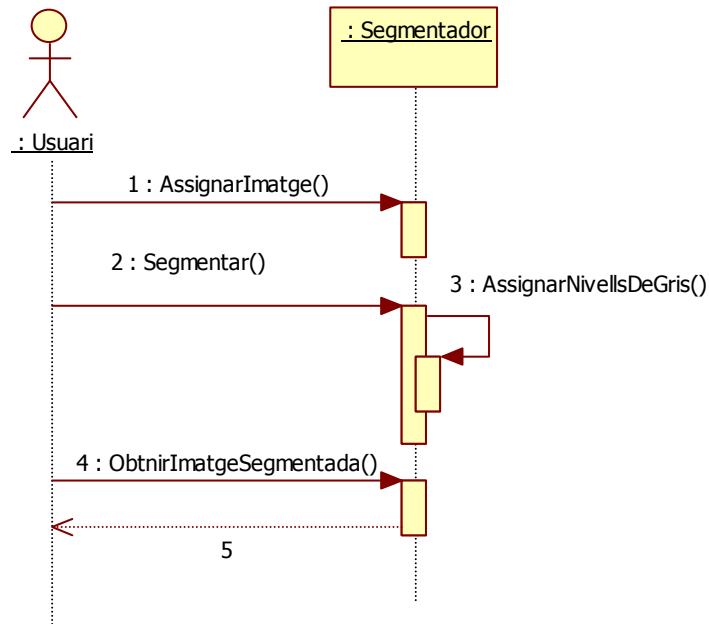


fig. 3.2-4 Diagrama seqüència "Segmentar Volum"

### 3.2.4 Classes del mòdul de segmentació

#### Segmentador

Aquesta classe conté els algorismes i mètodes per segmentar un volum a partir d'una llavor, i seleccionant un rang de nivells de gris de la imatge. La imatge segmentada es retorna com a imatge de sortida.

Alguns mètodes d'interès:

- **Segmentar() :**

Inicialitza el procés de segmentació sobre la imatge. Genera i executa tots els filtres de itk necessaris per obtenir la imatge final.

### 3.2.5 Resultats i testeig

Mostrem un exemple de segmentació realitzada sobre models sintètics. Primer segmentant la zona del budell sola, i en segon cas el budell amb els tumors, i en mostrem una visualització de les dues segmentacions:

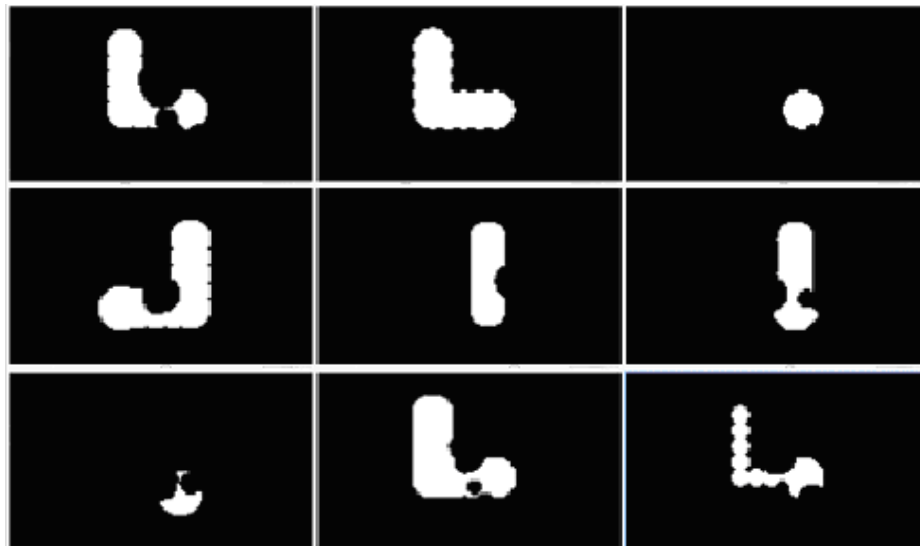


fig. 3.2-5 Segmentació sobre un volum generat per el generador

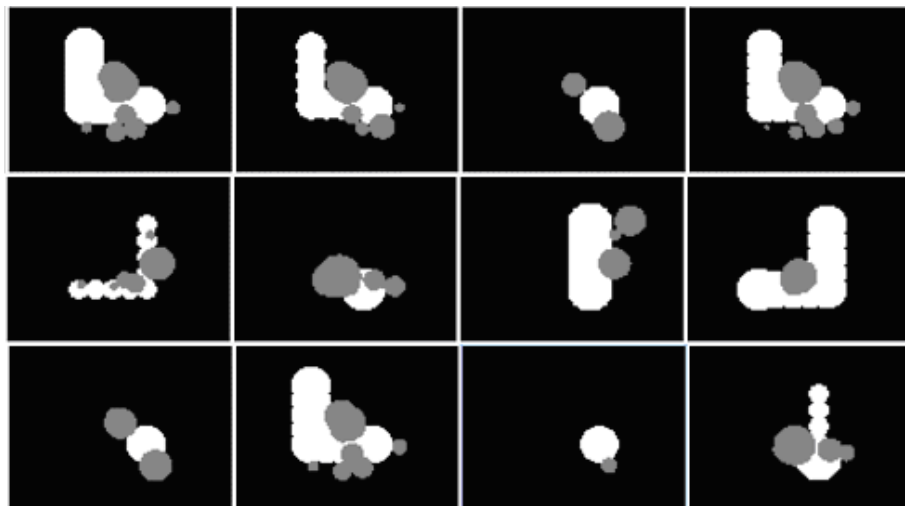


fig. 3.2-6 Segmentació del budell i els tumors sobre un budell sintètic

## FASE D'IMPLEMENTACIÓ

avaluar les tècniques de processament d'imatge que calen per diagnosticar algunes de les lesions de budell.

A continuació mostrem el resultat tridimensional del volum d'entrada de l'exemple i de la segmentació del mateix en els dos casos.



fig. 3.2-7 Visualització sobre el model sintètic, segmentació del model sintètic sense els tumors i visualització de la segmentació conjunta dels tumors i el budell.

Per fer el testeig del procés de segmentació hem validat que el resultat de la segmentació sigui igual que la imatge obtinguda en el procés de generació del model sintètic.

Per tant comparem diferents seccions del volum per diferents plans en les dues imatges, la inicial i la segmentada, validant que siguin iguals.

Mostrem a continuació imatges generades i imatges segmentades en el mateix pla de tall per veure el resultat obtingut.

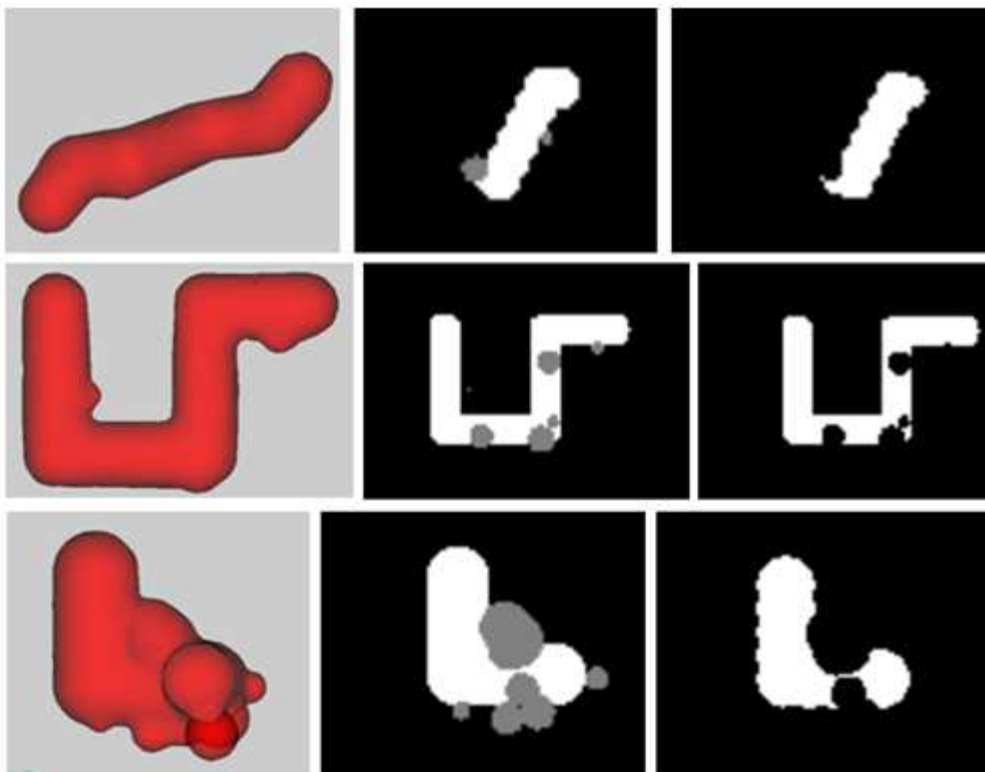


fig. 3.2-8 Resultats de la segmentació comparats amb el model inicial

Amb les imatges anteriors es veu com obtenim en la segmentació uns resultats idèntics a la imatge inicial, que es el que en el nostre cas esperàvem. OK !!!!

### 3.3 DISSENYAR I IMPLEMENTAR UN SISTEMA D'EXPLORACIÓ DEL BUDELL

La distribució de les dades i la morfologia de la zona d'estudi dificulten en aquest cas la visualització de les dades. El que ens proposem és dotar d'un mòdul que permeti fer una exploració més còmode d'una reconstrucció tridimensional d'un budell. Serà una tècnica simple que serà una primera aproximació a una colonoscòpia virtual.

Per entendre millor la dificultat del problema mostrem a continuació un parell d'imatges que poden il·lustrar amb més claredat el problema en concret.

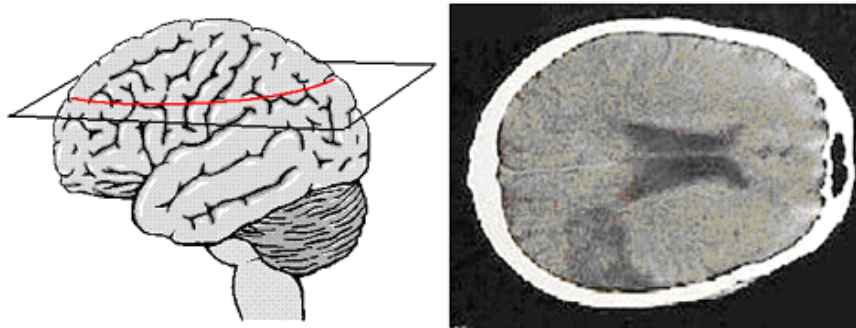


fig. 3.3-1 Simulació pla de tall de la secció d'un cervell

En la primera imatge podem observar que un TAC craneal, amb el qual es pot veure el cervell, obté un fitxer DICOM que conté el cap o part del cap, però aquesta és una zona en que trobem molt ben definit el contingut ja que el cervell es troba delimitat i definit pels límits del fitxer DICOM.

D'altra banda, el TAC pèlvic a més de contenir altres òrgans es troba en una zona en que el soroll dins de les dades augmenta ja que son zones on els òrgans els afecte un canvi o que contenen variacions en l'interior. En el cas del budell, el contingut de gas o excrements afegeix un punt de dificultat en la segmentació. La imatge següent mostra una secció d'un volum de TAC pèlvic, on hi podem veure el soroll contingut i la complexitat morfològica del tipus de dades.



fig. 3.3-2 Tac pèlvic

D'altra banda es pot veure que la forma del budell no queda definit pels límits del extensió espacial del volum de dades, ja que ens podem trobar casos en que el pla contingui varies seccions del budell, com mostrem a la imatge posterior.

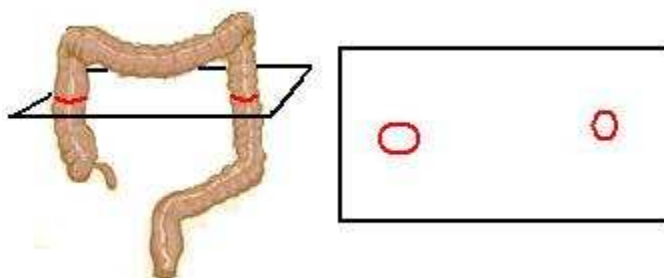


fig. 3.3-3 El volum pot contenir més d'una secció del budell en un mateix pla

L'estratègia que proposem per visualitzar el budell consisteix en segmentar el budell i una vegada el tenim segmentat definir la trajectòria que passa pel seu centre.

### 3.3.1 Tècniques utilitzades en el sistema d'exploració

#### ***Cercar el centre del budell***

---

La primera pregunta que se'ns planteja és com podem identificar el *centerline*. Per marcar les zones que cauen dins el budell haurem de fer dos modificacions de la imatge. La primera i que ja hem explicat en l'apartat anterior és la segmentació de la imatge, i la segona és aplicar un filtre que ens generi un mapa de distàncies sobre la imatge del budell.

#### ***Mapa de distàncies***

---

El mapa de distàncies fa un degradat del nivell de gris dels vòxels que indica si un punt és més lluny o més a prop de la zona determinada per la imatge segmentada. D'aquesta



forma obtindrem una imatge que ens marcarà clarament el centre de l'objecte segmentat, en el nostre cas el budell.

Per clarificar l'explicació, mostrarem un imatge d'una llesca del volum, al qual hem aplicat el filtre de mapa de distàncies. Com podem veure quan més ens aproximem al centre del budell més augmenta el valor de nivell de gris i ens anem acostant al valor 255 en nivell de gris (blanc).



fig. 3.3-4 Mapa de distàncies sobre una secció del budell

En la imatge anterior podem veure una secció del budell sobre una imatge a la que li hem aplicat un mapa de distàncies. Seguint les zones amb un nivell de gris més alt, més pròxim al blanc, trobarem el centre del budell.

Al identificar les zones que cauen dins el budell i els punts més centrals de volum del budell mitjançant el nivell de gris de cada vòxel, podrem determinar els punts que formen part del centre del budell.

Un cop hem segmentat la imatge i li hem aplicat el mapa de distàncies, procedirem a cercar el centre del budell. Primer cerquem els punts que poden formar part del centre del budell i llavors mitjançant tècniques de backtracking farem una fusió dels punts trobats per obtenir la trajectòria final.

### Backtracking

La tècnica de backtracking és una tècnica que utilitza recursivitat per a obtenir les múltiples solucions complexes a un problema. El mètode consisteix a anar reconstruint pas a pas solucions parcials. La complexitat recau en que cada solució parcial pot avançar cap a noves solucions parcials, i cada una d'aquestes a n altres. Fins que arribem a una solució final que satisfà els requeriments del problema, o a una solució que no serà satisfactòria i no tindrà cap opció per avançar a una nova solució. Es en aquest punt final en el que es torna enrere per explorar nous camins.

Gràficament l'estructura d'exploració per les solucions que fa l'algoritme és un arbre de cerca n-ari. Es parteix d'un node inici i a partir d'aquest s'exploren els fills, es a dir totes les seves possibles solucions parcials. Per cada node es fa la mateixa exploració dels fills. Finalment obtindrem una exploració en profunditat d'un graf.

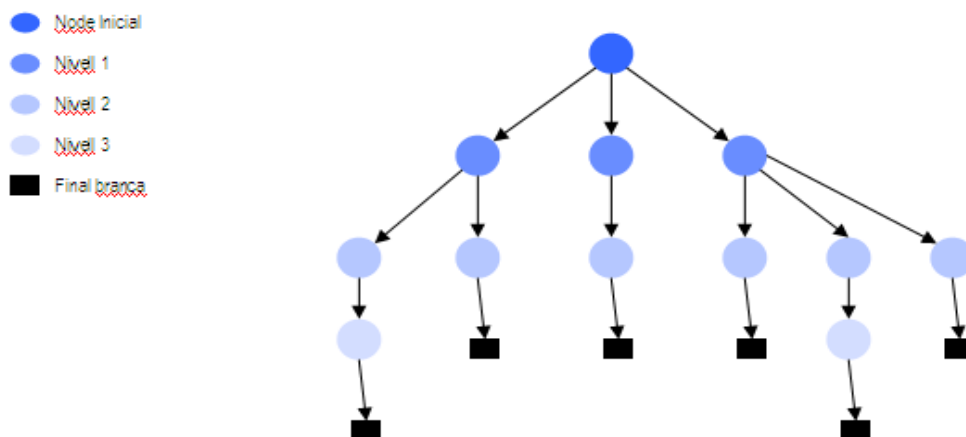


fig. 3.3-5 Esquema generat per un backtracking

L'arbre d'exploració creix molt ràpidament per això es necessiten tècniques de decisió per poder fer podes sobre l'arbre. Aquestes podes es fan mitjançant tècniques heurístiques per decidir quin dels fills de cada node té més possibilitats de formar part d'una solució satisfactòria. Aquestes tècniques són molt dependents del problema que s'intenta resoldre i canvien en cada cas.

Hi ha diferents parts a definir en l'algorisme de backtracking que determinen el desenvolupament del mètode. Seguidament les enumerem:

- Dades inicials: Dades a partir de les quals es formen les solucions.
- Selecció de possibles solucions parcials: Es la forma de determinar quines dades es poden utilitzar per generar una solució parcial a partir de la solució actual.
- Mètode per determinar millor solució: El mètode que s'utilitza per escollir la millor solució de totes les escollides.

En el nostre cas el backtracking farà un recorregut en profunditat del graf que anirà ajuntant les parts de la trajectòria. Les solucions parcials seran les trajectòries formades al afegir un nou punt, que encara no formi part de la trajectòria, a la trajectòria "solució actual". La definició variarà un mica dependent de les dades inicials.

#### MÈTODES DE CERCA DEL CENTERLINE DEL BUDELL

Durant el desenvolupament d'aquesta part del projecte hem realitzat dues fases. Primer s'ha desenvolupat un mètode senzill per obtenir el centre de tubs senzills, seleccionant només un sol centre per cada pla explorat. I en una segona fase s'ha millorat el procés per poder obtenir el centre de tubs més complexes, seleccionant més d'un centre per cada pla.

Tant mateix, els dos mètodes es poden dividir en dues tasques que realitzen per trobar el centre del budell que són:

- **Cercar punts que poden forma parts del centre**

- **Unir els punts per obtenir trajectòries**

La diferència entre els dos mètodes recau en el tipus de dades seleccionades. I degut a les dades obtingudes la forma de connectar les dades per obtenir la trajectòria també variarà.

**CERCA CENTERLINE AMB UN CENTRE PER PLA**

Aquest mètode realitzarà les dues fases comentades anteriorment. En la primera ha de trobar recorrent tot el volum els millors centres per cada pla. En la segona fase unirem els punts amb un mètode de backtracking pensat per les dades obtingudes en la primera fase.

- **Cerca punts que poden forma parts del centre:**

Recorrem la imatge, llesca a llesca, seleccionant els millors punts de cada pla, es a dir els punts que continguin el nivell de gris més alt. Per no perdre cap punt del budell farem el recorregut per la imatge pels plans perpendiculars als eixos de coordenades.

Per veure el funcionament del mètode adjuntem imatges en que es veuen els centres trobats per els 3 escombrats x, y i z.

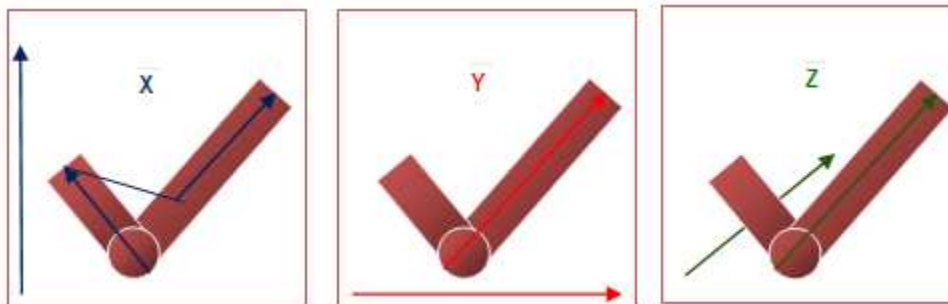


fig. 3.3-6 Subtrajectòries obtingudes a partir del recorregut per els eixos x, y i z

En la primera imatge es pot veure que el centre trobat per la trajectòria x blava, fa un salt entre un punt on s'acaba la part curta del tub i comença de nou en la següent llesca però a una altre part del tub.

La trajectòria obtinguda pel recorregut sobre l'eix x es pot veure que perd un seguit de punts necessaris per formar la trajectòria, es per aquest motiu que s'obtenen les tres trajectòries.

- **Unir els punts per obtenir trajectòries:**

Un cop trobats els punts pels tres eixos fusionarem els punts trobats. Posarem en comú els punts trobats per les tres trajectòries, eliminant així els punts repetits i aquells punts que estiguin tant pròxims a un altre que els considerem iguals i els eliminem. Escollirem uns possibles punts inicials i a partir d'aquests iniciarem un algoritme de backtracking per trobar el millor centre.

Definim el backtracking:

- **Dades inicials** : Conjunt de punts.
- **Selecció de solució parcial**: S'escullen tots els punts pròxims a l'últim afegit a la trajectòria i que no formin part de la trajectòria actual.
- **Mètode selecció millor solució**: Per validar quin és la millor trajectòria de les resultants ho escollirem per la que retorni més punts.

Per escollir la millor solució s'han testejat i fet algunes proves sobre els models de proves fins que s'han aconseguit una solució que s'aproxima a la solució desitjada. Tot i que en el mètode en concret no acaba de ser una solució completament correcte.

