

*Sistema informàtic de gestió, processament
i anàlisi d'imatges confocals per l'ajuda
en la diagnosi del càncer de pell.*

Jaume Delgado Fernandez
Enginyeria Tècnica en Informàtica de Sistemes.

Universitat de Girona, 2006

ÍNDEX

1	<u>DESCRIPCIÓ DEL PROJECTE</u>	5
1.1	<u>MOTIVACIÓ</u>	5
1.2	<u>OBJECTIUS DEL PROJECTE</u>	5
1.3	<u>EINES DE DESENVOLUPAMENT</u>	7
1.4	<u>PLANIFICACIÓ</u>	8
2	<u>MARC DE TREBALL</u>	13
2.1	<u>CÀNCER DE PELL</u>	13
2.1.1	<u>Incidència</u>	14
2.2	<u>TIPUS I DESCRIPCIÓ</u>	15
2.2.1	<u>Càncer de pell no melanoma</u>	15
2.2.2	<u>Càncer de pell melanoma</u>	17
2.3	<u>FACTORS DE RISC</u>	18
2.4	<u>PREVENCIÓ</u>	20
2.5	<u>SÍMPTOMES</u>	20
2.6	<u>DIAGNÒSTIC</u>	21
2.7	<u>ESTATS DEL CÀNCER</u>	23
2.8	<u>TRACTAMENT</u>	24
2.9	<u>SUMARI</u>	26
3	<u>ANÀLISI, DISSENY I IMPLEMENTACIÓ DE L'ALGORITME DE REGISTRE 2D</u>	28
3.1	<u>INTRODUCCIÓ</u>	28
3.2	<u>REGISTRE D'IMATGES 2D</u>	28
3.2.1	<u>Mètriques utilitzades</u>	29
3.2.2	<u>Interpolador lineal</u>	31
3.2.3	<u>Optimitzador: Gradient descent (gradient descent)</u>	32
3.2.4	<u>Transformació rígida 2D i Transformació Afí</u>	34
3.2.5	<u>Exemple de Registre</u>	35
3.3	<u>SUMARI</u>	35
4	<u>AVALUACIÓ DELS MÈTODES DE REGISTRE</u>	37
4.1	<u>INTRODUCCIÓ</u>	37
4.2	<u>MÈTODE PER L'AVALUACIÓ</u>	37
4.2.1	<u>Diferència d'imatges</u>	37
4.2.2	<u>Checkerboard</u>	38
4.2.3	<u>Informació Mútua</u>	38
4.2.4	<u>Diferència de quadrats</u>	39
4.3	<u>PROVES I AVALUACIÓ DELS RESULTATS</u>	39
4.3.1	<u>Imatges Utilitzades</u>	39
4.3.2	<u>Registre</u>	40
4.3.3	<u>Avaluació</u>	45
4.4	<u>ESTUDI COMPARATIU</u>	53
4.5	<u>CONCLUSIONS</u>	58
4.6	<u>SUMARI</u>	58
5	<u>RECONSTRUCCIÓ DEL VOLUM 3D</u>	60
5.1	<u>OBJECTIUS</u>	60
5.2	<u>ADQUISICIÓ I MICROSCOPI CONFOCAL</u>	60
5.2.1	<u>Microscopi Confocal</u>	60
5.2.2	<u>Talls confocals i el seu format</u>	62
5.3	<u>REGISTRE D'IMATGES 2D</u>	64
5.3.1	<u>Registre amb varies imatges referència</u>	64
5.3.2	<u>Registre amb una imatge de referència (tall del mig)</u>	65
5.3.3	<u>Registre mixte (Agafar cada x talls una imatge de referència i aplicar en aquests talls el registre anterior)</u>	66
5.4	<u>RECONSTRUCCIÓ 3D DEL ISOVOLUM</u>	67
5.4.1	<u>Metodologia</u>	67