El que pretenem és que fusionant els punts de les trajectòries puguem obtenir una trajectòria que realment recorri el centre. Al fer la fusió de les trajectòries obtindrem una trajectòria que seleccionarà els millors punts que formaran el centerline.

Seguint amb l'exemple hauríem de fer la fusió de les trajectòries. Aconseguint :



fig. 3.3-7 Trajectòries trobades sobre el tub

#### CERCA CENTERLINE AMB MÚLTIPLES CENTRES PER PLA

Igual que el mètode anterior, per trobar el centre es farà en dos passos. El primer serà trobar els punts que poden formar part del centre, tot i que en aquest cas a mida que els seleccionem els anirem ajuntant per proximitat formant subtrajectòries. La segona part consistirà en ajuntar aquestes subtrajectòries per generar un centerline.

- **Cerca punts que poden formar part del centre:**

Per trobar els punts obtindrem un mapa de distàncies del volum a partir d'una imatge segmentada d'aquest. Treballarem sobre el mapa de distàncies per obtenir els punts que podrien ser centres recorrent la imatge per un eix determinat.

Per obtenir els punts recorrerem els volum pels tres eixos obtenint els punts del centre a partir dels recorreguts en amplada, alçada i profunditat.

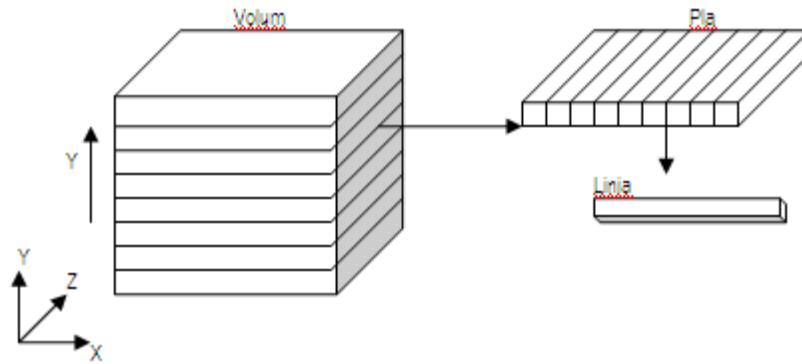



fig. 3.3-8 Recorregut del volum



Per cada pla del recorregut del volum, cercarem els punts del pla que siguin centres. Per buscar els punts del pla que formen part de la trajectòria obtindrem  $n$  punts de cada línia del pla que puguin ser part del centre quedant-nos amb els millors.

Els recorreguts per cada pla del volum segueixen les següents fases per cada centre:

#### **M0 : Fase 0 Cerca del primer punt del tub**

De la llesca actual recorrerem columna a columna fins que trobem un punt  que forma part del tub

#### **M1 : Fase1 Acostant-se al centre del tub**

En aquesta fase els punts que ens retornin les línies seran millors fins que arribem al centre. En el moment en que la línia ens retorna un punt  pitjor que la anterior  , voldrà dir que l'anterior és un centre del pla , ja que entrem en la segona fase i ens estarem allunyant del centre del tub.

#### **M2 : Fase 2 Allunyant-se del centre del tub**

En aquesta fase estem allunyant-nos del centre per tant fins els punts retornats per les línies han de ser pitjors. És a dir, seguim endavant per sortir de la zona del centre trobat, fins que ens acostem a un altre centre que és quan millorarà el valor dels possibles centres retornats per les línies. En aquell moment tornem a entrar en fase1

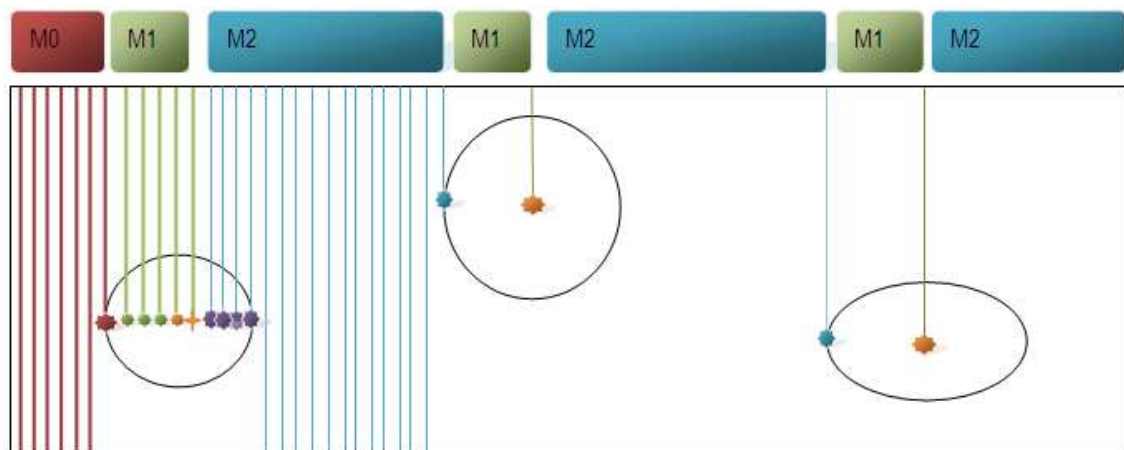


fig. 3.3-9 Recorregut per cercar més d'un centre sobre la llesca actual

Aquest procés de fases es tracta per separat per cada centre del mateix pla, de forma que identifiquem les fases en que es troba cada centre independentment dels altres.

Els punts obtinguts de cada línia es compararan amb els centres ja obtinguts de les línies anteriors, del mateix pla, per proximitat. Els valorarem segons el seu nivell de gris per decidir quin centre és millor el nou o l'anterior.

Els punts obtinguts de les línies poden classificar-se de 3 formes diferents:

- **(A)-Punt millor que l'anterior més pròxim:** En aquest cas el punt trobat es millor que l'anterior perquè és més pròxim al centre ens trobem en una fase d'aproximació al centre.
- **(B)-Punt pitjor que l'anterior més pròxim:** Si l'anterior és pitjor que l'anterior, ens indica que el punt anterior és un centre del pla. Passarem a fase 2 per sortir del centre.
- **(C)-Punt sense cap punt trobat pròxim:** Si no tenim cap punt pròxim voldrà dir que hem trobat una altre zona del tub i que ens estem acostant a un altre centre.

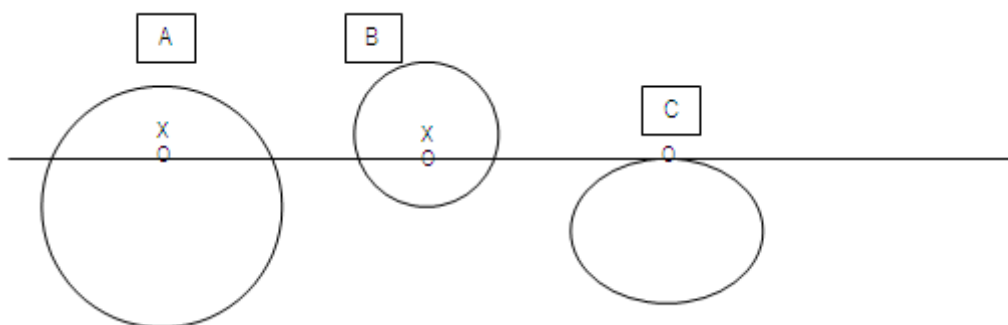


fig. 3.3-10 Tipus de possibles centres retornats per una trajectòria

Depenent de cada cas es tractarà el punt de forma corresponent creant nous centres o modificant el centre de la zona segons el cas.

El punts obtinguts de cada pla de tall del volum, a diferència dels punts que obtenim de les línies que poden o no ser centre, són realment centres. Per tant només els hem de connectar per

proximitat amb els últims punts de les trajectòries trobades. Si trobem un centre que pot ser o no pròxim a un final de tub, però aquest ja te assignat el centre del pla actual, i no ens queden mes tubs pròxims per assignar-li, es genera un nou tub.

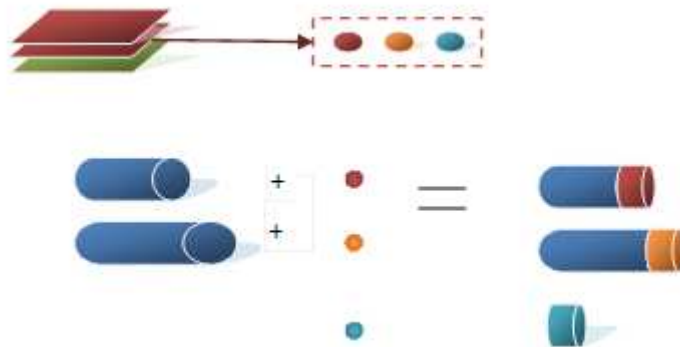


fig. 3.3-11 Exemple del tractament dels punts retornats per un pla.

En la imatge següent veiem les diferents parts que trobaríem en el mètode recorrent la imatge d'esquerra a dreta amb els dos mètodes.

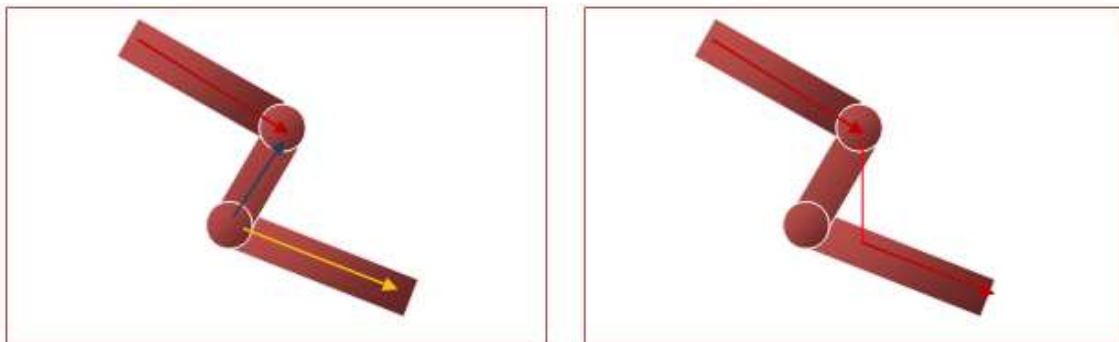


fig. 3.3-12 D'esquerra a dreta punts aconseguits per mètode multicentre i punts aconseguits pel mètode simple, tots dos sobre el mateix eix.

Si el comparem amb el recorregut que realitzàvem abans en que només detectàvem un sol punt la diferència és la següent, es veu com en aquest mètode obtenim zones que no obteníem anteriorment.

Seguint amb el cas anterior, en la figura següent, podem veure com el punt final del segment A es pot unir amb a el punt final del segment B i el punt inicial d'aquest últim amb el inicial del segment C. Obtenint així la trajectòria final.

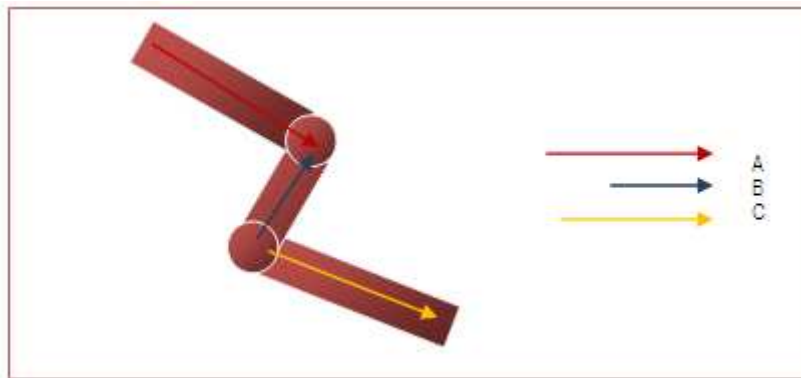


fig. 3.3-13 Subtrajectòries trobades per el mètode multacentre

▪ **Unir els punts per obtenir trajectòries:**

Un cop recorregudes totes les llesques obtindrem N subparts de la trajectòria que haurem de ajuntar per generar la trajectòria final.

Inicialment agruparem les subtrajectòries trobades pels tres eixos de manera que els punts iguals o quasi iguals ( distàncies petites) els hi assignarem els mateixos valors. D'aquesta forma aconseguim que els punts dels tres grups de subtrajectòries comparteixin els mateixos punts.

Seguidament eliminarem les parts repetides entre totes les subtrajectòries per evitar que es recorri més d'un cop una mateixa zona. Recorrem trajectòria a trajectòria i anem eliminant les parts de les trajectòries que estiguin repetides.

En el procés d'eliminar trajectòries s'ha tingut en compte que les subparts de trajectòries que estan repetides també poden esta invertides en les altres trajectòries. Es a dir que les trajectòries no es diferencien en el sentit si no en el contingut i l'ordre. Les compararem en els dos sentits per validar les parts invertides en altres trajectòries.

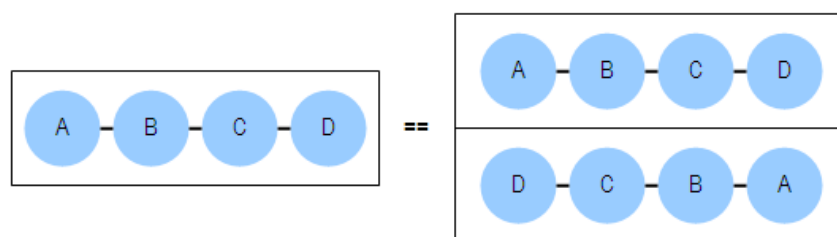


fig. 3.3-14 El sentit de la trajectòria no implica que sigui diferent

Comparació de subtrajectòries, el sentit de la trajectòria no és important, sinó el contingut i l'ordre. A partir d'aquesta premissa hem desenvolupat un mètode per eliminar parts de trajectòries repetides.

En l'exemple de la imatge següent es veu el procés d'eliminació de trajectòries. En el primer pas s'elimina de totes les resta de trajectòries les parts que ja conte la trajectòria 1, en el següent pas eliminem de totes les trajectòries els que continguin la trajectòria dos, en el 3 eliminem tot el que ja estigui a la trajectòria 3 i així aniríem seguint fins que no hi hagués cap part repetida.



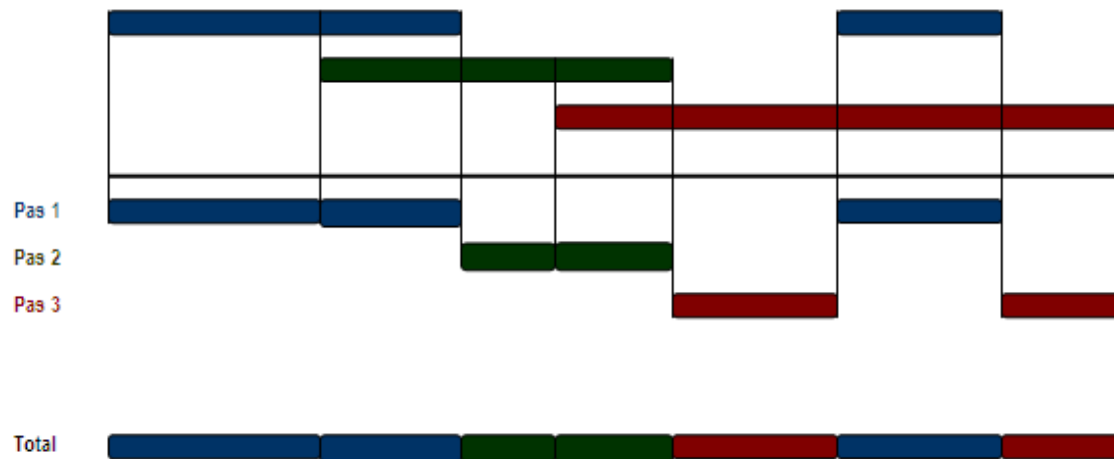


fig. 3.3-15 Eliminació de parts de trajectories repetides.

Un cop hem eliminat les parts repetides ens assegurem que les trajectòries es poden connectar pels punts inicials i finals. Per ajuntar-les ho farem amb un algorisme de backtracking que ens permetrà aconseguir la millor trajectòria.

Definim el backtracking:

- **Dades inicials** : Conjunt de trajectòries.
- **Selecció de solució parcial**: S'escullen totes les trajectòries connexes amb la última trajectòria annexada..
- **Mètode selecció millor solució**: Per validar quin es la millor trajectòria de les resultants ho escollirem per la que retorni mes punts.

Al tenir trajectòries les possibilitats de camins possibles a l'hora de generar la trajectòria disminueixen considerablement respecte el primer mètode , ja que les subtrajectòria engloben n punts. De manera que els n punts que esdevenen en n nodes del arbre de backtracking en el mètode inicial en aquest cas esdevenen en un sol node del arbre.

Un cop finalitzat el procés ens trobem que el resultat obtingut és una trajectòria que en alguns punts provoca uns desviaments que no són correctes. Això tant pot ser degut a la forma d'escollir la trajectòria com a la resolució de la imatge. Per solucionar aquest problema es farà un refinament de la trajectòria.

El refinament consistirà en recol·locar punts que es desviïn molt de la trajectòria trobada. Per determinar quins són els punts a refinar, es farà mitjançant l'angle format entre el punt en contret i els seus dos veïns. Segons els angles formats pels vectors, determinats per els tres punts sobre el pla que defineixen, s'avaluarà si el punt és eliminable. Com mostrem en la imatge següent:

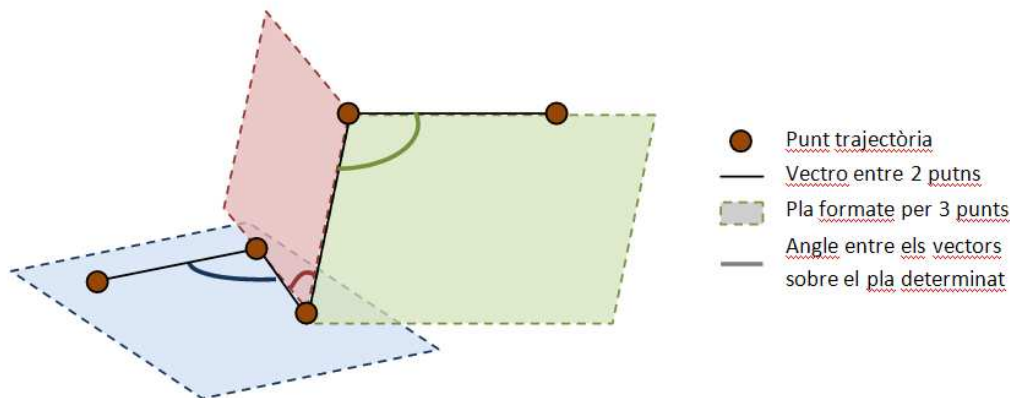


fig. 3.3-16 Angles formats pels punts de la trajectòria

Quan es troba un punt que s'ha de reubicar s'elimina el punt i s'uneixen els dos punts veïns, o en cas que aquests dos siguin molt llunyans es determina un punt mig entre els dos punts, reubicant d'aquesta forma el punt desviat.

### ***Visualització adaptada al sistema d'exploració del budell***

Per aportar la funcionalitat d'exploració, el primer que hem de pensar es la forma de visualització sobre el volum. Ja que un cop trobada la trajectòria haurem de poder veure clarament els tumors en el visor.

#### **Visualització**

Juntament amb l'exploració s'han implementat mètodes de visualització que permetran veure les zones d'interès. Com els pòlips i els tumors diferenciant-los de les parets del budell.

Aquest visualitzador implementa diferents funcions de visualització de volums, alguns representen el volum de manera poligonal i d'altres de manera volumètrica. A continuació es comenten les diferents funcions de visualització que s'han implementat per aquest visualitzador.

#### **Visualització poligonal**

La visualització poligonal permet visualitzar els contorns dels volums, veient el límit de cada zona pintant només la part que engloba la zona a visualitzar.

#### **CONTOURING**

Aquest mètode de representació interpreta únicament el contorn del volum de dades. L'algorisme permet fer el contorn de malles de triangles arbitràries i ens retorna una sortida de tipus poligonal. A sota es veu es resultat d'una renderització amb aquesta tècnica:



fig. 3.3-17 Imatge obtinguda mitjançant mètode de conturing

### *ISOSURFACE*

---

Si en la tècnica anterior s'extreien per cada imatge contorns que delimitaven una zona d'interès, en el cas de la tècnica de visualització de isosuperfícies el que es fa és crear com una malla de triangles que delimita una zona d'interès definint una superfície. Aquesta superfície ve definida per un valor constant anomenat isovalor. La tècnica més coneguda de reconstrucció d'isosuperfícies és el Marching Cubes. A continuació mostrem un exemple de reconstrucció per isosuperfícies.

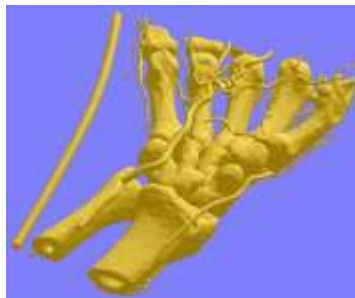


fig. 3.3-18 Imatge obtinguda amb isosurface

### Visualitzacio Volumetrica

Quan parlem de visualització volumètrica ens referim a que volem veure tots els vòxels que estan representats en el model de vòxels i no només les superfícies o els contorns que delimiten una determinada regió.

### *TEXTURE 2D – 3D*

---

Aquesta tècnica genera una textura a partir de la informació representada en el model de vòxels. A sota veiem la renderització d'una textura 2D.

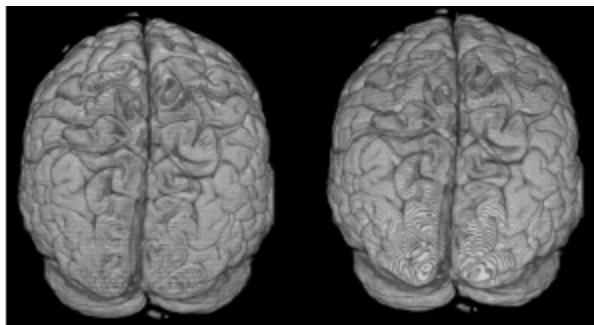


fig. 3.3-19 Visualització texture 2D

A sota es veu la projecció d'una textura 3D sobre el model de dades.

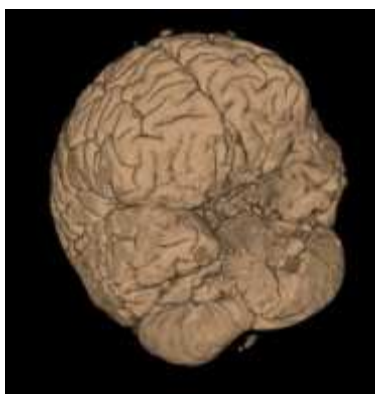


fig. 3.3-20 Visualització Texture 3D

### RAY CASTING

Per a cada píxel de la pantalla en que es vol visualitzar el model, es llença un raig des de l'observador i cap al píxel en qüestió (veure figura 3.3-21 ). De tots els punts del model que travessa el raig llençat, s'aplica una funció de composició que determina el color que cal assignar al píxel. Existeixen diferents variants d'aquesta tècnica que es diferencien per la forma que tenen de compondre els valors dels punts mostrejats sobre el raig.

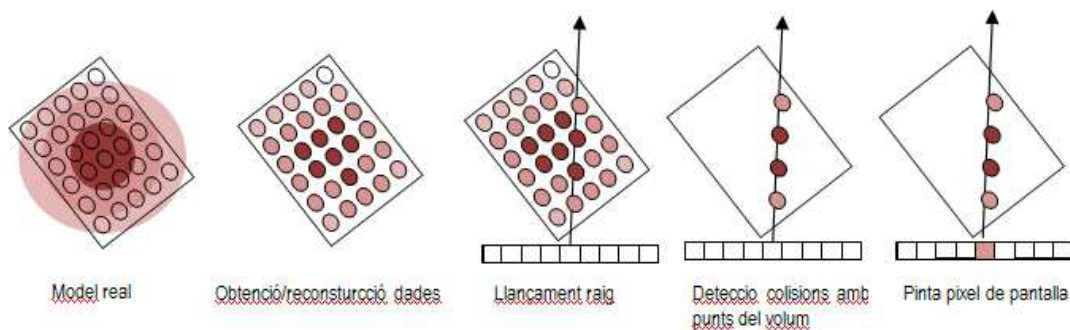


fig. 3.3-21 Ray-casting

Per generar la imatge final s'assignen els atributs gràfics (color i transparència) a tots els valors de propietat representats en el model de vòxels. Aquest mètode no usa primitives geomètriques

com a sistema de representació intermig, per tant resulten més lents, ja que cal tenir en compte tota la informació del model en el moment de generar la imatge. Les imatges que generen sempre tenen més informació que els models de superfície, ja que aquests només representen una part del model.

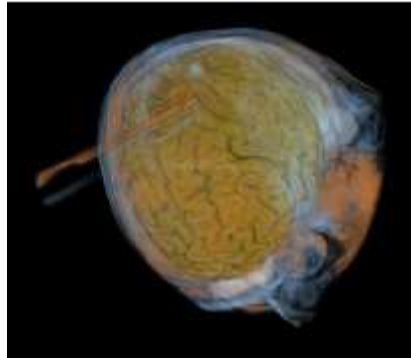


fig. 3.3-22 Imatge aconseguida per ray-casting

La imatge de sobre s'ha aconseguit mitjançant ray-casting.

De les tècniques esmentades utilitzarem una combinació de contouring i ray-casting. Utilitzarem la tècnica poligonal contouring per representar el budell complet, delimitant el volum i els tumors. Juntament amb la tècnica raycasting de visualització volumètrica.

### ***Recórrer Trajectòria***

---

Un cop decidit el tipus de visualització que requereix el nostre mòdul, haurem de desenvolupar un seguit de funcions i funcionalitats per permetre al usuari navegar a través del tub seguint la trajectòria.

Les necessitats principals d'aquest apartat són la navegació dins el volum tridimensional i la visualització de l'interior. Simplement haurem de complimentar el visor del budell de forma que l'usuari pugui interactuar-hi avançant i retrocedint la càmera sobre la trajectòria trobada. A més proporcionarem un visor general que mostrarà el volum complet, la centre del budell i la posició actual de la càmera d'aquesta manera l'usuari sabrà exactament en quina zona de l'exploració es troba.

Per desenvolupar aquest apartat haurem de conèixer el funcionament de les càmeres en els visors de *Vtk*.

La visualització d'una escena necessita, a part de conèixer la situació del volum en coordenades reals, tenir els paràmetres que defineixen la posició i abast de la càmera. Per entendre-ho millor seguidament descriurem algunes propietats referents a les càmeres que descriuen la forma que tenen de visualitzar les escenes 3D en imatges 2D.

Els atributs que defineixen les càmeres son els següents:

- **Posició:** Situació de la càmera en l'espai.
- **Focal point:** Punt de l'espai que determina la posició on apunta la càmera.
- **Direcció de projecció:** El vector format per la posició de la càmera i el *Focal point*.

- **Mètode de visualització:** Mètode utilitzat per visualitzar els objectes de l'escena.
- **Orientació:** La orientació de la càmera ve definida pel vector de projecció i el vector *View-up*, vector perpendicular al de projecció.
- **Pla de la imatge:** Pla situat al punt focal i perpendicular al vector de projecció.
- **Situació dels plans de tall:** Són dos plans que tallen la línia definida pel vector de projecció delimitant la zona de l'escena que es visualitzarà amb la càmera.
- **Angle de visió:** Angle que delimita una zona cònica de la imatge que conté el que veurà la càmera. Juntament amb els plans de tall defineixen una piràmide escapaçada que determina els límits de la zona de l'escena que veurà la càmera.

La imatge que mostrem a continuació mostra els paràmetres enumerats.

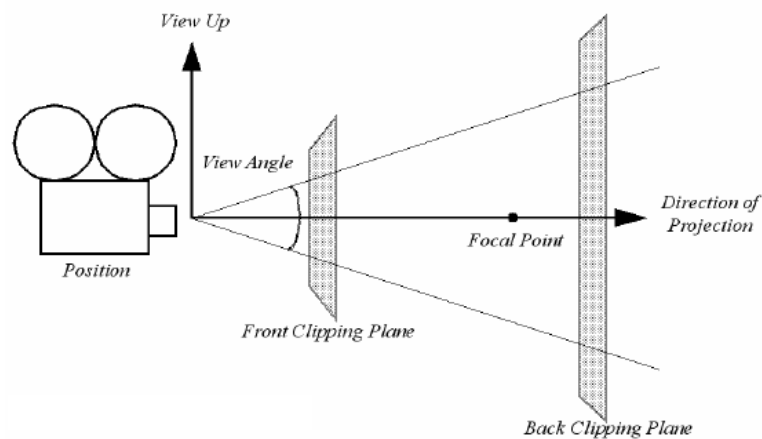


fig. 3.3-23 Atributs de la càmera

Per altre banda els factors que descriuen la càmera també defineixen els moviments que es poden realitzar amb aquesta. Els moviments que comentem estan descrits en la imatge següent.

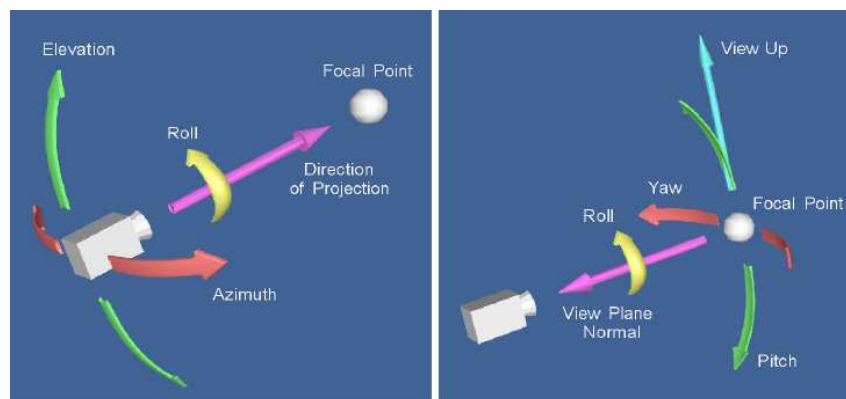


fig. 3.3-24 Moviments de la càmera centrats en el focal point , i moviments de la càmera centrats en la seva pròpia posició

- **Azimuth:** Rotació de la càmera, sobre el seu vector *View-up*, centrada en el punt focal.
- **Elevation:** Rotació de la càmera, sobre el vector resultat del producte vectorial dels vectors projecció i *view-up*, centrada en el "focal point".

- **Yaw:** Rotació del punt focal, sobre el vector view-up de la càmera, centrada a la posició de la càmera.
- **Pitch:** Rotació del punt focal, sobre el vector resultat del producte vectorial dels vectors projecció i view-up, centrada en la posició de la càmera.
- **Roll:** Rotació del vector view-up sobre el vector normal del pla de la imatge.
- **Dollying:** Acostar o allunyar la càmera del *focal point*.
- **Zomming:** Canvia l'angle de visió de la càmera.

Definits tots els paràmetres que determinen el comportament de la càmera. Descriurem els factors que més rellevància tenen en el nostre mòdul.

Per descomptat, la posició de la càmera ja que ens interessa canviar-la per moure'ns sobre la trajectòria. El *focal point* ens interessa per fer que la càmera miri al punt següent de la trajectòria. A partir d'aquestes dues posicions podrem moure la càmera a través del budell.

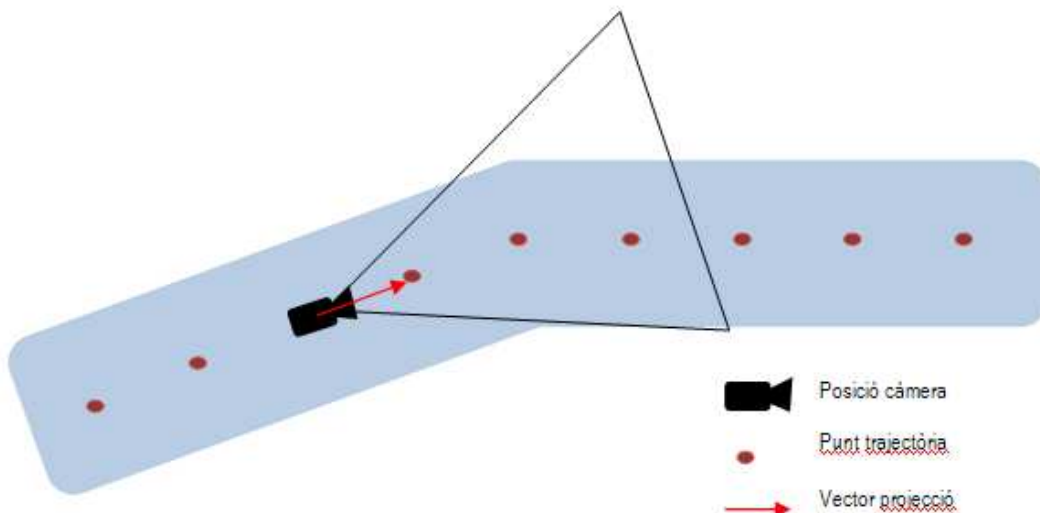


fig. 3.3-25 Posició càmera sobre la trajectòria

Tal com mostra la imatge anterior la posició de la càmera serà el punt actual de la trajectòria, mentre que el punt focal estarà definit per el punt següent de la trajectòria. Els altres paràmetres es definiran automàticament.

### 3.3.2 Diagrames del mòdul sistema d'exploració del budell

El següent diagrama conté tots els casos d'ús que formen part del mòdul d'exploració del budell.

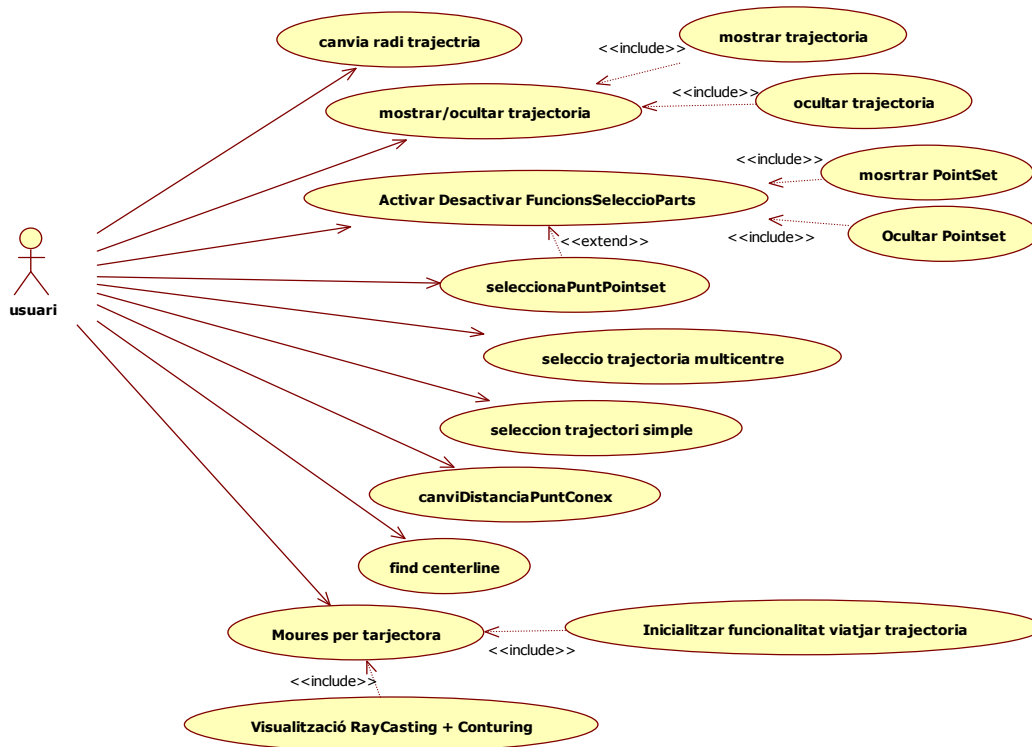


fig. 3.3-26 Diagrama cas d'ús d'exploració del budell

En aquest mòdul es desenvoluparan les següents funcionalitats:

- » **Canviar radi de trajectòria**
- » **Mostrar/Ocultar trajectòria**
  - Mostrar trajectòria
  - Ocultar trajectòria
- » **Seleccionar trajectòria simple**
- » **Seleccionar trajectòria multacentre**
- » **Canviar distància connexió punts**
- » **Cercar trajectòria**
  - Cerca centerline amb mètode multacentre
  - Cerca centerline amb mètode simple
- » **Moure's per la trajectòria**
  - Inicialitzar funcionalitats viatjar per la trajectòria
  - Aplicar visualització ray-casting + conturing.



## **Cercar trajectòria**

---

- **Diagrama cas d'ús:**

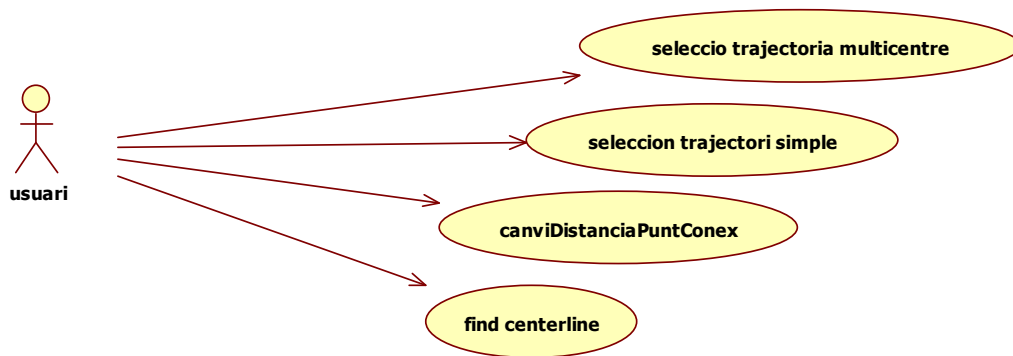


fig. 3.3-27 Diagrama cas d'ús "Cercar trajectòria"

Aquest mòdul serà l'encarregat d'indicar a l'aplicació el tipus de mètode de cerca del centre a utilitzar i d'executar-lo a petició del usuari. A més es permetrà definir la distància de connexió entre els punts.

Seguidament exposarem els casos d'ús de l'apartat actual:

### Selecció trajectòria multicentre

- **Fitxa cas d'ús:**

Cas d'ús	Selecció trajectòria multicentre
Descripció	Marcar/Desmarcar l'indicador de cerca pel mètode multicentre
Actors	Usuari
Precondició	
Flux Principal	1. Es marcar/desmarcar el checkbox d'execució del mètode multicentre 2. S'assignarà el valor corresponent cert/falç al indicador de cerca per mètode multicentre
Fluxos Alternatius	
Postcondició	Hem canviat el valor de l'indicador de cerca pel mètode multicentre

Aquest procés simplement marcarà o desmarcarà l'indicador de cerca pel mètode multicentre. En el moment d'execució el valor d'aquest indicador determina si el mètode de cerca per multicentre s'haurà d'executar o no.

- **Diagrama de seqüència:**

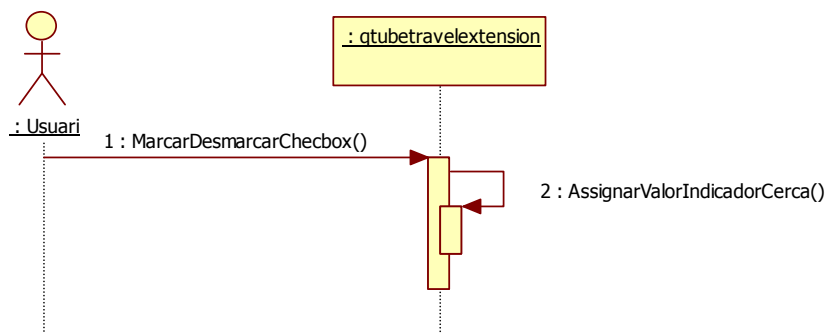


fig. 3.3-28 Diagrama de seqüència "Selecció mètode multicentre"

*Selecció trajectòria simple*

- **Fitxa cas d'us:**

Cas d'ús	Selecció trajectòria simple
Descripció	Marcar/Desmarcar l'indicador de cerca pel mètode simple
Actors	Usuari
Precondició	
Flux Principal	<ol style="list-style-type: none"> <li>3. Es marcar/desmarcar el checkbox d'execució del mètode simple</li> <li>4. S'assignarà el valor corresponent cert/falç al indicador de cerca per mètode simple</li> </ol>
Fluxos Alternatius	
Postcondició	Hem canviat el valor de l'indicador de cerca pel mètode simple

Igual que en el cas anterior es marcarà o desmarcarà l'indicador de cerca per el mètode simple. I en el moment d'execució el valor d'aquest indicador determinarà si el mètode de cerca simple s'haurà executar o no.

- **Diagrama de seqüència:**

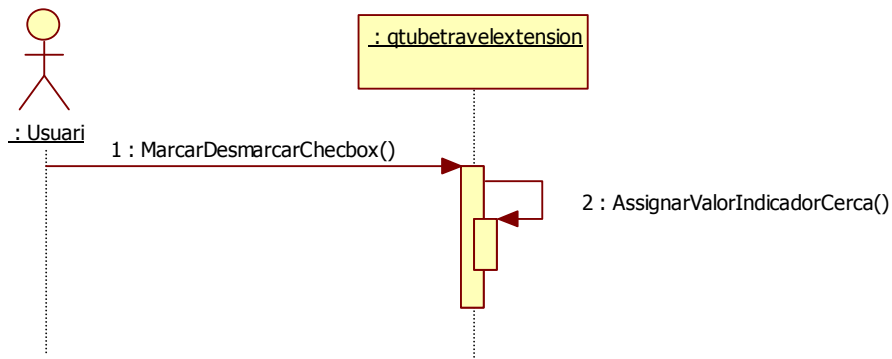


fig. 3.3-29 Diagrama seqüència "Selecció trajectòria simple"

### Canviar distàncies connectivitat punts

- **Fitxa cas d'us:**

Cas d'ús	Canviar distància punts connexió
Descripció	Permet canviar la distància de connexió
Actors	Usuari
Precondició	
Flux Principal	1. Canviar valor distància 2. Assignar valor distància
Fluxos Alternatius	
Postcondició	Hem canviat el valor de la distància de connexió

Aquesta opció s'ha definit degut a que en diferents execucions, utilitzant el paràmetre de connectivitat dels punt per defecte del mètode, no es trobava el centreline complert. Per aquest motiu s'ha permès que l'usuari pugui augmentar o disminuir el factor que determina el rang de connectivitat dels punts.

Simplement es determina a quina distància es considera que dos punts són connexes. El valor entrat per l'usuari es multiplica per la mitjana dels valors de spacing dels tres eixos.