5.5	<u>EXEMPLES DE RECONSTRUCCIÓ</u>	68
5.6	<u>SUMARI</u>	68
6	<u>APLICACIÓ VISUAL PER L'ANÀLISI D'IMATGES CONFOCALS</u>	70
6.1	<u>OBJECTIUS</u>	70
6.2	<u>REQUERIMENTS</u>	70
6.3	<u>DESENVOLUPAMENT</u>	70
6.3.1	<i>Obrir el volum.</i>	71
6.3.2	<i>Visualització i interacció amb el volum 3D.</i>	72
6.3.3	<i>Selecció dels plans.</i>	73
6.3.4	<i>Navegació dins del pla</i>	74
6.3.5	<i>Altres funcionalitats.</i>	75
6.4	<u>EXEMPLES</u>	75
6.5	<u>SUMARI</u>	77
7	<u>CONCLUSIONS I TREBALLS FUTURS</u>	79
7.1	<u>CONCLUSIONS</u>	79
7.2	<u>TREBALLS FUTURS</u>	80
8	<u>BIBLIOGRAFIA</u>	82
9	<u>ANNEX. DISSENY I IMPLEMENTACIÓ</u>	84
9.1	<u>DISSENY</u>	84
9.1.1	<i>Registrar els talls.</i>	85
9.1.2	<i>Reconstrucció del volum 3D.</i>	91
9.1.3	<i>Visualització</i>	92
9.2	<u>IMPLEMENTACIÓ</u>	93
9.2.1	<i>Registre 2D de les imatges mèdiques.</i>	93
9.2.2	<i>Reconstrucció volum tridimensional</i>	96
9.2.3	<i>Diferència</i>	97
9.2.4	<i>Checkerboard</i>	98
9.2.5	<i>Visualització 3D</i>	98
10	<u>ANNEX. IMPLEMENTACIÓ EINA VISUAL</u>	104
10.1	<u>IMPLEMENTACIÓ</u>	104
11	<u>ANNEX. EINES DE DESENVOLUPAMENT</u>	118
11.1	<u>ENTORN DE DESENVOLUPAMENT</u>	118
11.2	<u>LLENGUATGE C++</u>	119
11.3	<u>ITK</u>	119
11.4	<u>VTK</u>	120
11.5	<u>CMAKE</u>	120
11.6	<u>QT</u>	121
11.7	<u>CONFIGURACIÓ CMAKE, ITK I VTK</u>	121
12	<u>ANNEX. CONTINGUT DEL CD ROM</u>	124

1. Descripció del Projecte

Jaume Delgado Fernandez
Enginyeria Tècnica en Informàtica de Sistemes.

Universitat de Girona, 2006

1.Descripció del Projecte

1.1 Motivació

El càncer de pell es considera un dels tipus de càncer més freqüents actualment, entre d'altres factors degut a l'augment en l'exposició a la radiació ultraviolada (UV). Com passa amb molts tipus de càncers, la detecció precoç augmenta les perspectives de curació.

Recentment la utilització de la Microscòpia Confocal (MCF) per a l'avaluació i diagnosi del càncer de pell ha rebut un important interès. El principal avantatge és la capacitat de visualitzar en temps real la regió d'interès a nivell cel·lular, similar a la informació obtinguda en una biòpsia, sense el patiment que suposa per el pacient. El principal inconvenient però, és que les imatges obtingudes amb MCF són difícils d'interpretar per els metges en el format actual (conjunt de talls 2D a diferents profunditats de la pell).

El microscopi confocal és una de les tècniques més actuals de diagnòstic, i s'ha establert com a una eina per obtenir imatges d'alta resol·lució i reconstruccions 3-D d'una gran varietat de mostres biològiques. És capaç d'escombrar diferents plans en l'eix Z, obtenint imatges 2D de diferent profunditat juntament amb la informació dels paràmetres de captura (com ara la profunditat, potència del làser, posicionament en x,y,z, etc). Mitjançant eines informàtiques es pot integrar aquesta informació en un model 3D de la regió d'interès. L'objectiu principal d'aquest projecte és el desenvolupament d'una eina per a l'ajuda en la interpretació de les imatges MCF i així poder millorar el diagnosi del càncer de pell.

1.2 Objectius del Projecte

Aquest Projecte Final de Carrera (PFC) es centra en la construcció d'una eina informàtica per l'ajuda en la interpretació, processament i visualització d'imatges obtingudes mitjançant el microscopi confocal (MCF), per obtenir un model tridimensional que permeti ajudar als dermatòlegs a avaluar i diagnosticar una possible lesió cancerígena a la pell. Dividirem els objectius en 4 apartats:

Les tasques planificades són:

- **Anàlisi, Disseny i Implementació (ADI) del mòdul de registre d'imatges de pell 2D confocals.** S'estudiaran mètodes per registrar imatges 2D amb les llibreries ITK , ja que és important com a pas previ a la reconstrucció 3D.
- **Avaluació de resultats del registre.** S'estudiarà la necessitat de registrar o no les imatges 2D i s'avaluarà de manera quantitativa i qualitativa els diferents mètodes implementats. Es triarà l'opció que s'adeqüi millor a les nostres necessitats.
- **ADI del mòdul de reconstrucció del volum 3D de la pell.** Donada la informació provinent del MCF i amb les imatges registrades es procedirà a reconstruir el volum 3D de la regió d'interès.

- **ADI de l'aplicació visual.** L'aplicació s'encarregarà de la interpretació i visualització de les imatges dels pacients. Cal remarcar que el disseny tindrà en compte les necessitats dels especialistes així com la futura implantació de noves prestacions al sistema (adquisició i tractament d'imatge confocal).

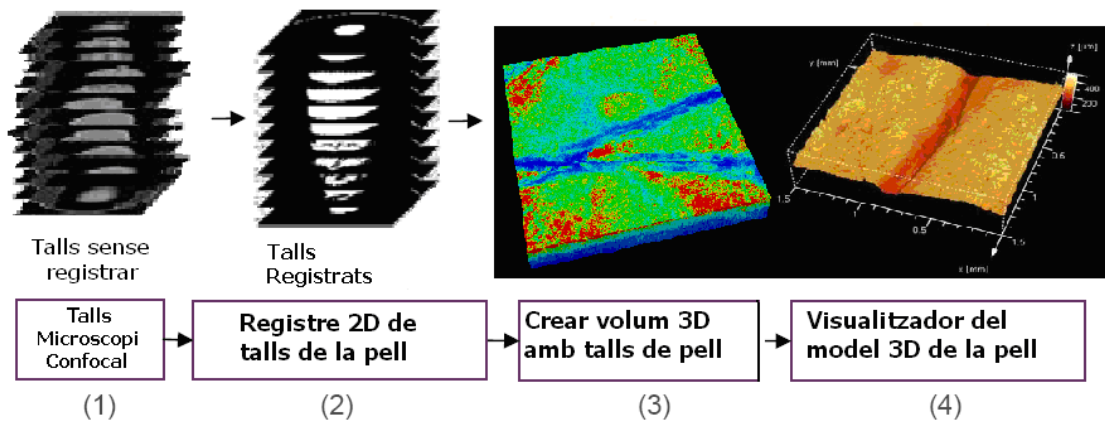


Figura 1.1. Esquema del sistema confocal per l'aplicació de tractament d'imatges.

Així com mostra la Figura 1.1, tenint com a base les dades provinents del microscopi confocal sobre imatges de la pell, l'objectiu final és la identificació de la pell i la seva reconstrucció volumètrica, cosa que permetrà l'ajuda en la interpretació visual del volum, per poder fer talls histològics i poder identificar zones de la pell en diferents profunditats i en qualsevol eix de coordenades x,y,z . Però abans de reconstruir el volum, la nostra feina és mirar quina és la millor manera de registrar les imatges amb diferents profunditats, o si en definitiva, és més eficient per la reconstrucció tridimensional que registrem les imatges o que no ho fem. Tots aquests conceptes s'explicaran detalladament en els següents capítols de la memòria.

El present projecte està estructurat en vuit capítols el contingut dels quals s'explica breument a continuació. El segon capítol, el denominat Marc de treball, consisteix en introduir el significat del càncer de pell en el context del nostre projecte, o sigui d'una manera més planera que si fos la part mèdica, com la malaltia des d'on partirem per aprofundir en tots els nostres estudis.

El tercer capítol tracta del tema del registre d'imatges 2D. S'explicarà que a partir d'unes imatges 2D de la pell obtingudes amb una tècnica anomenada Microscòpia Confocal, alinearem tall per tall de diferents profunditats per obtenir un registre el més eficient possible.

Una vegada introduït tot el capítol anterior amb el registre 2D, en el quart capítol es faran una sèrie de proves registrant imatges amb diferents tipus de Transformacions i mètriques. Això ens mostrarà si ens interessa registrar les imatges per reconstruir el volum en tres dimensions i quin serà el mètode que dóna més bons resultats a l'hora d'alinejar els talls. També es farà una presentació dels mètodes utilitzats en la fase d'avaluar els avantatges o desavantatges de registrar els talls de pell.

En el cinquè capítol d'aquest projecte es tracta la reconstrucció del volum 3D. Aquí es detallarà i s'intentarà explicar diferents fases de la reconstrucció tridimensional, i entre elles la part més important de llegir la informació necessària dels fitxers de text que conté cada tall de pell, per obtenir la profunditat de cada imatge i poder fer el volum exacte.

Al capítol sis es parla de l'eina de visualització que crearem per la interacció entre l'especialista i tota la informació de la pell. La missió d'aquesta eina serà ajudar a la interpretació de les imatges MCF i així poder millorar el diagnosi del càncer de pell.

Els capítols set i vuit, finals d'aquest projecte, són les conclusions obtingudes amb la realització d'aquest projecte i les seves futures millores, i la bibliografia utilitzada.

1.3 Eines de desenvolupament

Aquest PFC s'ha desenvolupat sota l'entorn Windows, utilitzant eines de desenvolupament multiplataforma.

El llenguatge de programació emprat ha estat el C++ mitjançant el Visual Studio 2005 C++. Les eines utilitzades per desenvolupar el projecte han estat les llibreries ITK i VTK, juntament amb Qt Designer per la part de visualització. Mitjançant el cmake s'ha pogut garantir la portabilitat del codi a diferents entorns com el Linux, Windows, ..

Per més informació sobre les eines de desenvolupament consultar l'annex C.

1.4. Planificació

Per tal d'organitzar la feina a realitzar cal temporalitzar les tasques per tal d'assolir els objectius. A continuació es detalla en forma de calendari el pla de treball realitzat a 27 de Febrer del 2006.