- **Diagrama de seqüència:**

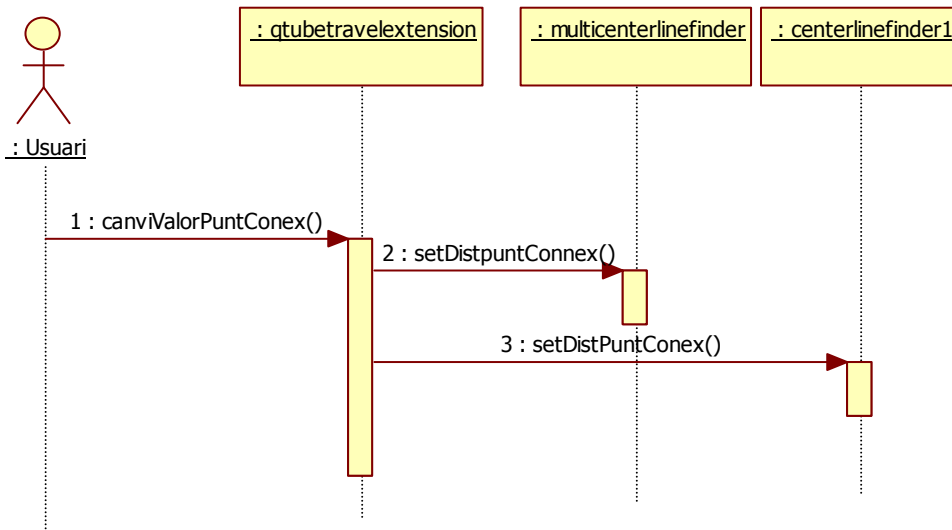


fig. 3.3-30 Diagrama de seqüència "Canviar distàncies connectivitat punts"

**Cerca centre**

**- Fitxar cas d'ús:**

Cas d'ús	Cerca centre
Descripció	Cercar el centre del budell
Actors	Usuari, Sistema
Precondició	
Flux Principal	<ol style="list-style-type: none"> <li>1. Si cerca per mètode multicentre seleccionat             <ol style="list-style-type: none"> <li>1.1. Cridar mètode cerca centerlin multicentre</li> </ol> </li> <li>2. Si cerca per mètode simple seleccionat             <ol style="list-style-type: none"> <li>2.1. Cridar mètode cerca centeline simple</li> </ol> </li> </ol>
Fluxos Alternatius	
Postcondició	S'han trobat els centerlines que l'usuari ha decidit

Al cercar el centre avaluarem els valors dels indicadors de cada mètode per determinar quins dels dos mètodes s'han d'executar. El valor dels indicadors estan indicats per l'usuari mitjançant la interfície.

**- Diagrama seqüència:**

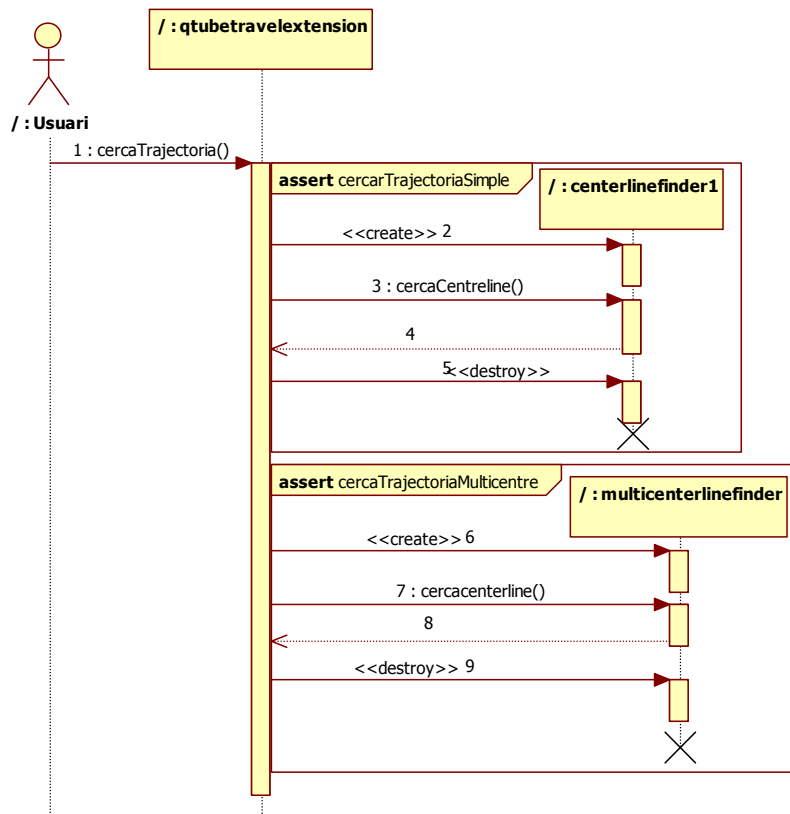


fig. 3.3-31 Diagrama de seqüència "Cerca centreline"

### Cerca centerline simple

- **Fitxa cas d'ús:**

Cas d'ús	Cerca centerline simple
Descripció	A partir d'una imatge segmentada (binaria). Es cercarà el centerline del budell i es mostrarà per pantalla sobre la representació en 3D del volum.
Actors	Sistema
Precondició	Ha de tenir una imatge segmentada.
Flux Principal	<ol style="list-style-type: none"><li>1. Crear Centerlinefinder</li><li>2. Cercar centerline<ol style="list-style-type: none"><li>2.1. Obtenir mapa distàncies sobre imatge</li><li>2.2. Obtenir els punts del centre recorrent la imatge per els tres eixos X Y i Z.</li><li>2.3. Agrupar punts de les trajectòries obtingudes</li><li>2.4. Eliminar repetits</li><li>2.5. Backtraking per Obtenir millor trajectòria, ajuntant les parts de subtrajectòries</li></ol></li><li>3. Afegir trajectòria a l'escena dels visualitzadors</li><li>4. Refrescar visualitzadors</li></ol>
Fluxos Alternatius	
Postcondició	S'ha obtingut el centre i s'ha mostrat pels visors 3D

Crearem un objecta de la classe cercacenterline i li passarem el volum a tractar (1).

El procés de cerca del centerline (2) es basa en la obtenció de la imatge segmentada i la aplicació del mètode mapa de distàncies sobre la segmentació (2.1). A partir d'aquest punt s'obtenen amb recorreguts, pels eixos, sobre la imatge els punts que formen la trajectòria (2.2).

Un cop s'obtenen els punts que poden formar part de la trajectòria farem una unificació dels punts de les trajectòries (2.3). Llavors eliminarem els repetits o punts que per proximitat es poden considerar repetits (2.4).

Un cop hem polit les dades obtingudes aplicarem el mètode de backtracking per obtenir la millor solució(2.5).

Finalment obtenim un actor que representa la trajectòria i l'afegim al render de l'escena 3D del visor determinat (3) i el refresquem perquè mostri la trajectòria (4).

Al finalitzar el procés haurem trobat el centre del budell. Es guardarà la trajectòria per poder-hi interactuar accedint als punts, visualitzant-lo, entre d'altres.

- **Diagrama de seqüència:**

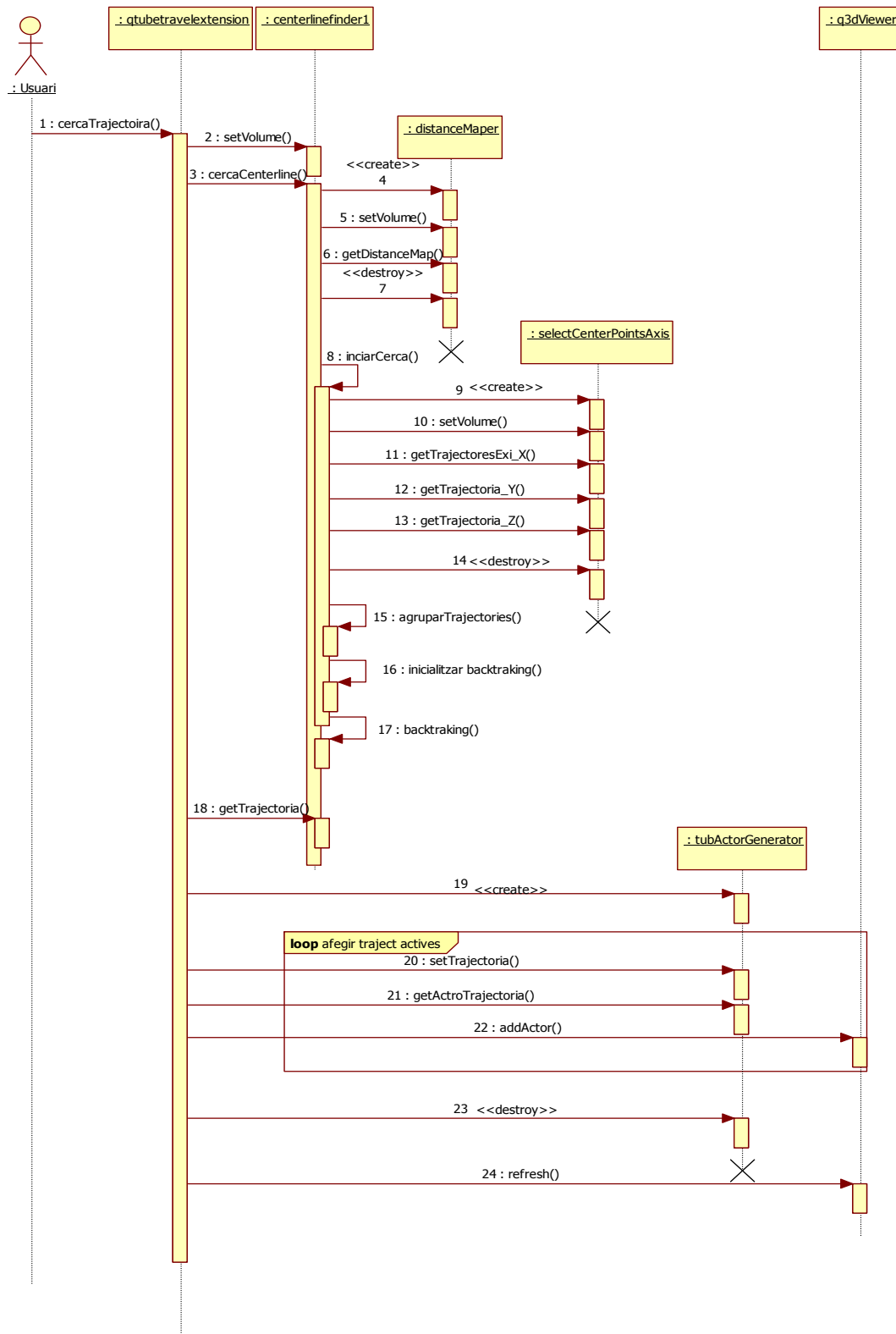


fig. 3.3-32 Diagrama de seqüència "Cerca centerline simple"

**Cerca centerline multicentres per pla**

- **Fitxa cas d'ús:**

Cas d'ús	Cerca centerline multicentre
Descripció	A partir d'una imatge segmentada (binària). Es cercarà el centerline del budell i es mostrarà per pantalla sobre la representació en 3D del volum.
Actors	Sistema
Precondició	Ha de tenir una imatge segmentada.
Flux Principal	<ol style="list-style-type: none"> <li>1. Crear multiCenterlinefinder</li> <li>2. Cercar centerline <ol style="list-style-type: none"> <li>2.1. Obtenir mapa distàncies sobre imatge</li> <li>2.2. Obtenir els punts del centre recorrent la imatge per els tres plans X Y i Z.</li> <li>2.3. Agrupar punts de les trajectòries obtingudes</li> <li>2.4. Eliminar repetits</li> <li>2.5. Backtracking per obtenir millor trajectòria, unint subtrajectòries</li> <li>2.6. Refinar trajectòria obtinguda.</li> </ol> </li> <li>3. Afegir trajectòria a l'escena del visualitzador</li> <li>4. Refrescar el visualitzador</li> </ol>
Fluxos Alternatius	
Postcondició	S'ha obtingut el centre i s'ha mostrat per el visor 3D

Crearem un objecte de la classe multcenterlinefinder i li passarem el volum a tractar (1).

El procés de cerca del centerline (2) es basa en la obtenció de la imatge segmentada i la aplicació un mapa de distàncies sobre la segmentació (2.1). A partir d'aquest punt s'obtenen amb recorreguts, per els eixos, sobre la imatge els punts que formen la trajectòria (2.2).

Un cop s'obtenen els punts que poden formar part de la trajectòria farem una unificació de les subtrajectòries (2.3). Llavors eliminarem els repetits o punts que per proximitat es poden considerar repetits (2.4).

Un cop hem polit les dades obtingudes aplicarem el mètode de backtracking per obtenir la millor solució(2.5).

Quan ja tenim la trajectòria en fem un refinament eliminant punts massa pròxims o suavitzant els desviaments excessius de la trajectòria (2.6).

Finalment obtenim un actor que representa la trajectòria i l'afegim al render de l'escena 3D del visor (3) i el refresquem perquè mostri els canvis (4).

Al finalitzar el procés haurem trobat el centre del budell. Es guardarà la trajectòria per poder-hi interactuar accedint als punts, visualitzant-lo, entre d'altres.



- **Diagrama de seqüència:**

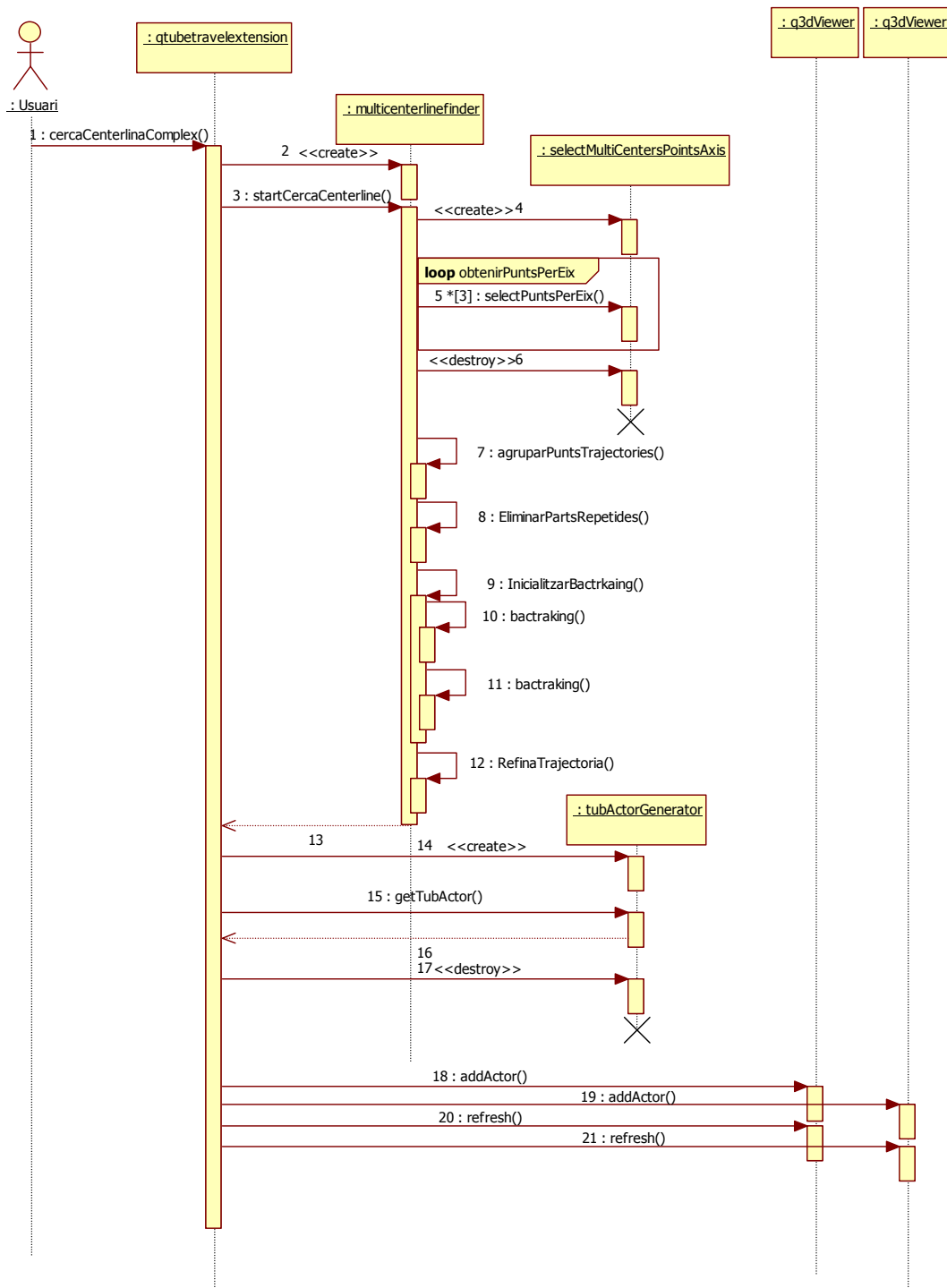


fig. 3.3-33 Diagrama de seqüència "Cerca centerline amb multicentres per pal"

### **Visualització del volum**

- **Diagrama cas d'ús:**



fig. 3.3-34 Diagrama visualització

La visualització ens permetrà diferenciar les zones d'interès del budell gràficament. La visualització la farem sobre un imatge que contindrà dos zones a visualitzar que seran els tumors i el budell.

Seguidament exposarem el cas d'ús de l'apartat actual:

### **Visualització RayCasting + Conturing**

- **Fitxa cas d'us:**

Cas d'ús	Visualització Raycasting + Conturing
Descripció	Mètode de visualització per diferenciar visualment els tumors del budell.
Actors	Sistema
Precondició	Visor generat
Flux Principal	<ol style="list-style-type: none"> <li>1. Assignem volum al visor</li> <li>2. Assignar al visor el mètode de visualització RayCasting+Conturing</li> <li>3. Aplicar filtres d'extracció de volum (raycasting)</li> <li>4. Aplicar filtres d'extracció de contorn (conturing)</li> <li>5. Afegir al render del visor la visualització de raycasting</li> <li>6. Afegir al render del visor la visualització de conturing</li> <li>7. Refrescar render</li> </ol>
Fluxos Alternatius	
Postcondició	Permet al usuari moure la càmera a través del budell seguint el centre trobat

La visualització serà transparent per l'usuari, es el visor el que s'encarregarà d'executar el mètode. Per tant es en el visor on haurem d'implementar el mètode de visualització.

El procés seguirà els següents passos. Assignem el volum a visualitzar al visor (1). Determinem el mètode de visualització que haurà d'aplicar el visor (2). I executem el mètode de refresc que actualitza el visor. Aquest mètode executarà internament el mètode de raycasting amb el que obtindrem una visualització volumètrica(3) i el mètode de contouring amb el que obtindrem un contorn del volum (4). Després assignarem els resultats dels mètodes al render del visor perquè els incorpori a l'escena (5 i 6) i finalment actualitzem el visor per mostrar les dues visualitzacions conjuntes (7).

- **Diagrama de seqüència:**

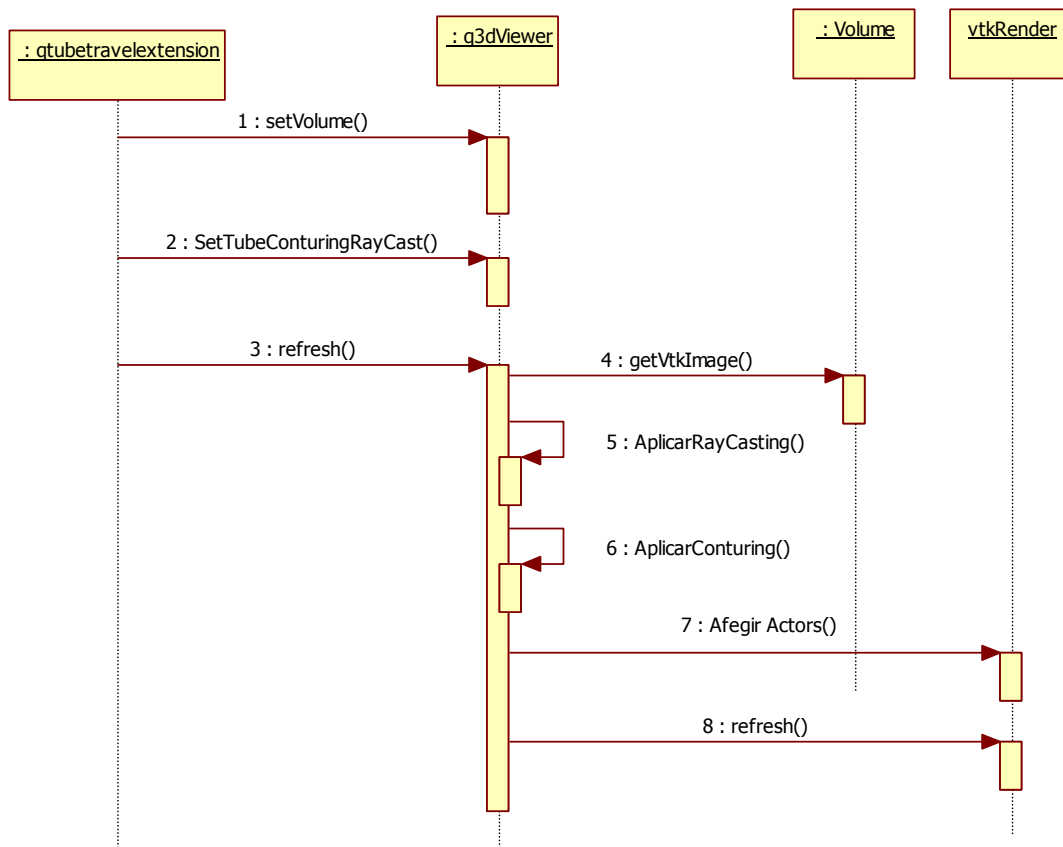


fig. 3.3-35 Diagrama de seqüència Raycasting + Conturing

### ***Viatjar pel centre del budell***

---

- **Casos d'ús:**

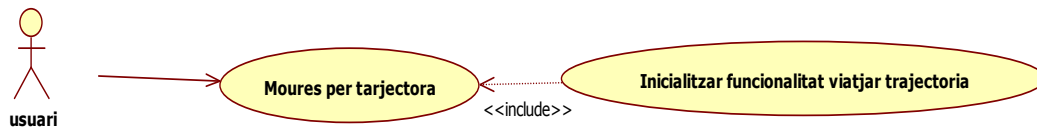


fig. 3.3-36 Diagrama de casos d'ús "Viatjar pel centre del budell"

Aquest mòdul aportarà la funcionalitat d'exploració del budell. Permetent al usuari viatjar amb la càmera a través de la trajectòria. El primer cop que s'inicia el viatge es prepararà la interfície i les variables internes que intervenen en el recorregut de la trajectòria.