		DL	DM	DX	DJ	DV	DS	DG
				1	2	3	4	5
		Primer contacte amb el projecte.						
6	7	Fase d'aprenentatge de les eines que utilitzarem.						
13	14	15	16	17	18	19		
20	21	22	23	24	25	26		
		Eines bàsiques ITK						
27	28							

FEBRER
2006

		DL	DM	DX	DJ	DV	DS	DG
				1	2	3	4	5
		Eines bàsiques ITK						
6	7	8	9	10	11	12		
13	14	15	16	17	18	19		
20	21	22	23	24	25	26		
		Fase 1: Registre 2D/2D d'imatges mèdiques						
27	28	29	30	31				

MARÇ
2006

	DL	DM	DX	DJ	DV	DS	DG
ABRIL 2006						1	2
	3	4	5	6	7	8	9
	10	11	12	13	14	15	16
	17	18	19	20	21	22	23
	24	25	26	27	28	29	30

Fase 1: Registre 2D/2D d'imatges mèdiques

Fase 2: Estudi de resultats sobre l'importància de registrar o no registrar les imatges 2D

	DL	DM	DX	DJ	DV	DS	DG
MAIG 2006	1	2	3	4	5	6	7
	8	9	10	11	12	13	14
	15	16	17	18	19	20	21
	22	23	24	25	26	27	28
	29	30	31				

Fase 2: Avaluació del registre d'imatges

Fase 3: Modelatge del Volum de la pell en 3D (ITK)

JUNY 2006	DL	DM	DX	DJ	DV	DS	DG
				1	2	3	4
	5	6	7	8	9	10	11
	Fase 3: Modelatge del Volum de la pell en 3D (ITK)						
	12	13	14	15	16	17	18
	19	20	21	22	23	24	25
26	27	Eines bàsiques VTK i Qt					

JULIOL 2006	DL	DM	DX	DJ	DV	DS	DG
						1	2
	3	4	5	6	7	8	9
	Eines bàsiques VTK i Qt						
	10	11	12	13	14	15	16
	Fase 4: Visualització VTK en 3D.						
17	18	19	20	21	22	23	
Construcció GUI							
24	25	26	(Interfície gràfica)			29	30
31							

AGOST 2006	DL	DM	DX	DJ	DV	DS	DG
		1	2	3	4	5	6
	7	8	9	10	11	12	13
	Documentació						
	14	15	16	17	18	19	20
	21	22	23	24	25	26	27
28	29	30	31				

La primera etapa Fase d'aprenentatge d'eines que utilitzarem correspon a l'etapa inicial de presa de contacte amb el/s tutor/s i de les eines necessàries per poder desenvolupar el projecte. Un cop aclarits els conceptes bàsics, el següent pas ha estat introduir-se en el món ITK i VTK. Lectura de manuals, lectura de codi i execució d'exemples,... han estat la majoria de coses a fer en la primera etapa del projecte.

Un cop assimilats tots aquests conceptes, el següent pas consisteix en començar a implementar els algoritmes corresponents a la primera tasca. L'objectiu d'aquesta fase consisteix en alinear tots els talls, és a dir, registrar cada tall amb l'anterior per tal d'alinear totes les imatges d'entrada (Microscopi confocal) de diferent profunditat.

Posarem tres talls de referència, i inicialment registrarem des del tall de referència amb els seus anteriors, després des de el segon tall de referència fins al anterior tall de referència, i així successivament. Un cop hem registrat tots els talls utilitzant diferents tipus de mètrica i de transformacions avaluarem quina és la manera més eficient d'alinear les imatges.

Aquesta serà la segona fase del projecte, que consistirà en estudiar les millores o problemes que es pot donar en el registre d'imatges 2D (imatges confocals agafades amb profunditats diferents). Es mesurarà la qualitat del registre quantitativament i qualitativament. Es triarà l'opció que s'adeqüi millor a les nostres necessitats.

L'objectiu de la tercera tasca consisteix en modelar un volum 3D, generat a partir dels talls registrats. Crearem un algorisme que a partir dels fitxers de text que porta cada imatge 2D, llegirem tota la informació necessària per modelar el volum amb les corresponents profunditats.

Amb el volum resultant creat en la tercera fase s'ha passat a fer la quarta tasca. Primerament, estudiarem les eines VTK i Qt per poder modelar la interfície a partir de ITK, VTK juntament amb les Qt.

L'objectiu d'aquesta tasca és crear una interfície visual per poder gestionar les necessitats dels especialistes amb el volum de la pell.

2. Marc de Treball

Jaume Delgado Fernandez
Enginyeria Tècnica en Informàtica de Sistemes.

Universitat de Girona, 2006

2. Marc de treball

Per tal de situar-se en l'entorn en que es treballarà, es veuran diferents aspectes que els dermatòlegs es troben en el dia a dia pel que fa al càncer de pell. Veurem quins són els factors que influeixen i quines són les eines en que es recolzen per tal de diagnosticar els seus pacients.

2.1 Càncer de pell

El càncer de pell es considera un dels tipus de càncer més freqüents actualment, entre d'altres factors degut a l'augment en l'exposició a la radiació ultraviolada (UV). Com passa amb molts tipus de càncers, la detecció precoç augmenta les perspectives de curació. La modalitat d'imatge mèdica que utilitzarem serà imatges Confocals (Microscopi confocal), molt útil per detectar i estudiar la malaltia. [Mundo Salud, 2006]

El càncer de la pell és una afecció en la qual cèl·lules malignes (canceroses) es formen en els teixits de la pell. La pell és l'òrgan més gran del cos. Protegeix contra la calor, la llum solar, les lesions i les infeccions. Ajuda també a controlar la temperatura corporal i emmagatzema aigua, grassa i vitamina D.

La pell té diverses capes, però les dues principals són la epidermis (capa superior o externa) i la dermis (capa inferior o interna), tal i com mostra la Figura 2.1.

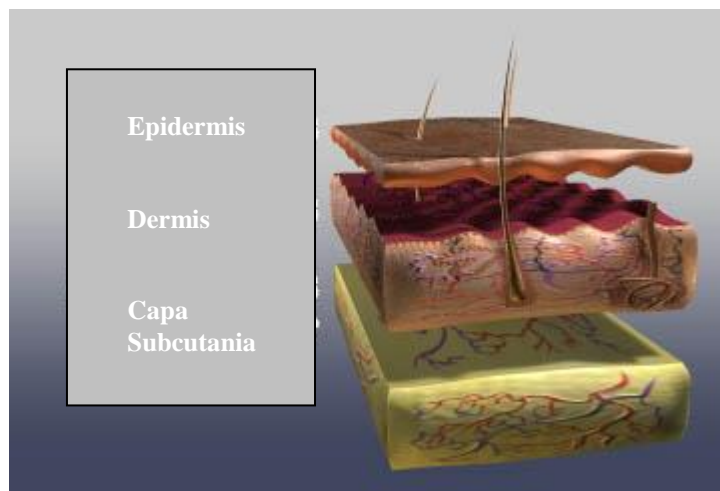


Figura 2.1. Capes de la pell.

El càncer de la pell comença en la epidermis, la qual està composta per tres tipus de cèl·lules, com són: (veure Figura 2.2).

- Cèl·lules escamoses: Cèl·lules primes, planes que formen la capa superior de la epidermis.
- Cèl·lules basals: Cèl·lules arrodonides sota les cèl·lules escamoses.
- Melanòcits: Aquestes cèl·lules que es troben en la part inferior de la epidermis produeixen melanina, el pigment que dona el color natural a la pell. Quan la pell està exposada al sol, els melanòcits produeixen més pigments, amb la qual cosa la pell s'enfosqueix.



Figura 2.2. Tipus de cèl·lules que componen la epidermis

El càncer de pell pot ocórrer arreu del cos, però és més comú en la pell que ha estat exposada a la llum solar, com la cara, el coll, les mans i els braços.

Existeixen bàsicament dos tipus de càncer de la pell:

- Melanoma.
- No Melanoma
 - Càncer de la pell basocel·lular (basalioma).
 - Càncer de la pell de les cèl·lules escamoses (Carcinoma escatós).

El més important és el melanoma, que és un tumor maligne dels melanòcits. La majoria dels melanomes es localitzen en la pell i menys freqüentment en mucoses, però donat l'origen dels melanòcits (deriven de la cresta neural), poden observar-se en altres localitzacions com retina i meninges on també migren els melanòcits.

2.1.1 Incidència

En els últims 50 anys ha existit un creixement exponencial dels casos de melanoma. La incidència als Estats Units ha augmentat des de 4,5/100.000 habitants en l'any 1970 fins a 11,7 i 15,2/100.000 en els anys 1987 i 1995. Un de cada 57 homes i una de cada 78 dones desenvoluparà un melanoma al llarg de la seva vida. El melanoma representa el 2,5% de tots els càncers i és responsable del 1-2% de les morts per càncer.

El melanoma és la causa més freqüent de mort entre les malalties cutània. Es desenvolupa de "novo" en el 75% dels casos i en el 25% es desenvolupa sobre un nevus preexistent. Els individus amb pell clara, rossos o pèl-rojos es veuen més afectats. Els pacients de raça negra tenen una incidència 10 vegades menor que els de raça blanca. Un 50% de melanomes s'observen en persones menors de 50 anys i el 35% en menors de 35 anys. Tot i que el 95% de melanomes es desenvolupa en la pell, poden desenvolupar-se en altres localitzacions com l'ull i mucosa oral o genital. Un 3% de pacients desenvolupen melanomes ocults (malaltia metastàtica sense evidència de tumor primari).

2.2 TIPUS I DESCRIPCIÓ

El càncer de pell és una malaltia produïda pel desenvolupament de cèl·lules canceroses en les capes exteriors de la pell. Existeixen dos tipus: el tipus no melanoma i el melanoma.

2.2.1 Càncer de pell no melanoma

El càncer no melanoma, bàsicament el podem dividir en dos apartats:

- Càncer de la pell de les cèl·lules escamoses (Carcinoma escatós).
- Càncer de la pell basocel·lular (basalioma).