A continuació exposem els diagrames corresponents a cada cas d'ús.

*Inicialitzar funcionalitats per viatjar per la trajectòria*

**- Fitxa cas d'ús:**

Cas d'ús	Inicialitzar viatge manual
Descripció	Prepararà la interfície i inicialitzarà els objectes necessaris per recórrer la trajectòria.
Actors	Sistema
Precondició	S'ha de tenir un centerline generat
Flux Principal	<ol style="list-style-type: none"> <li>1. Usuari inicialitza funcionalitats</li> <li>2. Seleccionem trajectòria</li> <li>3. Inicialitzar funcionalitats interfície</li> <li>4. Crear càmera</li> <li>5. Pintar càmera en el visor general, a la posició inicial de la trajectòria.</li> <li>6. Actualitzar la posició de la càmera del visor intern, a la posició inicial de la trajectòria.</li> </ol>
Fluxos Alternatius	
Postcondició	S'ha inicialitzat el necessari perquè l'usuari interactuï amb el viatge a través del tub.

A petició de l'usuari(1) es seleccionarà la trajectòria(2) i a partir de les dades d'aquesta s'inicialitzaran les funcionalitats de la interfície, com els botons, spinbox, atributs, etc (3). Llavors es crearà una càmera sobre la posició inicial de la trajectòria (4). I es pintarà en el visor general (5) i situarem la càmera del visor, que mostra el budell internament, al inici de la trajectòria (6). Finalment refrescarem els visors perquè mostrin els canvis (7 i 8 ). Aquest procés es pot seguir en el diagrama de segmentació següent.

**- Diagrama de seqüència:**

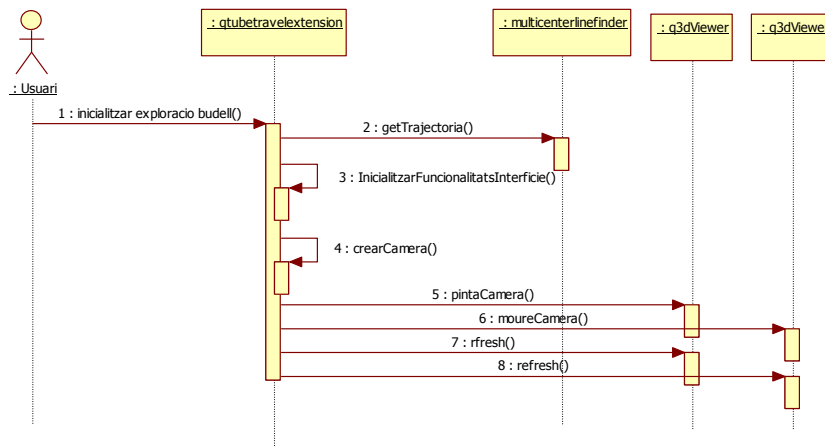


fig. 3.3-37 Diagrama de seqüència "Inicialitzar funcionalitats trajectòria"

*Moure's per la trajectòria*

- **Fitxar cas d'us:**

Cas d'ús	Moure's per la trajectoria
Descripció	Simular un viatge a través del tub, recorrent amb una càmera el centre del budell seguint el centerline trobat. Movent-se sota petició de l'usuari
Actors	Sistema
Precondició	S'ha de tenir un centerline generat
Flux Principal	<ol style="list-style-type: none"> <li>1. Seleccionar posició del tub</li> <li>2. Eliminar càmera anterior</li> <li>3. Genera nova càmera</li> <li>4. Assignar posició de la trajectòria a la càmera</li> <li>5. Pinta càmera a la vista general</li> <li>6. Actualitzar visor general i el visor de la trajectòria</li> </ol>
Fluxos Alternatius	
Postcondició	Permet al usuari moure la càmera a través del budell seguint el centre

A petició de l'usuari (1), eliminarem la càmera actual. I amb les dades del punt en concret cap o es vol moure l'usuari, en generarem una de nova (3) l'afegirem als visor 3D per indicar la posició (4). I movem la càmera del visor intern del budell, seguint la trajectòria (5). Finalment refrescarem els visors perquè mostrin els canvis efectuats ( 6 i 7 ).

- **Diagrama de seqüència:**

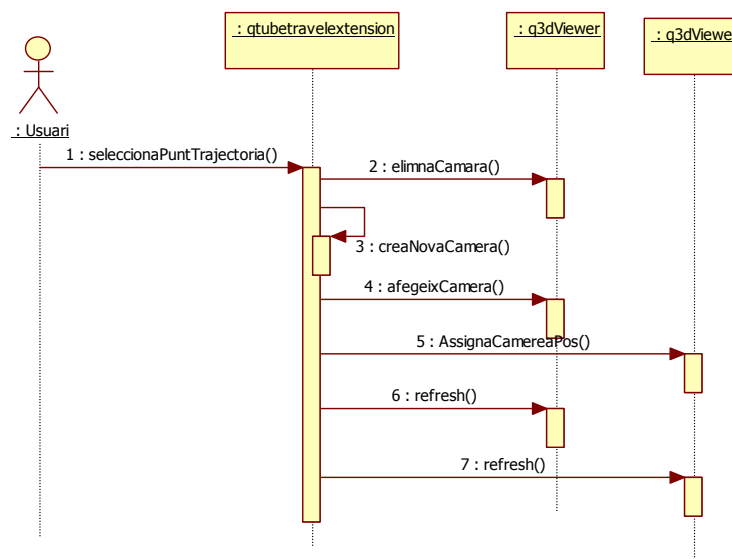


fig. 3.3-38 Diagrama de seqüència "Moure's per la trajectòria"

### ***Funcionalitat visualització trajectòries***

---

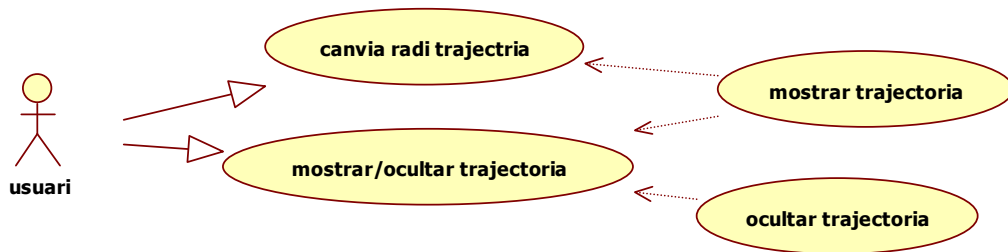


fig. 3.3-39 Diagrama cas d'ús visualització trajectòries

Aquest mòdul es desenvoluparà per poder interactuar amb el visor de trajectòries permetent a la usuari decidir quines trajectòries vol que es mostrin i quines no. A més s'ha permès que l'usuari determini el radi de la trajectòria, no ens hem torbat en cap cas que sigui necessari, però la mida dels volums desenvolupats són petits en comparació amb els volums reals, i aquesta funció permetrà canviar el radi per poder veure millor la trajectòria en volums mes grans.

La part de visualització de trajectòries ha estat implementada per poder fer comparacions sobre els resultats obtinguts pels diferents mètodes.

### Mostra/ocultar trajectòria

- **Fitxa cas d'ús:**

Cas d'ús	Mostrar/ocultar trajectòria
Descripció	Serveix per mostrar i ocultar una trajectòria determinada
Actors	Usuari, Sistema
Precondició	S'ha d'haver creat la trajectòria que es vol mostrar.
Flux Principal	1. Activar checkbox trajectòria 2. Generar Actor a partir de la trajectòria 3. Pintar trajectòria al visor general 4. Refrescar vista general
Fluxos Alternatius	1. Desactiva checkbox trajectòria 2. Eliminar trajectòria del visor general 3. Refrescar vista general
Postcondició	Permet al usuari mostrar o ocultar una trajectòria determinada

Mitjançant la interfície l'usuari pot mostrar les trajectòries que s'han trobat. A partir del valor d'un checkbox si esta marcat o desmarcat es mostrarà o ocultarà la trajectòria.

- **Diagrama seqüència:**

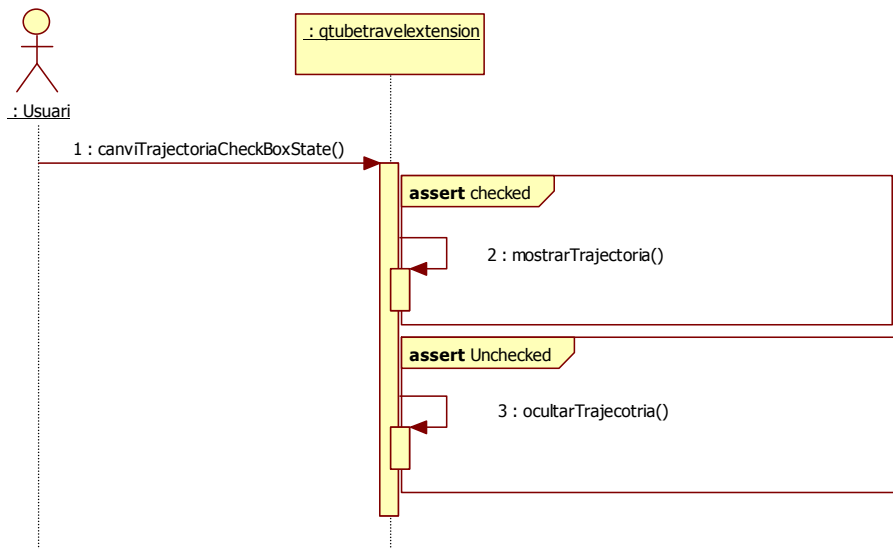


fig. 3.3-40 Diagrama seqüència mostrar/ocultar trajectòries



### Canviar radi trajectòries

- **Fitxar cas d'ús:**

Cas d'ús	Canvi radi de trajectòries
Descripció	Canvia el radi de la trajectòria
Actors	Usuari, Sistema
Precondició	
Flux Principal	1. Canvi valor checkbox 2. Assignar valor radi_tubs a generador d'actors
Fluxos Alternatius	
Postcondició	S'ha assignat el nou radi al tubaActorGenerator

L'usuari activarà o desactivarà la opció, i assignarà el valor per calcular la connexió dels punts.

- **Diagrama seqüència:**

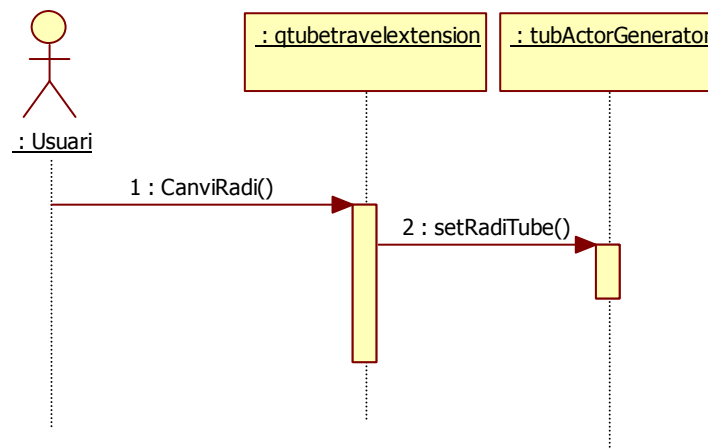


fig. 3.3-41 Diagrama seqüència canviar radi trajectòries

### Mostrar trajectòria

- **Fitxar cas d'ús:**

Cas d'ús	Mostrar trajectòria
Descripció	Mostra una trajectòria al viso general
Actors	Sistema
Precondició	S'ha d'haver creat la trajectòria que es vol mostrar.
Flux Principal	1. Marca el checkbox 2. Afegeix trajectòria a visor 3. Repintar visor per mostra els canvis
Fluxos Alternatius	
Postcondició	Permet al usuari mostrar o ocultar una trajectòria determinada

Mitjançant la interfície l'usuari marcant un checkbox indicarà a l'aplicació que vol mostrar la trajectòria (1).

S'actualitzarà el visor afegint la trajectòria en concret(2).

Un cop actualitzada l'escena es refrescarà el visor perquè mostri els canvis efectuats(3).

- **Diagrama seqüència:**

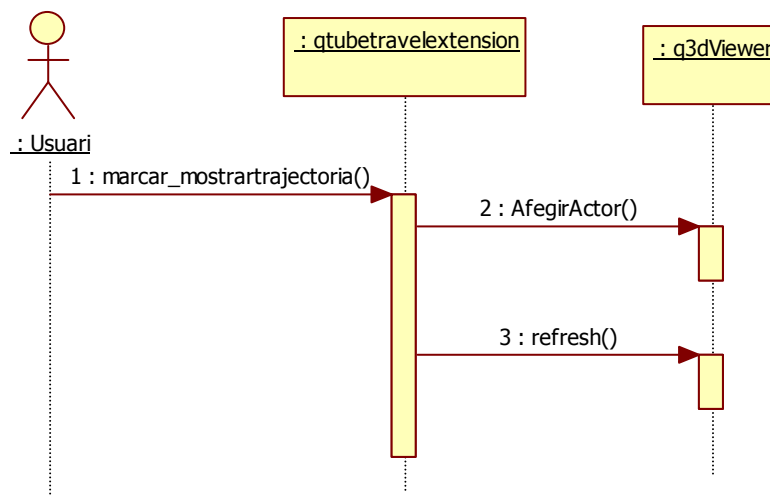


fig. 3.3-42 Diagrama de Seqüència "Mostrar trajectòria"

Ocultar trajectòria

- **Fitxa cas d'ús :**

Cas d'ús	Ocultar trajectòria
Descripció	Elimina una trajectòria del visor general
Actors	Sistema
Precondició	S'ha d'haver creat la trajectòria que es vol ocultar.
Flux Principal	1. Desmarca el checkbox 2. Elimina trajectòria del visor general 3. Repinta visor per actualitzar canvis
Fluxos Alternatius	
Postcondició	Permet al usuari ocultar una trajectòria determinada

Mitjançant la interfície l'usuari desmarcant un checkbox indicar a l'aplicació que vol ocultar la trajectòria (1).

S'actualitzarà el visor traient la trajectòria en concret(2).

Un cop actualitzada l'escena es refrescarà el visor perquè mostri els canvis efectuats(3).

- **Diagrama seqüència:**

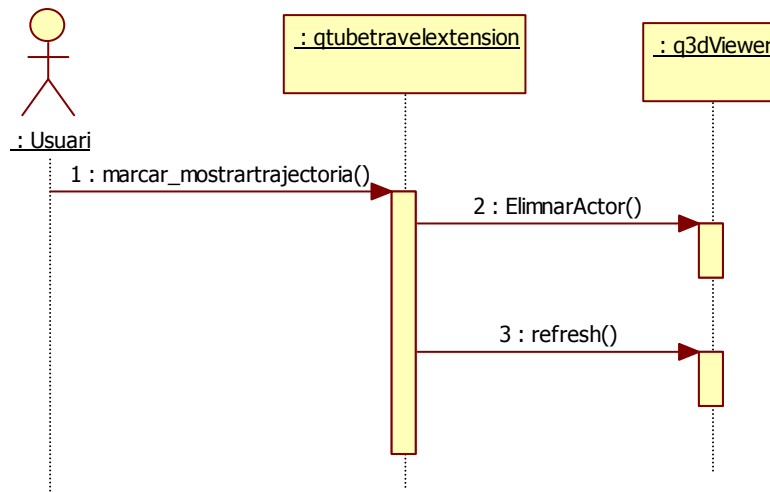
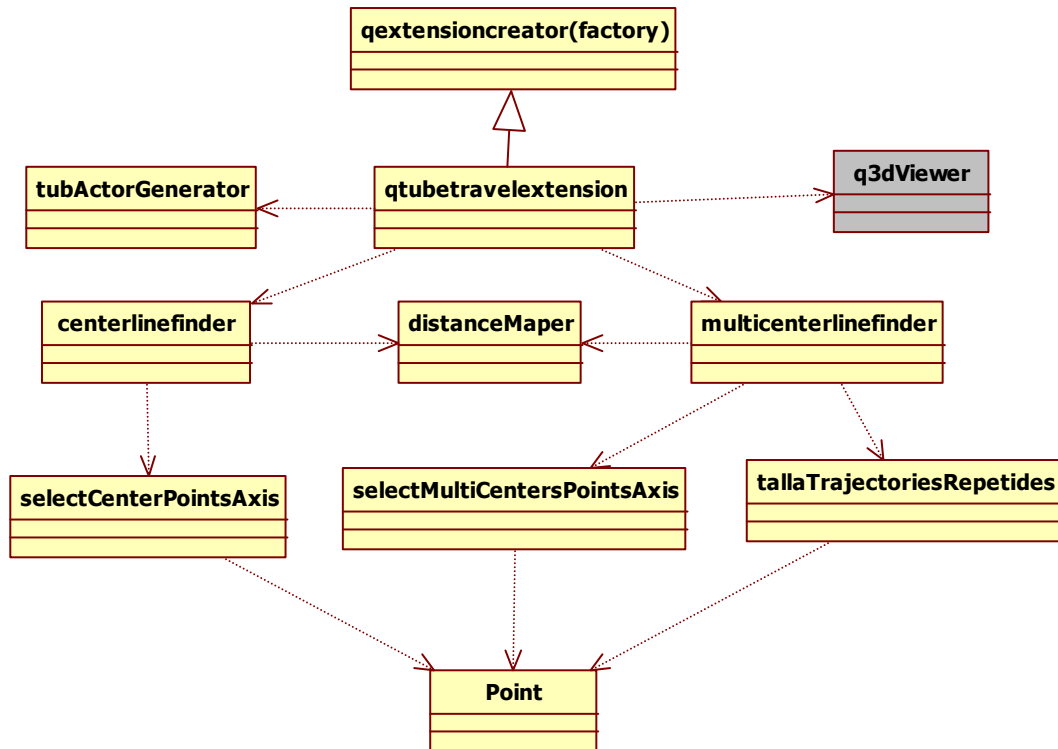


fig. 3.3-43 Diagrama seqüència ocultar trajectòria

### 3.3.3 Classes del mòdul implementar un sistema d'exploració del budell



Per dotar l'aplicació de les funcionalitats del mòdul descrit s'han implementat les classes que es mostren en el diagrama de classes anterior. Les classes de color gris ja estaven implementades en la aplicació però s'han hagut de modificar i/o ampliar degut als requeriments del mòdul.

#### DistanceMaper

Aquesta classe ens permet obtenir un distanceMap sobre una imatge binària que es passa com a paràmetre d'entrada al crear l'objecte de la classe.

Alguns mètodes d'interès:

- **aplicarFiltreDistanceMap()** :

Aquest mètode és el que aplica el filtre itkDanielssonDistanceMapFilter sobre la imatge entrada. Internament es defineixen paràmetres que es necessiten per la seva execució i s'executa, obtenint com a resultat una nova imatge amb el mapa de distàncies.

#### Centerlinefinder

Aquesta classe conté els algorismes i mètodes de cerca del centre a partir d'un volum. Internament s'obté el mapa de distàncies sobre la imatge entrada.

Alguns mètodes d'interès:

- ***divideixTrajectoria( llistaPuntsADividir, ColeccióDeLListes, ColeccióDePuntsGenerals )*** :

La forma de selecció de les trajectòries per els diferents eixos , provoca que les trajectòries tornades tinguin punts consecutius que no son connexes. Aquest mètode parteix la trajectòria per aquest punts.

- ***netejarTrajectoria( Trajectoria )*** :

Recorrem la trajectòria passada per paràmetre i l'eliminem els punts consecutius que son tant pròxims que els considerem iguals.

- ***fusioTrajectories( )*** :

Treballa sobre els atributs de classe que contenen les trajectòries a fusionar. La fusió consisteix en unificar els punts de les diferents trajectòries. És a dir, que si un punt d'una trajectòria és molt pròxim a un altre punt d'una altre trajectòria només ens quedarem amb una de les dos opcions a l'hora de generar la trajectòria.

#### ***Backtracking:***

- ***ibuscarmillorCentre( PuntsTotals, Trajectòries)*** :

Inicialitza els atributs necessaris per executar el mètode de backtracking.