2.2.1.1 Carcinoma de cèl·lules escamoses

El **carcinoma de cèl·lules escamoses** es desenvolupa en les capes intermèdies de l'epidermis i suposa el 20% de tots els casos de càncer de pell.

Apareix normalment en àrees que han estat exposades al sol, com la part superior del nas, orelles, front, llavi inferior i dors de les mans.

Pot donar-se també en aquelles zones de la pell que hagin estat en contacte amb productes químics, que hagin estat sotmeses a radioteràpia, o que s'hagin cremat. A l'àrea genital, apareix amb menor freqüència.

Generalment, sol presentar-se un bony vermell i dur. A vegades, pot tenir un aspecte, escamós, o sagnar i desenvolupar una crosta que mai cura. A mesura que augmenta de mida va tenint forma nodular i, en ocasions, presenta una superfície berrugosa. Al final, es converteix en una úlcera oberta i creix dins del teixit subjacent.

A través del sistema limfàtic, pot estendre's a qualsevol part del cos. (Figura 2.3)



Figura 2.3. Imatges del carcinoma de cèl·lules escamoses.

2.2.1.2 Carcinoma de cèl·lules basals

El **carcinoma de cèl·lules basals** és el tipus més comú de càncer de pell no melanoma. Al voltant del 75% de tots els tumors cancerosos de la pell són carcinomes de cèl·lules basals. Afecta a la capa inferior de l'epidermis, formada per cèl·lules basals. I sol donar-se en aquelles zones que han estat exposades al sol, com el cap i el coll. El seu creixement és lent i rares vegades s'estén a altres parts del cos, per la qual cosa és menys agressiu que el carcinoma de cèl·lules escamoses. Encara que no

per això s'ha de deixar de tractar, ja que si no es fa, amb el temps, s'estendrà als teixits circumdants i als ossos.

L'aspecte que presenta és un petit bony amb una aparença suau i brillant, però també pot tenir forma de cicatriu, sent ferm al tacte. Generalment es desenvolupa de forma lenta, encara que la velocitat varia d'un a un altre. No és infreqüent que torni a aparèixer una vegada que la persona ha estat tractada. Del 35% al 50% de les persones diagnosticades, desenvoluparan un nou càncer de pell en un termini de cinc anys a partir del primer diagnòstic. A continuació podem veure exemples d'aquest carcinoma (Figura 2.4).

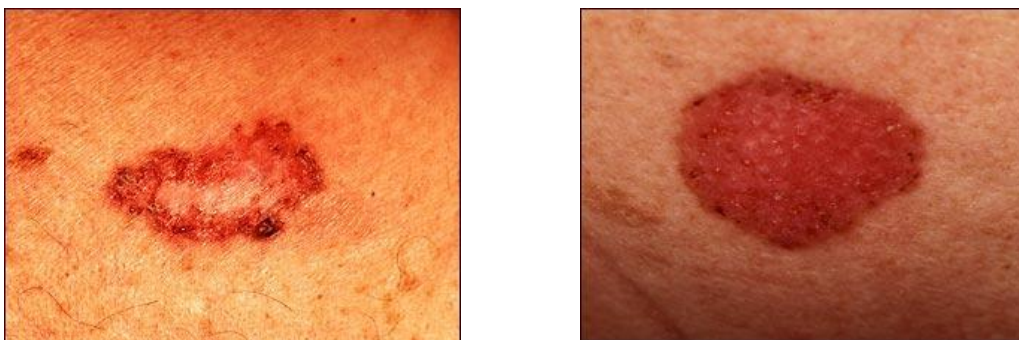


Figura 2.4. Imatges del carcinoma de cèl·lules basals.

Altres tipus menys freqüents que podrem trobar són:

- **Sarcoma de kaposi:** El Sarcoma de Kaposi (SK), és un tipus de càncer no melanoma. Les cèl·lules del càncer SK causen un creixement excessiu de petits vasos sanguinis, els quals formen tumors petits (punts o taques) cridats lesions. Aquestes lesions poden aparèixer en la pell i dintre del cos. Encara que algunes persones sense la infecció del VIH poden desenvolupar SK, el SK relacionat amb el VIH és molt més comú i agressiu. A la Figura 2.5 veiem unes imatges del Sarcoma de Kaposi.



Figura 2.5. Imatges del Sarcoma de Kaposi.

- **El limfoma cutani** es produeix quan els limfòcits de la pell (cèl·lules del sistema immunològic o de defensa) es tornen malignes. La dermis conté un nombre considerable de limfòcits. El tipus més comú d'aquests, és el limfoma cutani de cèl·lules To micosis fungoide. Es desenvolupa durant diversos anys, de forma molt lenta. Aquesta malaltia pot estendre's a través dels vasos limfàtics a altres òrgans com el pulmó, fetge, etc. A continuació, Figura 2.6 podem veure uns exemples d'imatges amb limfoma cutani.



Figura 2.6. Imatges del limfoma cutani.