- ***buscarmillorCentre( PuntsTotals, Trajectòries, PuntsVisitats, trajectòriaActual, ColeccióTrajectòriesResultants)*** :

Mètode backtracking que va generant la solució sobre l'atribut trajectòria actual, i obté els possibles punts pròxims dels paràmetres puntsTotals i trajectòries que contenen tota la informació a partir de la qual s'ha de generar el resultat. Quan s'arribi a una crida final del mètode haurem construït una solució i la guardarem a la col·lecció de solucions finals en l'atribut ColeccióTrajectòriesResultants.

- ***buscarPuntSeguent( PuntActual, PuntsTotals, Trajectòries, PuntsVisitats, trajectòriaActual )*** :

Mètode que ens retorna la millor opció com a punt següent a partir del punt actual que estigui dins de les dades generals i que encara no hagi estat visitat.

- ***seleccionaTrajectoriaFinal ( )*** :

Obté la millor trajectòria dels resultats obtinguts per el mètode de backtracking. Per valorar quina és la millor obtenim la que conté més nombre de punts, amb la intenció de assegurar-nos que la solució escollida passa per tots els punts del centre del budell.

#### ***SelectCenterPointAxis***

---

Aquesta classe ens serveix per obtenir a partir d'un volum (amb mapa de distàncies) i un eix tots els punts que poden formar part del centre del budell, recorrent la imatge per els plans perpendiculars al eix descrit. El grup de punts es retonen com una trajectòria que conte els punts ordenats per ordre d'obtenció.

Alguns mètodes d'interès:

- ***CercaTrajectòria()*** :

Recorre cada pla de la imatge obtinguen un punt per cada pla, i els va guardant en una trajectòria que es retornarà com a resultat.

### *MultiCenterLineFinder*

---

Aquesta classe conte els algorismes i mètodes de cerca del centre a partir d'un volum. Però aquest mètode te en compte que el volum es irregular i que es poden contenir parts separades en el mateix pla.

Alguns mètodes d'interès:

- ***netejarTrajectoria( Trajectoria )***:

Recorrem la trajectòria passada per paràmetre i l'eliminem els punts consecutius que son tant pròxims que els considerem iguals.

- ***polirTrajectoriaPerAngles( )***:

A partir d'un angle límit i una variació definits per la classe, elimina els punts que formen un angle que supera la variació permesa sobre l'angle límit.

- ***fusioTrajectories( )***:

Treballa sobre els atributs de classe que contenen les trajectòries a fusionar. La fusió consisteix en unificar els punts de les diferents trajectòries. És a dir que si un punt d'una trajectòria és molt pròxim a un altre punt d'una altra trajectòria només ens quedarem amb una de les dos opcions a l'hora de generar la trajectòria.

- ***eliminarPartsRepetidesTrajectòries( )***:

De les trajectòries obtingudes per la classe *cercaMultiCenterPointAxis* s'eliminen les parts que es repeteixen en les diferents subtrajectòries.

### *Bactracking:*

- ***ibuscarMillorCentre( PuntsTotals, Trajectòries)***:

Inicialitza els atributs necessaris per executar el mètode de backtracking.

- ***buscarMillorCentre( )***:

Treballem amb atributs de classe. A partir de una trajectòria inicial busquem les trajectòries connexes anirem construint la trajectòria final ajuntant les subtrajectòries. Igual que en el mètode simple anirem afegint, a una col·lecció de trajectòries, els resultats obtinguts en cada branca de l'arbre generat per el backtracking.

- ***seleccionaTrajectoriaFinal ( )***:

Obté la millor trajectòria dels resultats obtinguts per el mètode de backtracking. Per valorar quin és el millor obtenim el que conté més nombre de punts en el resultat, amb la intenció de assegurar-nos que passa per els punts inicial i final del budell.

### *SelectMultiCenterPointAxis*

---

Aquesta classe ens serveix per obtenir, a partir d'un volum de budell amb el mapa de distàncies aplicat i un eix d'exploració, els punts que poden forma part del centre. Per obtenir-los es recorre la imatge pels plans perpendiculars al eix descrit. Es retornen tots els punts agrupats amb subtrajectòries.

Alguns mètodes d'interès:

- ***BuscarPuntsPla()*** :

Busca els centres que hi ha en el pla actual.

- ***BuscarPuntsLinia()*** :

Busca els punts semi centres que conté la línia actual.

- ***comprobarPuntsLinia()*** :

A partir dels punts retornats per la línia actual i l'anterior es van destriant els millors punts per anar determinant els centres del pla.

- ***assignarPuntATrajectòria()*** :

S'uneixen els punts del pla actual amb els punts anteriors per proximitat. Construint pla a pla les subtrajectòries que defineixen els centres del volum.

### *TubActorGenerator*

---

Amb aquesta classe generem un tub a partir d'una trajectòria determinada o un grup de punts depenen del cas que ens interressi.

Alguns mètodes d'interès:

- ***MakeTube()*** :

Es el mètode que genera el tipus d'actor que es requereix a partir de la trajectòria.

### *qtubeTravelExtension*

---

Aquesta classe connecta la interfície del mòdul amb les classes desenvolupades. S'han connectat les parts de la interfície amb la tècnica de signal & slot amb diferents mètodes que

interactuen amb les classes implementades pel mòdul. Interactua activament amb els visors 3D que formen part de l'extensió, degut al viatge virtual a través del budell que requeria el mòdul.

#### [\*qtubeTravelExtension.ui\*](#)

---

És la interfície desenvolupada amb el qtDesigner, conté dos visors 3d per veure el volum renderitzat complet, i per veure una part del volum i viatjar-hi a través seguint la trajectòria. A més conté un conjunt de widgets per que l'usuari interactuï amb l'extensió.



### 3.3.4 Resultats i testeig

#### Resultats dels mètodes de cerca del centre

Per fer la demostració hem utilitzat el mateix tub amb el que hem explicat els mètodes de cerca de centerline per veure que es complien correctament els objectius de cada mètode.

#### Resultats obtinguts amb el mètode simple per centres únics per pla

A continuació mostrem els resultats obtinguts per l'aplicació:

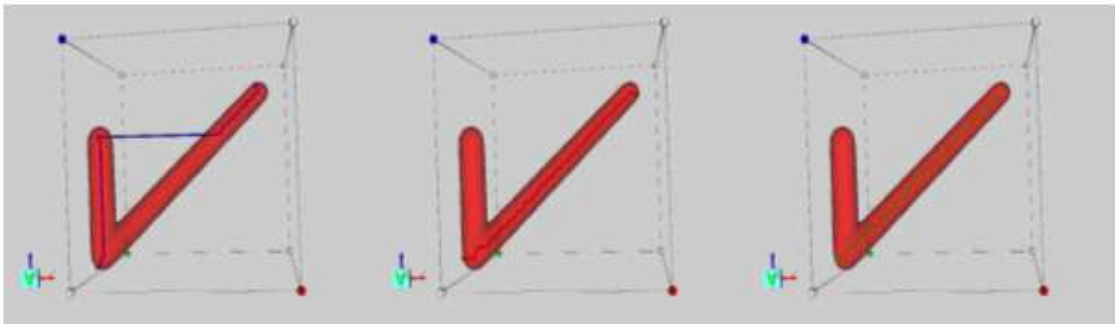


fig. 3.3-44 Resultats obtinguts del amb selecció de centre únic per pla, D'esquerra a dreta trajectòria x, y i z

El resultat de la fusió de les trajectòries obtingudes no és del tot correcta. Segurament el problema es deu a la forma de selecció de la millor trajectòria. El determinar com a millor trajectòria la que retorna mes punts ens dona una trajectòria que recorre zones que no hauria de seleccionar.

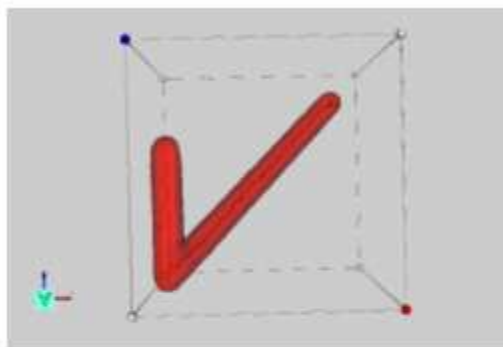


fig. 3.3-45 Fusió de les trajectòries.

#### Resultats obtinguts amb el mètode cerca amb multi-centres per pla

Tot seguit mostrem els resultats obtinguts de l'aplicació per el mètode millorat:

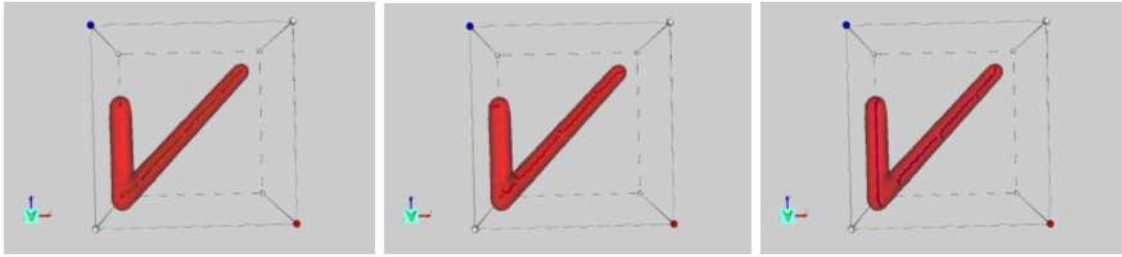


fig. 3.3-46 Subtrajectòries obtingudes per el mètode multicentres per pla

En aquest cas veiem com per exemple en el cas de la trajectòria torbada per l'eix de les Y (blava) pràcticament ella sola ens troba la trajectòria completa. Com em comentat en la implementació es pot donar el cas que tan sols un recorregut per un dels eixos de la imatge ens podria determina el centre del budell. El problema es l'elecció del eix idoni. Per aquest motiu en el segon mètode seguim obtinguem els tres grups de punts per cada eix i els agrupem.

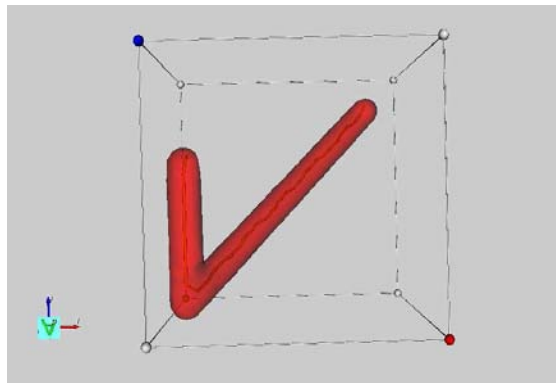


fig. 3.3-47 Trajectòria obtinguda amb mètode multicentre

### *Diferències entre els mètodes de simple i multicentre sobre budells complexes*

La diferència entre els dos mètodes recau principalment en la forma de selecció dels punts que són possibles centres del volum. Mentre que en el primer mètode es busca un sol centre per cada llesca del volum, en el segon es busca tots els centres que hi puguin haver en una mateixa pla de la imatge. A més degut a la forma de selecció dels punts, també varia la forma de tractar les trajectòries a l'hora de connectar-les per formar el centerline final.

Amb les imatges que mostrem a continuació, es poden veure els diferents resultats entre el mètode senzill i multicentre alhora de seleccionar els punts dels centres per cada eix.

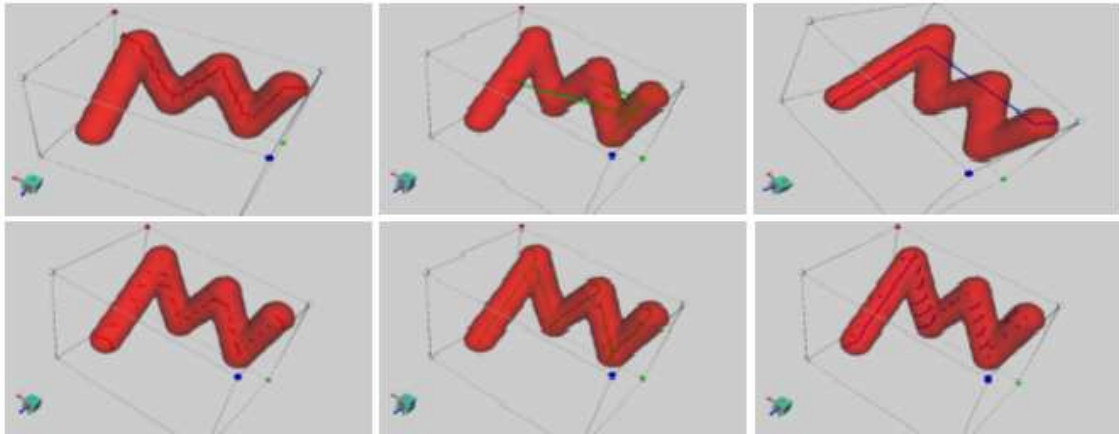


fig. 3.3-48 Diferències entre mètode simple i multicentre. (d'esquerra a dreta trajectòries trobades recorrent els eixos X, Y i Z. I de dalt a baix eixos trobats per el mètode simple i multicentre)

Com podem veure en l'exemple mostrat amb dos casos poden arribar a la solució idònia, però es podria donar el cas que amb el primer mètode no es pogués assolir la solució correcta mentre que amb el mètode de selecció de multicentre sempre obtindrem tots els punts que siguin centre del tub.

Sobre el mètode multicentres per pla, cal destacar que en algun cas les subtrajectòries obtingudes, a partir d'un sol dels tres eixos, podrien determinar casi per sí soles el centre del budell sense necessitat de recórrer a les subtrajectòries obtingudes a partir dels altres eixos. Com seria el cas de les subtrajectòries trobades per l'eix de les Y en el model d'exemple.

Per altra banda, els mètodes també es diferencien en les tècniques utilitzades per ajuntar els punts trobats, però la diferència ve donada per la forma de seleccionar els punts que formen part de la trajectòria. Tal com hem explicat en el primer mètode obtenim només punts, mentre que en el segon obtenim subtrajectòries i per això en fem un tractament diferent.

A continuació mostrem diferents centres trobats amb diferents budells sintètics:

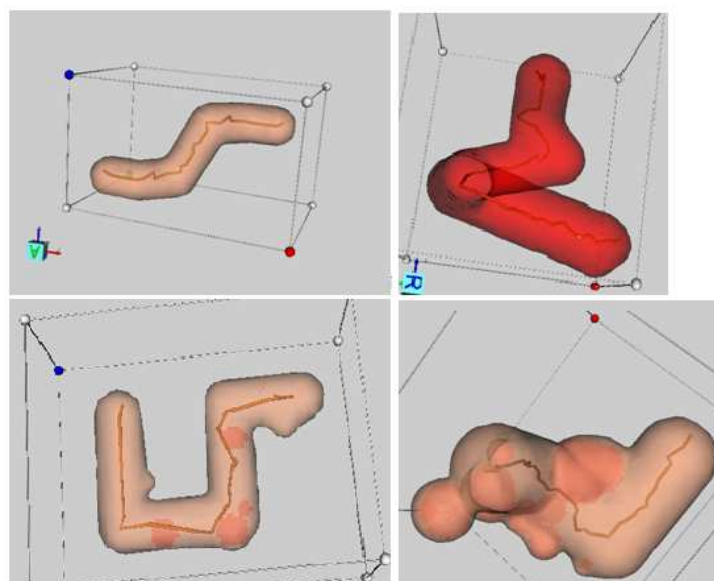


fig. 3.3-49 Centres trobats amb diferents budells

Per testejar la cerca del centre simplement hem comprovat que el resultat obtingut passi realment per dins el budell i que descrigui la forma d'aquest.

**Resultats del sistema visualització i exploració**

Mostrem la interfície del mòdul d'exploració en el moment d'explorar el model sintètic.

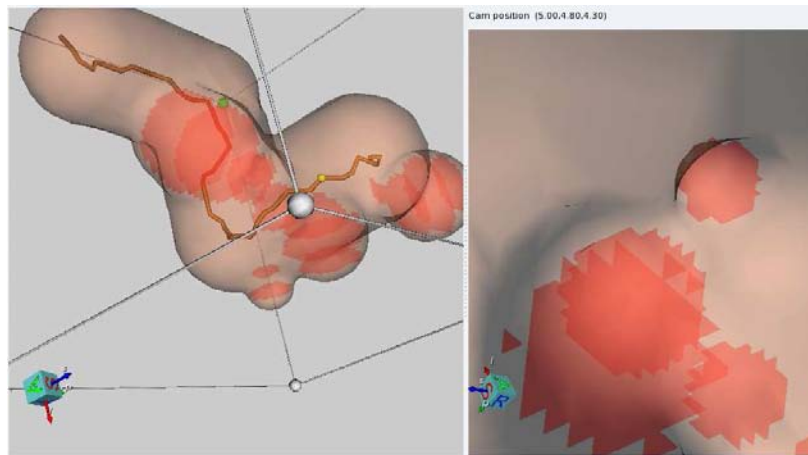


fig. 3.3-50 Exploració del budell; A la imatge de la esquerra es veu el model complet, la trajectòria, i es visualitza amb una esfera la posició de la càmera. A la imatge de la dreta es veu l'interior del budell.

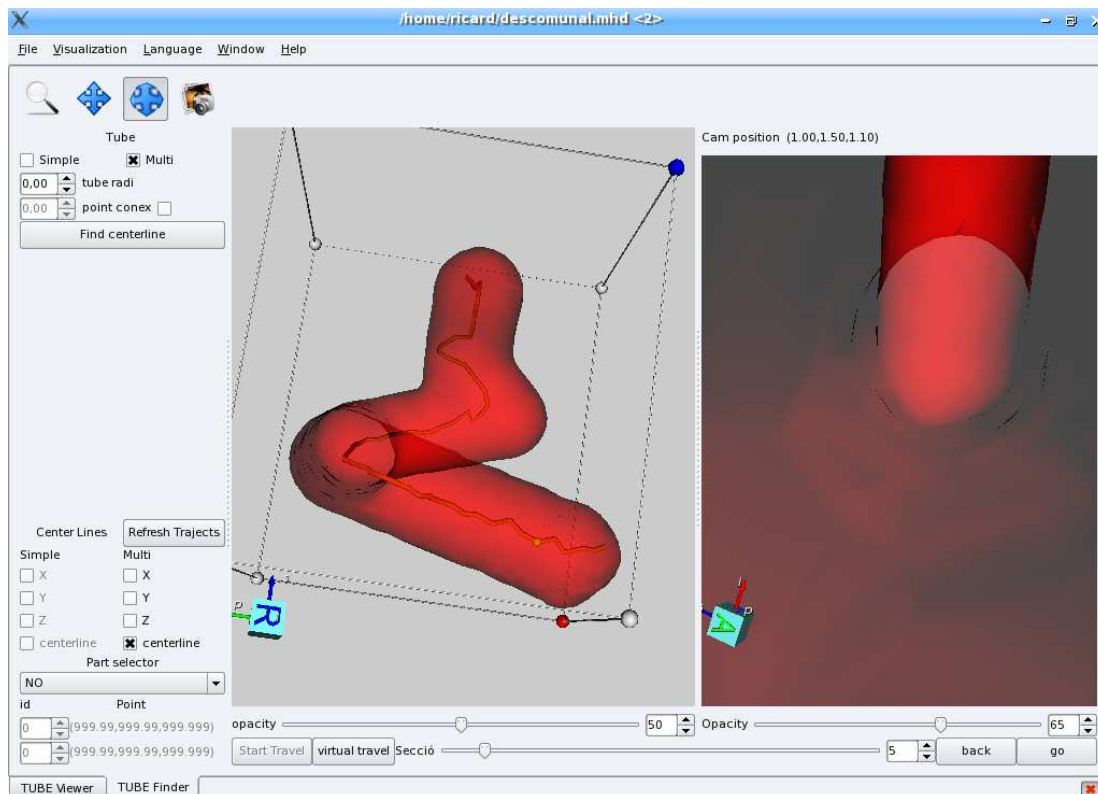


fig. 3.3-51 Exploració del budell 2

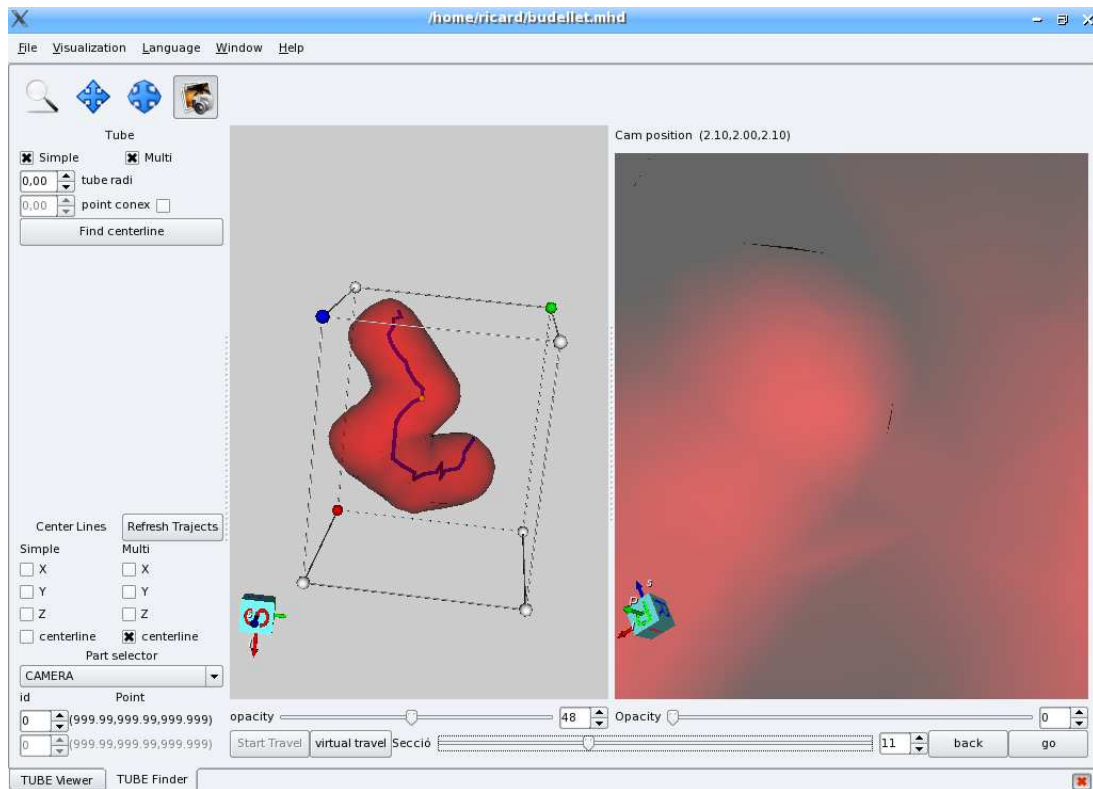


fig. 3.3-52 Exploració del budell 2

El testeig que es fa sobre el sistema d'exploració es senzill, només validem que la càmera passi realment per el centre del budell i segueixi la trajectòria.