2.2.2 Càncer de pell melanoma

Aquesta malaltia es desenvolupa en els melanòcits, cèl·lules de la pell que produeixen la melanina que li dona color. Els melanòcits es troben a l'epidermis, capa de la pell més externa. El melanoma és el tipus de càncer de pell menys comú i més greu. Pot estendre's a altres parts del cos a través de la sang o del sistema limfàtic. Si es diagnostica aviat, el seu pronòstic és millor. Es pot presentar a partir de pigues ja existents, o com una formació nova, petita i pigmentada, que apareix en una zona de la pell normal. Caldrà consultar amb el metge sempre que es noti un canvi en la mida, forma o color d'una piga, o una piga que s'està inflant, fa mal al tocar-la, traspua o sagna.

En general, en els homes sol aparèixer en el tronc i en les dones, és més freqüent en la part inferior de les cames. Aquestes persones solen tenir la pell clara. Pot aparèixer en persones de pell més fosca, en zones de les mans, planta dels peus, i sota les ungles. El melanoma representa un 4% de tots els tipus de càncer de pell. Però, a causa de la seva malignitat, és responsable de, aproximadament, un 79% de les morts per càncer de pell. El nombre de persones afectades per aquest tipus de càncer ha augmentat en els últims anys. A Espanya s'ha multiplicat per més de tres en homes i per una mica més de dos en dones, en els darrers vint anys. En Estats Units, el nombre de casos s'ha multiplicat per dos, des de 1973, i moren aproximadament unes 6.000 persones a l'any.

En la següent figura (Figura 2.7) podem observar un exemple de seguiment de les lesions d'un pacient amb melanomes amb un any de diferència. Aquest pacient té més de 150 lesions i es fa un estudi comparatiu on es detecta una lesió melanocítica amb un creixement ràpid.

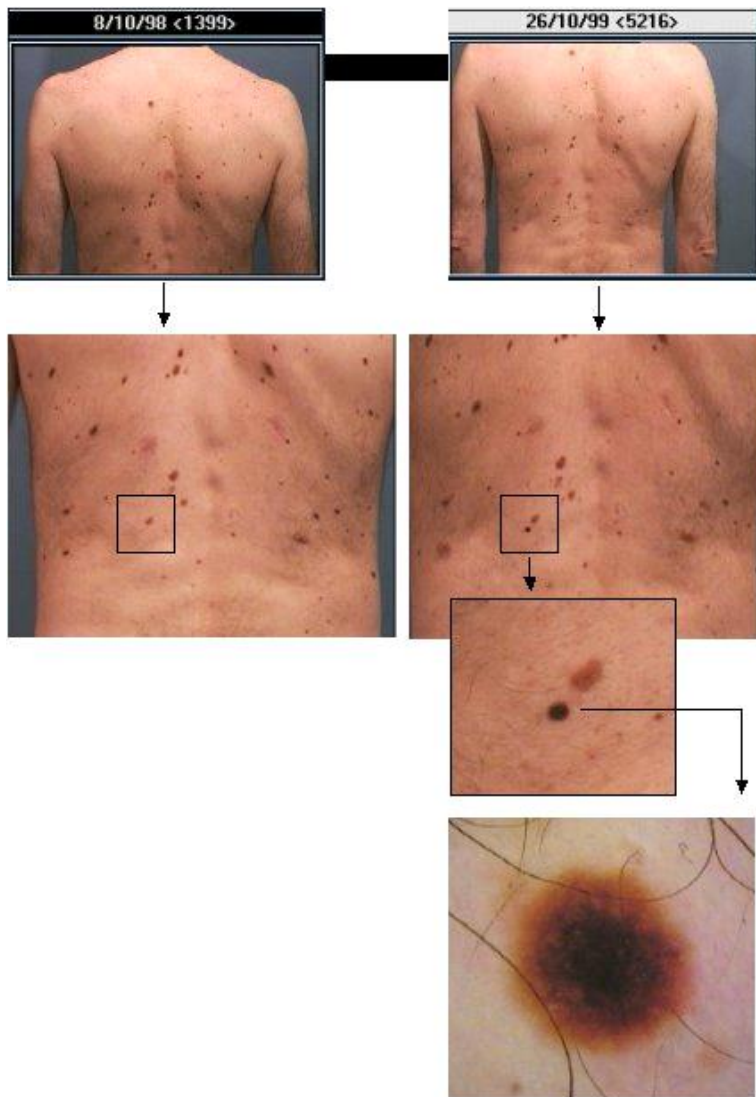


Figura 2.7. Seguiment d'un pacient amb lesions de melanomes.

2.3 FACTORS DE RISC

La causa que provoca aquest tipus de càncer no és del tot coneguda. Si que s'han estudiat determinats factors que influeixen en la seva aparició.

Factors ambientals

L'exposició excessiva al sol influeix en la producció d'aquests càncers. Persones que treballen en l'exterior, com agricultors i mariners, presenten una major incidència. Se solen produir a la pell exposada, freqüentment al cap i coll. Encara que la llum solar ajuda a sintetitzar les vitamines A i D, l'exposició excessiva, produeixi o no cremades, augmenta el risc de desenvolupar càncer de pell, incloent el melanoma maligne.