### 3.4 Integració de tots els mòduls implementats en la plataforma Starviewer.

Per integrar el mòduls es crearà una Extensió per cada funcionalitat que es precisi. En el nostre cas hem creat dues extensions que permetran a l'usuari interactuar amb els processos de generació de volums, cerca de "centerline" i viatge a través del tub.

Les extensions que s'han implementat son :

*tubeGeneratorExtension.ui* :

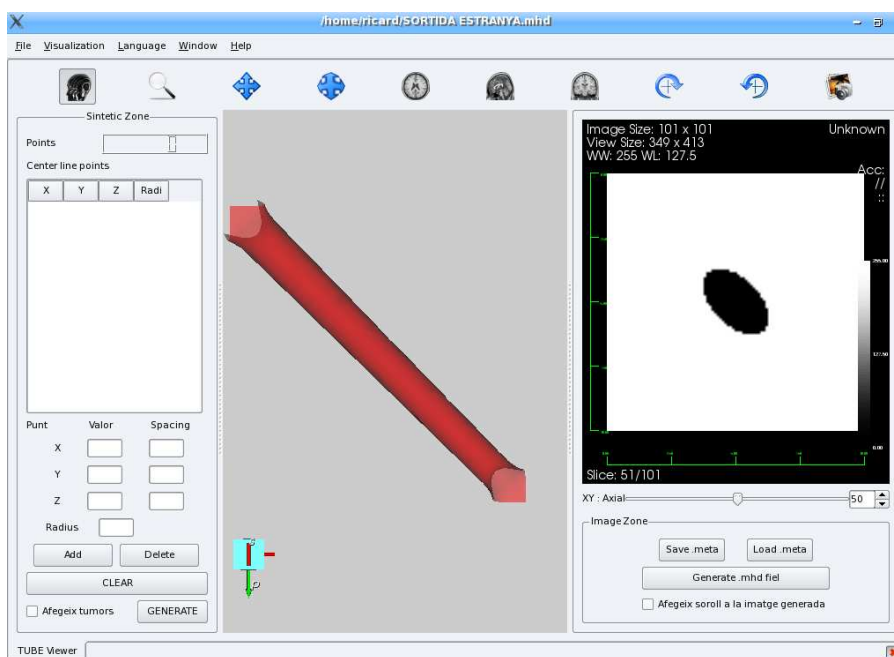


fig. 3.4-1 Panella generador budell

Conté les parts d'interacció amb l'usuari que permeten genera un tub i guardar-lo com a imatge .mhd, o com fitxer pla amb el format .meta.

A cada extensió se li agreguen les funcionalitats (tools) que precisi. D'aquesta forma es podran utilitzar per els diferents visors de la pantalla. En la extensió de generació de models sintètics hem habilitat les següents tools :



ZoomTool: Activa la tool de zoom que utilitzarem al interactuar amb algun dels visors.



TranslateTool: Activa la tool de translació que utilitzarem al interactuar amb algun dels visors. Aquesta tool permetrà moure el visor sobre el pla actual de la càmera en coordenades x-y.



RotateTool: Activa la tool de rotació que utilitzarem al interactuar amb algun dels visors. Aquesta tool només actuarà sobre visors 3D, fent una rotació de la càmera amb centre al punt focal.



AxialTool: Inicialitzar la visió dels visor 2D al centre del silice per les llesques axials.



SagittalTool: Inicialitzar la visió dels visor 2D al centre del silice per les llesques sagitals.



CoaxialTool: Inicialitzar la visió dels visor 2D al centre del silice per les llesques coronal.



SilicingTool: Inicialitza el visor 2D en el centre del silice.



RotateTool : Aplica un rotació sobre els visors 2D.



ScreenShotTool: Permet capturar la imatge que mostra actualment el visor sobre el que s'utilitza la tool i es guarda en un fitxer .bmp.

A part dels visor 3D i 2D de l'interfície i de les tools s'han desenvolupat les següents funcionalitats que responen a les necessitats del mòdul.

#### ▪ **Definició de l'estructura del budell:**

	X	Y	Z	Radi

Punt	Valor	Spacing
X	<input type="text"/>	<input type="text"/>
Y	<input type="text"/>	<input type="text"/>
Z	<input type="text"/>	<input type="text"/>
RADIUS	<input type="text"/>	

fig. 3.4-2 Part interfície per definició estructura budell

Per definir l'estructura del budell hem afegit a l'extensió una llista que visualitza els punts introduïts , i uns quadres de text i botons per afegir-hi nous punts.

Aquesta zona ens permet definir els valors de posició i el radi dels punts que volem, introduint-los en els textbox de la columna "Valor".

Els punts s'afegiran premen el botó

El botó  eliminar l'últim punt de la llista.

I el botó  ens permet buidar completament la llista.

Els valors dels spacings només tindran en compte el primer valor entrat, si no se li dona

cap valor n'assigna uns per defecte.

- **Generar imatge del budell**

Un cop definit el budell per l'usuari s'haurà de generar el model sintètic. La interfície permet a l'usuari definir si el budell ha de tenir tumors o no, marcant o desmarcant respectivament el checkbox  Afegix tumors .

Un cop s'han definit tots els punts es pot generar el budell premen el botó  .



### TubTravelExtensió.ui

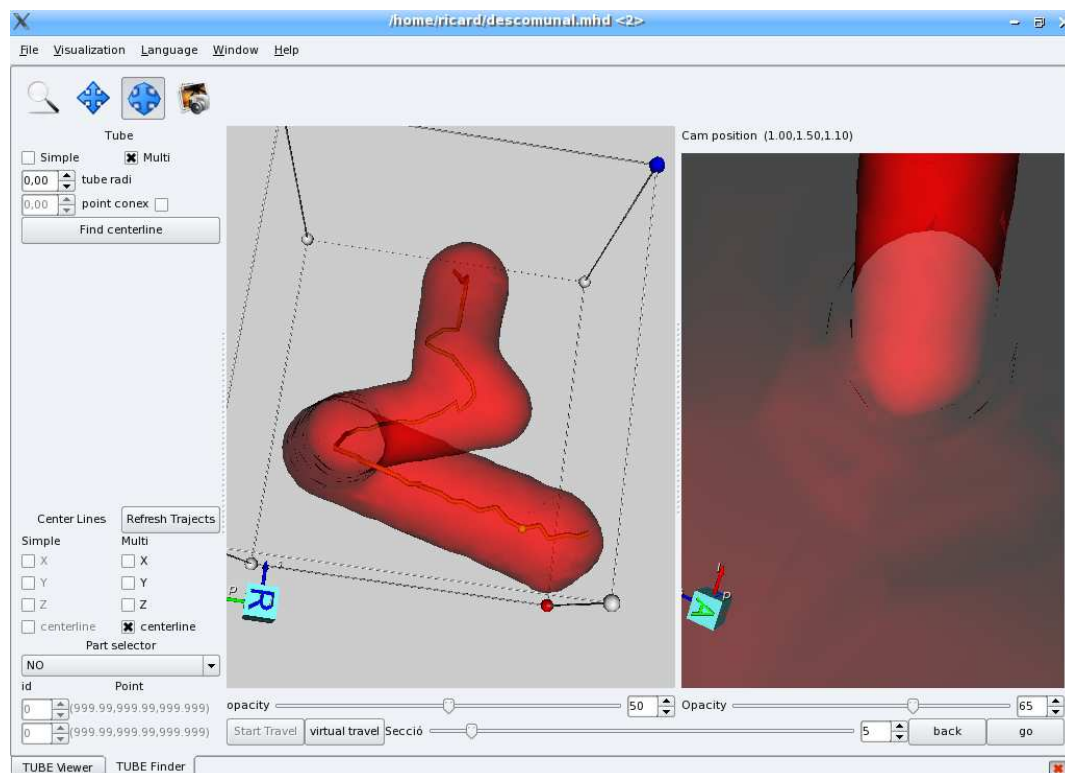


fig. 3.4-3 Interfície sistema exploració del budell

Permet segmentar un imatge que conté un colon (sintètic o real) i a partir d'aquesta obté la trajectòria i en permet fer un recorregut.

Selecció de parts del tub i veient-les en un visor més petit que mostrarà la part actual del "trajecte" mentre en l'altre es veu el colon/tub complet amb una esfera que indica la posició de la càmera actual.

A cada extensió se li agreguen les funcionalitats, tools que precisi. D'aquesta forma es podran utilitzar per els diferents visors de la pantalla. En la extensió de generació de models sintètics hem habilitat les següents tools aplicables als visors 3D:



**ZoomTool:** Activa la tool de zoom que utilitzarem al interactuar amb algun dels visors.



**TranslateTool:** Activa la tool de translació que utilitzarem al interactuar amb algun dels visors. Aquesta tool permetrà moure el visor sobre el pla actual de la càmera en coordenades x-y.



**RotateTool:** Activa la tool de rotació que utilitzarem al interactuar amb algun dels visors. Aquesta tool només actuarà sobre visors 3D, fent una rotació de amb centre al punt focal.



**ScreenShotTool:** Permet capturar la imatge que mostra actualment el visor sobre el que s'utilitza la tool i es guarda en un fitxer .bmp.

En aquesta extensió s'han desenvolupat diferents funcions que descrivim a continuació:

- **Definir paràmetres de cerca de centerline:**

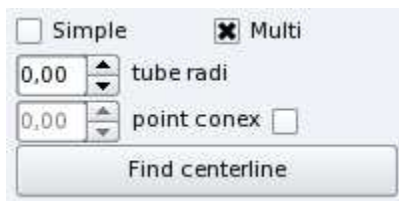



fig. 3.4-4 Part interfície paràmetres cerca centreline

L'usuari podrà escollir quins dels mètodes s'utilitzaran amb els checkbox "Simple" i "Multi". Si tria els dos s'executaran tots dos i s'obindrà el resultat dels dos mètode.

També podem definir l'amplada del radi dels tubs generats per mostrar la trajectòria amb l'spinbox "tube radi".

De la mateixa forma podem definir el factor de connexió dels punts a l'hora de buscar el centre del budell amb el spinbox "point conex". Finalment un cop definitos tots els paràmetres executem el mètode de cerca de centerline premen el botó .

- **Mostrar/ocultar les diferents trajectòries obtingudes:**

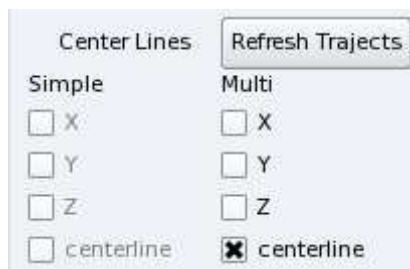


fig. 3.4-5 Part mostrar/ocultar trajectòries

Quan ja s'han trobat les trajectòries l'usuari per mitjà dels checkbox de la imatge 3.4-5 podrà mostrar o ocultar les trajectòries que tinguin els checkbox activats. Les columnes Simple i Multi separen els resultats del dos mètodes. Les trajectòries finals són les "centerline" les altres (x, y i z) són les trajectòries trobades al seleccionar els punts per els diferents eixos.

- **Activar/desactiva el selector de punts/rangs**

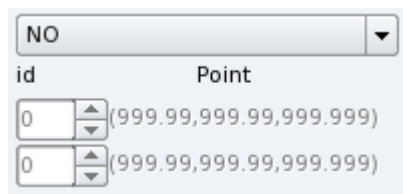


fig. 3.4-6 Activar/Desactivar point selector

La funcionalitat permetrà activar o desactivar la funció de selecció de punts de la imatge. Al activar aquesta funció la trajectòria es visualitzarà en forma de grup de punts, i en podem seleccionar la posició que es mostrar en les etiquetes de sota el desplegable.

Aquesta opció ha estat pensada per en un futur poder guardar zones d'interès del budell. A partir de punts escollits sobre la imatge. Podrem guardar un rang o un punt en concret. També es podrà guardar la posició actual de la càmera. Totes aquestes opcions de selecció es trien des de el desplegable com mostrem a la imatge següent.

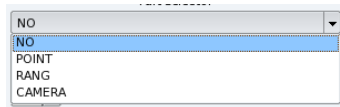


fig. 3.4-7 Desplegable per escollir opcions de pointset

En la figura 3.4-7 podem veure com queda la trajectòria de la imatge al transformar-la en un conjunt de punts.

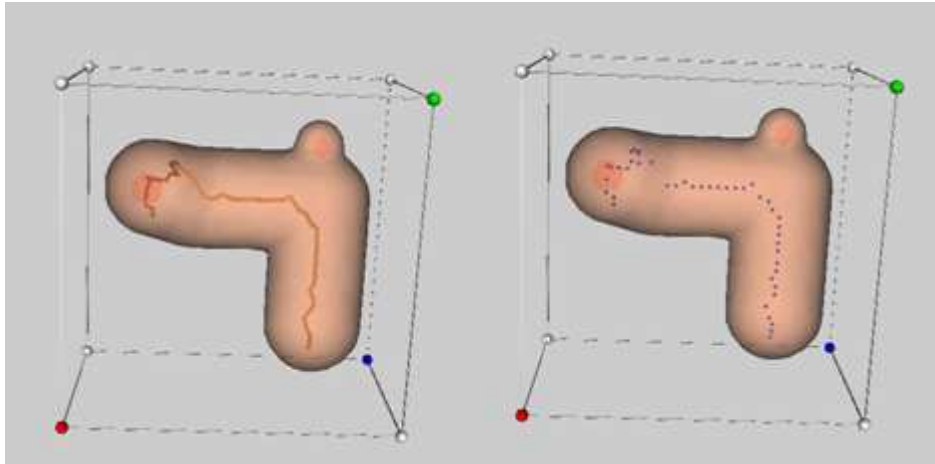


fig. 3.4-8 Comparació entre trajectòria i pointset per selecció de rangs o punts

#### ▪ **Exploració del budell**

Per interactuar amb l'exploració del budell s'han afegit els següents widgets que permeten avançar o retrocedir per la trajectòria trobada.

El botons   inicialitzen les variables d'exploració i preparen els visors per l'exploració.

Amb els botons   ens serveixen per avançar o retrocedir sobre la trajectòria. El *silice* "Secció" ens permet igualment viatjar a través de la trajectòria. Els canvis de posició queden reflectits en els visors de l'extensió.



Els *silice* "opacity" serveixen per variar l'opacitat de les parets dels budells en cada visor.

### 3.5 Diagrama de classes final

Amb tot el que s'ha implementat hem modificat el diagrama de classes inicial. S'han desenvolupat tots els mòduls previstos per el projecte. A continuació mostrem el diagrama final amb les classe principals implementades i les que s'han modificat o han tingut més rellevància a l'hora de treballar amb la plataforma "Starviewer".

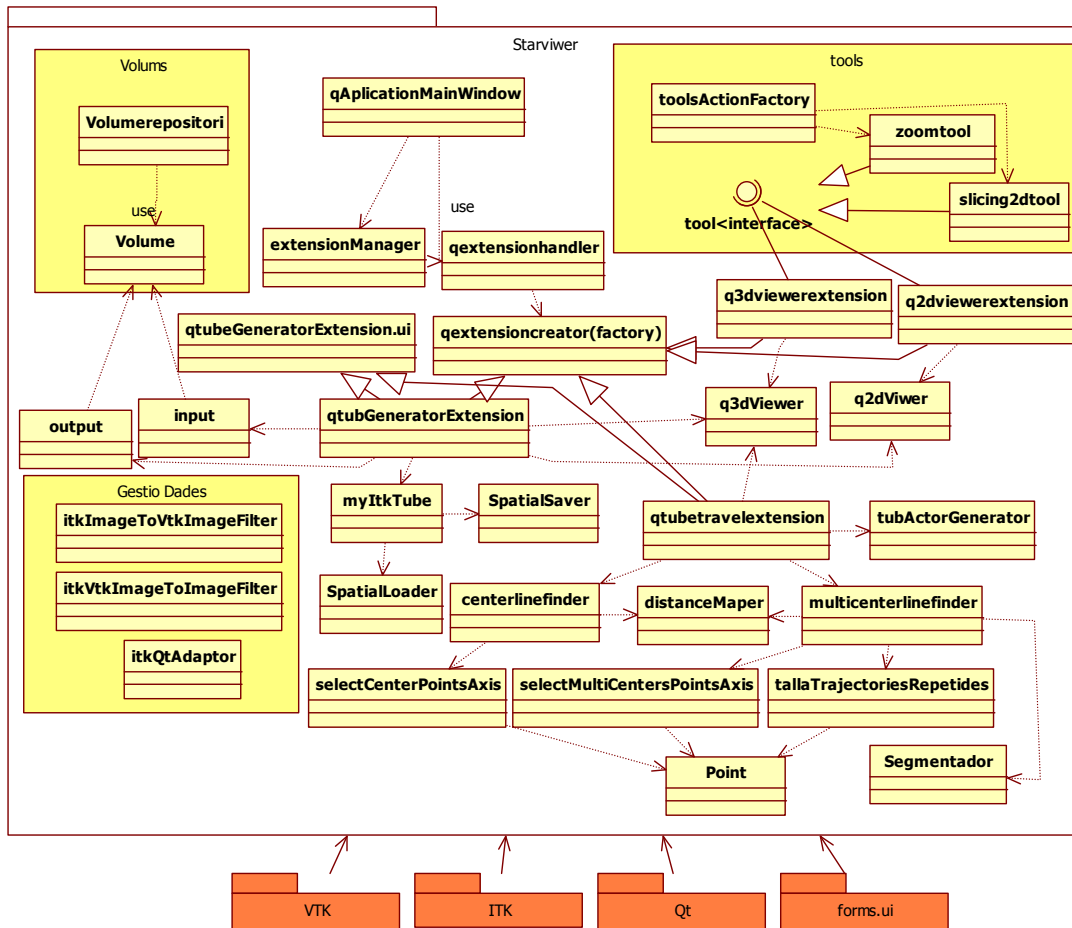


fig. 3.5-1 Diagrama final de classes

## 4. Conclusions i treball futur

En aquest projecte ens hem centrat en el diagnòstic del càncer colorectal i el desenvolupament de mètodes i tècniques de suport al seu diagnòstic. El nostre objectiu principal era desenvolupar eines informàtiques que s'integressin en la plataforma Starviewer per tal de facilitar el procés de diagnòstic d'aquestes patologies.

Podem considerar que aquest objectiu s'ha assolit ja que hem implementat:

- ***Un sistema d'exploració del volum del budell a través d'una trajectòria trobada.***
- ***Un model de visualització que permet veure tridimensionalment els tumors sobre el budell.***

Però les dues funcionalitats només s'han pogut provar sobre models sintètics, per aquest motiu ens hem hagut de desenvolupar una eina de generació de models per poder testejar els resultats de les funcionalitats implementades.

Tot i que els resultats obtinguts que han estat acceptables es necessiten aplicar algunes millores als mètodes per millorar-ne la eficiència, que es deixen com a treball futur. A continuació expliquem les millores proposades sobre el mòdul implementats.

### 4.1 Millores sobre el generador de models sintètics

Descriurem alguns punts que serien interessants per millorar la definició de budells sintètics, sense perdre de vista que aquest mòdul només té l'objectiu de generar models de budells i no de cap altre objecte. Les millores aportades anirien encarades a millorar la interacció del usuari amb la interfície.

- Mostrar a l'usuari visualment la part del budell definit, seria una tasca força senzilla utilitzant les eines de VTK.
- Permetre fer modificacions interactuant amb el visor 3D, aquesta tècnica seria una mica més complexa, però de gran ajuda a l'hora de definir models sintètics.

### 4.2 Millores sobre els mètodes de cerca del centre

#### 4.2.1 Millorant la cerca del centre

Per millorar els mètodes de cerca del centre hauríem d'estudiar detingudament, el mètode de selecció dels punts per veure si en poguéssim fer alguna millora per poder descartar punts que a l'hora de reconstruir el budell fan desviar la trajectòria.

D'altra banda s'hauria d'utilitzar ,en el backtracking, un altre mètode per seleccionar la millor solució del conjunt de solucions donats. Posarem alguns exemples que podrien donar bon resultat:

- Seleccionar la millor trajectòria a partir del valor de nivell de girs dels punts.
- Seleccionant el que aporti mínim nivell de girs i a mes mínima distància des de inicial a final.
- El que tingui menys variació d'angles entre els punts seleccionats.
- O combinacions entre les esmentades.

A més es podrien valorar algunes tècniques que permetessin fer podes sobre l'algoritme de backtracking. Per aquesta millora s'hauria de portar a terme un estudi mes complex tant del volum, com del mètode d'elecció de la pròxima solució parcial per cada node. Millorant aquests dos punts no caldria buscar totes les solucions sinó les més prometedores, augmentant considerablement l'eficiència.

#### **4.2.2 Altres mètodes per cercar el centre del budell.**

Aquestes tècnica son conegudes com l'extracció de l'esquelet d'un objecte, que en el nostre cas seria el budell.

Per obtenir l'skeleton d'un objecte hi han diferents tècniques:

- *Distance map*, la tècnica que hem utilitzat nosaltres. Però per millorar-ho es podria aplicar algun algoritme de grafs com per exemple amb l'algoritme de camins mínims de Dijkstra computant el cost amb el nivell de gris del punt.
- "Voronoi diagram" : separant en cel·les independents cada punt que forma l'objecte al que s'extreu l'skeleton que es va formant amb les arestes de les cel·les que es toquen.
- "Thinning": Utilitza tècniques de erosió de volums, per anar-se aproximant esquelet de l'objecte.

### **4.3 Milliores sobre el sistema d'exploració del budell**

Seria interessant dotar el mòdul d'exploració del budell de tècniques de selecció de volums d'interès. Permetre a l'usuari que seleccioni un tumor , o el budell i pugui fer-lo mes menys visible o canviar de color, fer-ne càlculs de volum, entre d'altres funcionalitats que siguin d'interès per els especialistes.

## 5. Valoració personal

Aquest projecte m'ha ajudat a conèixer formes d'aplicar els conceptes que he après a la carrera, i reaprès durant el desenvolupament del projecte. Treballar en una aplicació d'aquestes magnituds permet aplicar tècniques i conceptes que probablement no hauria utilitzat en l'entorn professional, almenys immediatament.

El món gràfic m'ha cridat sempre l'atenció. És un tipus de programació que m'omple i diverteix, per aquest motiu vaig voler participar en el projecte de l'Starviewer. Entrar a formar part d'un producte d'aquestes dimensions i en una aplicació que és realment útil i aplicable em va il·lusionar. He gaudit molt fent aquest projecte, l'aplicació d'algoritmes i tècniques sobre imatge com segmentació, mapa de distàncies per aconseguir-ne informació, la visualització de volums i moltes altres tècniques i estudis aplicables a aquests tipus de projectes són, indiscutiblement, moltes hores de feina però que aporten, i m'han aportat, una satisfacció personal enorme. Sobretot a l'hora de comprovar els resultats i veient com de mica en mica anava assolint els propòsits inicials. No puc negar que en el meu cas ha estat un procés laboriós. Treballar i fer el projecte a la vegada pot ser complicat (tot i que sempre hi ha casos excepcionals i admirables) i ha allargat considerablement la durada del projecte.

Gràcies a la realització del projecte he conegut eines i tecnologies que segurament utilitzaré després de l'entrega del mateix. Sempre és interessant conèixer noves eines i entorns de programació. Tothom sap que el nostre és un "món" que experimenta un constant canvi i la adaptabilitat és imprescindible, però curiosament no tothom sap que no TOTS els informàtics ho podem saber TOT, malauradament es un petit preu que paguem perquè la nostra feina és avui en dia aplicable a totes els àmbits, ja no només en els professionals sinó també quotidians. La magnitud de la informàtica és per nosaltres, els informàtics (vulgarment coneguts per frikies), una característica alentadora ja que esperem no tenir problemes laborals tot i aquest període de "NO CRISIS" que estem vivint (any 2008).

Vull agrair, a tu, que hagi arribat a llegir la conclusió, perquè suposo que hauràs llegit la resta del document. Per tant ens coneixem, i te demanat amigablement que et llegeixis tot el document, ets un dels afortunats del jurat de setembre, o estàs utilitzant el document per fer la teva pròpia documentació pel que et felicito per haver acabat el projecte. Ja pots respirar fons i gaudir de la vida.

Per acabar voldria agrair a tota la gent que m'ha ajudat en algun moment del projecte. Especialment a en Sergi que m'ha ajudat sempre que ha fet falta i a la Montse pel seu recolzament durant tot el projecte. A tots dos moltes gràcies.

## **6. Bibliografia**



## 6.1 Llibres

- Fco. Javier Ceballos; Enciclopedia del lenguaje C++; Ra-Ma 1a edició, 2003.
- L. Ibáñez, W. Schroeder, L. Ng, J. Cates; The ITK Software Guide vs 2.4; Kitware Inc, 2005.
- W. Schroeder, K. Martin, B. Lorensen; The VTK User's Guide; Kitwar Inc ,2003.
- Trolltech; Qt 3.3 Whitepaper; [www.trolltech.com](http://www.trolltech.com).

## 6.2 Articles

- [www.linux-magazine.es/issue/06/PythonVTK.pdf](http://www.linux-magazine.es/issue/06/PythonVTK.pdf)
- [www.radiolegsdecatalunya.cat/formacio/resums/GB35BA10\\_R.pdf](http://www.radiolegsdecatalunya.cat/formacio/resums/GB35BA10_R.pdf)
- [public.kitware.com/OpenSourceSoftwarePractice/images/c/c9/VTKOverview.pdf](http://public.kitware.com/OpenSourceSoftwarePractice/images/c/c9/VTKOverview.pdf)
- [www.vialab.org/methods\\_course/Lectures/itk\\_lecture8\\_neighborhoods.ppt](http://www.vialab.org/methods_course/Lectures/itk_lecture8_neighborhoods.ppt)
- [paper.ijcsns.org/07\\_book/200707/20070729.pdf](http://paper.ijcsns.org/07_book/200707/20070729.pdf)

## 6.3 Referències electròniques

- Pàgina web de l'estàndard DICOM: [DICOM.nema.org](http://DICOM.nema.org)
- Pàgina web de Doxygen: [www.doxygen.org](http://www.doxygen.org)
- Pàgina web de KDE Development Environment: [www.kdevelop.org](http://www.kdevelop.org)
- Pàgina web de Trolltech (Qt): [www.trolltech.com](http://www.trolltech.com)
- Pàgina web de Visualization Toolkit: [www.vtk.org](http://www.vtk.org)
- Pàgina web de Insight Sementation & Registration Toolkit: [www.itk.org](http://www.itk.org)
- Pàgina web de StarUML: [staruml.sourceforge.net](http://staruml.sourceforge.net)
- Pàgines webs d'extracció de skeleton :
  - [math.lbl.gov/~deschamp/phdthesis/node48.html](http://math.lbl.gov/~deschamp/phdthesis/node48.html)
  - [groups.csail.mit.edu/graphics/classes/6.838/S98/meetings/m25/mm.html](http://groups.csail.mit.edu/graphics/classes/6.838/S98/meetings/m25/mm.html)
  - [www.inf.u-szeged.hu/~palagyi/skel/skel.html](http://www.inf.u-szeged.hu/~palagyi/skel/skel.html)
- Pàgines mèdiques:
  - [www.medicorimaging.com/universal/default.htm](http://www.medicorimaging.com/universal/default.htm)
  - [www.elmundo.es/elmundosalud/especiales/cancer/colon5.html](http://www.elmundo.es/elmundosalud/especiales/cancer/colon5.html)
  - [www.taceimagenes.com](http://www.taceimagenes.com)
  - [www.portalesmedicos.com](http://www.portalesmedicos.com)
  - [www.centroemergencias.com](http://www.centroemergencias.com)
  - [www.investigacionesmedicas.com](http://www.investigacionesmedicas.com)

## 7. Anexes

### 7.1 Contingut de fitxer de meta dades de spatial objects

```

ObjectType = Scene
NDims = 3
NObjects = 11
ObjectType = Group
NDims = 3
ID = 10
Color = 1 0 0 1
TransformMatrix = 1 0 0 0 1 0 0
0 1
Offset = 0 0 0
CenterOfRotation = 0 0 0
ElementSpacing = 1 1 1
EndGroup =
ObjectType = Ellipse
NDims = 3
Name = EllipseX 1
ID = 1
ParentID = 10
TransformMatrix = 1 0 0 0 1 0 0
0 1
Offset = 1 1 1
CenterOfRotation = 0 0 0
ElementSpacing = 1 1 1
Radius = 1 1 1
ObjectType = Ellipse
NDims = 3
Name = Ellipse 2
ID = 2
ParentID = 10
TransformMatrix = 1 0 0 0 1 0 0
0 1
Offset = 1.5 1.5 1.5
CenterOfRotation = 0 0 0
ElementSpacing = 1 1 1
Radius = 1 1 1
ObjectType = Ellipse
NDims = 3
Name = Ellipse 3
ID = 3
ParentID = 10
TransformMatrix = 1 0 0 0 1 0 0
0 1
Offset = 2 2 2
CenterOfRotation = 0 0 0
ElementSpacing = 1 1 1
Radius = 1 1 1
ObjectType = Ellipse
NDims = 3
Name = Tumor 4
ID = 4
ParentID = 3
TransformMatrix = 1 0 0 0 1 0 0
0 1
Offset = 3.156 2.156 2.156
CenterOfRotation = 0 0 0
ElementSpacing = 1 1 1
Radius = 0.479 0.479 0.479
ObjectType = Ellipse
NDims = 3
Name = Ellipse 5
ID = 5
ParentID = 10
TransformMatrix = 1 0 0 0 1 0 0 0 1
Offset = 2.5 2.5 2.5
CenterOfRotation = 0 0 0
ElementSpacing = 1 1 1
Radius = 1 1 1
ObjectType = Ellipse
NDims = 3
Name = Ellipse 6
ID = 6
ParentID = 10
TransformMatrix = 1 0 0 0 1 0 0 0 1
Offset = 3 3 3
CenterOfRotation = 0 0 0
ElementSpacing = 1 1 1
Radius = 1 1 1
ObjectType = Ellipse
NDims = 3
Name = Tumor 7
ID = 7
ParentID = 6
TransformMatrix = 1 0 0 0 1 0 0 0 1
Offset = 4.156 3.156 3.156
CenterOfRotation = 0 0 0
ElementSpacing = 1 1 1
Radius = 0.479 0.479 0.479
ObjectType = Ellipse
NDims = 3
Name = Ellipse 8
ID = 8
ParentID = 10
TransformMatrix = 1 0 0 0 1 0 0 0 1
Offset = 3.5 3.5 3.5
CenterOfRotation = 0 0 0
ElementSpacing = 1 1 1
Radius = 1 1 1
ObjectType = Ellipse
NDims = 3
Name = EllipseX 9
ID = 9
ParentID = 10
TransformMatrix = 1 0 0 0 1 0 0 0 1
Offset = 4 4 4
CenterOfRotation = 0 0 0
ElementSpacing = 1 1 1
Radius = 1 1 1
ObjectType = Ellipse
NDims = 3
Name = Tumor 10
ID = 10
ParentID = 9
TransformMatrix = 1 0 0 0 1 0 0 0 1
Offset = 5.156 4.156 4.156
CenterOfRotation = 0 0 0
ElementSpacing = 1 1 1
Radius = 0.479 0.479 0.479

```

## 8. Índex d'imatges

fig. 2.1-1 Anatomia del colon .....	16
fig. 2.1-2 Fases dels pòlips .....	17
fig. 2.1-3 Pòlips en el colon.....	17
fig. 2.1-4 Colonoscòpia .....	18
fig. 2.1-5 Sigmoidoscòpia flexible .....	18
fig. 2.1-6 Enema Opac.....	18
fig. 2.1-7 Tacte rectal .....	19
fig. 2.1-8 Ecografia tansrectal .....	19
fig. 2.1-9 Colonoscòpia virtual.....	19
fig. 2.2-1 Pipline del procés de visualització .....	21
fig. 2.3-1 Obtenció de llesques del tac.....	23
fig. 2.3-2 Procés de captació de dades per CT.....	24
fig. 2.3-3 Tac pèlvic.....	24
fig. 2.3-4 Exemple fitxer capçalera .mhd.....	25
fig. 2.3-5 Exemple de fitxer de dades d'un DICOM.....	26
fig. 2.4-1 Representació de vòxel .....	27
fig. 2.4-2 Plans de representació del model.....	27
fig. 2.4-3 Representació volumètrica de les dades .....	27
fig. 2.5-1 Exemple reconstrucció 3D .....	29
fig. 2.6-1 Visualització de volums i Visualització de superfícies.....	30
fig. 2.8-1 Diagrama inicial de classes Starviewer.....	39
fig. 2.8-2 Interfície Starviewer .....	40
fig. 2.8-3 : Barra de Tools .....	40
fig. 2.8-4 : Pestanyes, permeten canvia d'extensió.....	40
fig. 3.1-1 Representació del model sintètic .....	43
fig. 3.1-2 Exemple de reconstrucció d'un tub a partir els pans de tall.....	44
fig. 3.1-3 Informació introduïda per l'usuari a l'hora de definir un budell sintètic.....	46
fig. 3.1-4 Representació de dos punts entrats per l'usuari amb radis R1 i R2 .....	47
fig. 3.1-5 Càlcul dels punts intermedis a partir del radi .....	47
fig. 3.1-6 Generació d'esferes intermèdies amb increment de radi.....	48
fig. 3.1-7 Zona sobre la que es pot generar un tumor.....	48
fig. 3.1-8 Càlcul del radi mínim del tumor respecte la posició.....	49
fig. 3.1-9 Generació model sintètic .....	50
fig. 3.1-10 Segmentació budell i tumors.....	50
fig. 3.1-11 Imatge obtinguda final i imatge obtinguda de la conversió dels spatial objects .....	51

fig. 3.1-12 Definició d'un spatialObject contingut arxiu .meta.....	52
fig. 3.1-13 Diagrama complet de casos d'ús de generació de model sintètic .....	53
fig. 3.1-14 Diagrama cas d'us Definir Budell .....	54
fig. 3.1-15 Diagrama de seqüència "Definir Budell" .....	55
fig. 3.1-16 Diagrama seqüència afegir punt al budell.....	56
fig. 3.1-17 Diagrama seqüència "Modificar punt" .....	57
fig. 3.1-18 Diagrama de seqüència "Eliminar Punt Budell" .....	58
fig. 3.1-19 Diagrama de casos d'ús generació del budell.....	59
fig. 3.1-20 Diagrama de seqüència "Generar Budell" .....	60
fig. 3.1-21 Diagrama seqüència "Definir Punts Intermedis" .....	62
fig. 3.1-22 Diagrama seqüència "Definir tumor" .....	63
fig. 3.1-23 Diagrama de seqüència "Generar Volum" .....	64
fig. 3.1-24 Cas d'ús "Guardar/Carregar el model sintètic" .....	65
fig. 3.1-25 Diagrama seqüència "Guardar Budell".....	66
fig. 3.1-26 Diagrama de Seqüència "Guardar fitxer Meta" .....	67
fig. 3.1-27 Diagrama seqüència "Guarda fitxer mhd" .....	68
fig. 3.1-28 Diagrama de seqüència "Carregar fitxer .meta" .....	69
fig. 3.1-29 Diagrama de classes que intervenen en la generació del model sintètic.....	70
fig. 3.1-30 Diferents tubs generats a partir del mòdul implementat.....	73
fig. 3.2-1 Seccions de plans sagital axial i coaxial Resultats tac .....	76
fig. 3.2-2 Segmentació de colon renderitzat en 3D .....	76
fig. 3.2-3 Diagrama de cas d'ús Segmentació .....	77
fig. 3.2-4 Diagrama seqüència "Segmentar Volum" .....	78
fig. 3.2-5 Segmentació sobre un volum generat per el generador .....	79
fig. 3.2-6 Segmentació del budell i els tumors sobre un budell sintètic.....	79
fig. 3.2-7 Visualització sobre el model sintètic, segmentació del model sintètic sense els tumors i visualització de la segmentació conjunta dels tumors i el budell. ....	80
fig. 3.2-8 Resultats de la segmentació comparats amb el model inicial.....	80
fig. 3.3-1 Simulació pla de tall de la secció d'un cervell .....	81
fig. 3.3-2 Tac pèlvic.....	82
fig. 3.3-3 El volum pot contenir més d'una secció del budell en un mateix pla .....	82
fig. 3.3-4 Mapa de distàncies sobre una secció del budell.....	83
fig. 3.3-5 Esquema generat per un backtracking .....	84
fig. 3.3-6 Subtrajectòries obtingudes a partir del recorregut per els exios x, y i z .....	85
fig. 3.3-7 Trajectòries trobades sobre el tub .....	86
fig. 3.3-8 Recorregut del volum.....	87
fig. 3.3-9 Recorregut per cercar més d'un centre sobre la llesca actual .....	88
fig. 3.3-10 Tipus de possibles centres retornats per una trajectòria .....	88

---

fig. 3.3-11 Exemple del tractament dels punts retornats per un pla.....	89
fig. 3.3-12 D'esquerra a dreta punts aconseguits per mètode multicentre i punts aconseguits pel mètode simple, tots dos sobre el mateix eix.....	89
fig. 3.3-13 Subtrajectòries trobades per el mètode multicentre.....	90
fig. 3.3-14 El sentit de la trajectòria no implica que sigui diferent.....	90
fig. 3.3-15 Eliminació de parts de trajectories repetides.....	91
fig. 3.3-16 Angles formats pels punts de la trajectòria.....	92
fig. 3.3-17 Imatge obtinguda mitjançant mètode de conturing.....	93
fig. 3.3-18 Imatge obtinguda amb isosurface.....	93
fig. 3.3-19 Visualització texture 2D.....	94
fig. 3.3-20 Visualització Texture 3D.....	94
fig. 3.3-21 Ray-casting.....	94
fig. 3.3-22 Imatge aconseguida per ray-casting.....	95
fig. 3.3-23 Atributs de la càmera.....	96
fig. 3.3-24 Moviments de la càmera centrats en el focal point , i moviments de la càmera centrats en la seva pròpia posició.....	96
fig. 3.3-25 Posició càmera sobre la trajectòria.....	97
fig. 3.3-26 Diagrama cas d'ús d'exploració del budell.....	98
fig. 3.3-27 Diagrama cas d'ús "Cercar trajectòria".....	99
fig. 3.3-28 Diagrama de seqüència "Selecció mètode multicentre".....	100
fig. 3.3-29 Diagrama seqüència "Selecció trajectòria simple".....	101
fig. 3.3-30 Diagrama de seqüència "Canviar distàncies connectivitat punts".....	102
fig. 3.3-31 Diagrama de seqüència "Cercar centerline".....	103
fig. 3.3-32 Diagrama de seqüència "Cerca centerline simple".....	105
fig. 3.3-33 Diagrama de seqüència "Cerca centerline amb multicentres per pal".....	107
fig. 3.3-34 Diagrama visualització.....	108
fig. 3.3-35 Diagrama de seqüència Raycasting + Conturing.....	109
fig. 3.3-36 Diagrama de casos d'ús "Viatjar pel centre del budell".....	110
fig. 3.3-37 Diagrama de seqüència "Inicialitzar funcionalitats trajectòria".....	111
fig. 3.3-38 Diagrama de seqüència "Moure's per la trajectòria".....	112
fig. 3.3-39 Diagrama cas d'ús visualització trajectòries.....	113
fig. 3.3-40 Diagrama seqüència mostrar/ocultar trajectòries.....	114
fig. 3.3-41 Diagrama seqüència canviar radi trajectòries.....	115
fig. 3.3-42 Diagrama de Seqüència "Mostrar trajectòria".....	116
fig. 3.3-43 Diagrama seqüència ocultar trajectòria.....	117
fig. 3.3-44 Resultats obtinguts del amb selecció de centre únic per pla, D'esquerra a dreta trajectòria x, y i z.....	123
fig. 3.3-45 Fusió de les trajectòries.....	123

fig. 3.3-46 Subtrajectòries obtingudes per el mètode multicentres per pla .....	124
fig. 3.3-47 Trajectòria obtinguda amb mètode multicentre.....	124
fig. 3.3-48 Diferències entre mètode simple i multicentre. (d'esquerra a dreta trajectòries trobades recorrent els eixos X, Y i Z. I de dalt a baix eixos trobats per el mètode simple i multicentre) .....	125
fig. 3.3-49 Centres trobats amb diferents budells .....	125
fig. 3.3-50 Exploració del budell; A la imatge de la esquerra es veu el model complet, la trajectòria, i es visualitza amb una esfera la posició de la càmera. A la imatge de la dreta es veu l'interior del budell. ....	126
fig. 3.3-51 Exploració del budell 2.....	126
fig. 3.3-52 Exploració del budell 2.....	127
fig. 3.4-1 Panalla generador budell .....	128
fig. 3.4-2 Part interfície per definició estructura budell .....	129
fig. 3.4-3 Interfície sistema exploració del budell .....	131
fig. 3.4-4 Part interfície paràmetres cerca centreline .....	132
fig. 3.4-5 Part mostrar/ocultar trajectòries.....	132
fig. 3.4-6 Activar/Desactivar point selector.....	132
fig. 3.4-7 Desplegable per escollir opcions de pointset.....	133
fig. 3.4-8 Comparació entre trajectòria i pointset per selecció de rangs o punts.....	133
fig. 3.5-1 Diagrama final de classes.....	134