Existeix una correlació entre el risc de melanoma i la latitud. També influeix el període de temps durant el que ha viscut una persona en països de latituds equatorianes, el que significa una exposició excessiva al sol mantinguda durant un període de la vida. No només influeix l'exposició perllongada sinó que una exposició intensa al sol, puntual, durant unes vacances per exemple, constitueix un factor de risc de melanoma.

Els melanomes difereixen dels tumors malignes de la pell de tipus no melanoma pel que fa al sexe, edat i localització corporal. La major part dels melanomes malignes cutanis no ocorren en zones que reben la dosi ultraviolada acumulada major. Encara que no es coneix exactament quin és l'esquema més perillós d'exposició al sol, alguns estudis donen suport la hipòtesi que l'exposició intensa i intermitent a la radiació ultraviolada de la pell que es troba normalment protegida és la responsable de la formació del melanoma. L'edat també sembla influir i les cremades produïdes pel sol abans dels 15 anys augmenten el risc de melanoma al doble.

Característiques físiques

Les persones de raça blanca d'origen escocès, anglès o irlandès amb el pèl ros o vermellós, els ulls clars i abundants pigues són especialment susceptibles. El melanoma és molt rar en la població negra o oriental sent predominant, quan es dona, en zones poc pigmentades com els palmells i plantes i el seu pronòstic és pitjor. La sensibilitat de la pell al sol i la dificultat per a bronzear-se, augmenta el risc de melanoma. La reacció cutània a la llum solar es relaciona amb factors com la pigmentació de la pell, el nombre de pigues en la infància o edat adulta i el nombre de nevus (formacions semblants a pigues i que són tumors melanocítics benignes), constituint tots ells factors de risc per al melanoma maligne cutani. La major incidència de nevus en individus de raça blanca condueix a la idea que la radiació ultraviolada desenvolupa un paper important en el desenvolupament de nevus. S'ha comprovat l'existència de major nombre de nevus en zones de la pell exposades al sol que en zones protegides, associant-se el seu augment amb una major propensió a cremar-se que a bronzear-se, el nombre de cremades solars, la tendència a les pigues i l'estil de vida relacionat amb una major exposició solar. Una persona amb pell sensible al sol, més de 150 nevus melanocítics i algun nevus displàsic, podria presentar 50 vegades més risc de melanoma que altra persona sense aquestes característiques.

Factors genètics

L'existència de casos de melanoma familiar és altre factor de risc de desenvolupament de melanoma. La presència del melanoma familiar pot ser deguda a diversos factors. Existeix un petit percentatge de casos en els quals el melanoma és hereditari i es relaciona amb presència de Alella del gen CDKN2A. L'existència prèvia d'un melanoma és altre factor de risc important, el risc relatiu de desenvolupar un segon melanoma és 70 vegades superior al de desenvolupar un primer melanoma. L'existència de nevus melanocítics congènits o de múltiples nevus adquirits representa també un factor de risc important de desenvolupament de melanoma maligne. Les persones que presenten més de 50 nevus melanocítics comunes tenen un factor de risc 3 vegades superior al de la població normal, i les persones que tenen més de 100 nevus tenen 7,6 vegades més risc de desenvolupar un melanoma. Els nevus melanocítics congènits gegants tenen un risc de transformació maligna del 6% al 8%, i generalment desenvolupen melanoma abans dels 10 anys d'edat del pacient.

Sobre-exposició a làmpades i cabines bronzejadors

Els llums i cabines bronzejadores són una font de radiació ultraviolada. Una exposició excessiva augmenta el risc de desenvolupar càncer de pell.

Edat

Aproximadament la meitat del nombre de casos de melanomes es presenta en persones majors de 50 anys.

Altres factors

L'exposició a carcinògens, els traumatismes o cicatrius, les lesions per radiació crònica i les infeccions víriques, són alguns dels factors predisposats al càncer de pell.

2.4 PREVENCIÓ

El principal factor de risc en aquest càncer és una exposició excessiva a la radiació ultraviolada. Evitar una intensa o perllongada exposició al sol, intentant no exposar-se en hores en les quals la radiació solar és major és la millor mesura de prevenció que es pot utilitzar. Altres formes són la utilització de materials que protegeixin aquelles zones més delicades com és l'ús de barrets, d'ulleres que absorbeixin els raigs ultraviolat d'un 99% a un 100%, o utilitzar teles adequades per a cobrir la pell. L'ús de cremes protectores solars redueix el perill de l'exposició. Han d'utilitzar-se correctament, hi ha diferents graus segons sigui el tipus de pell. A més deuen emprar-se amb un temps d'antelació a l'exposició solar. Com s'ha vist, les cabines i els llums bronzeadores resulten perilloses, pel que han d'evitar-se. Quan s'observi una piga que ha canviat d'aspecte, o que sagna, s'ha d'acudir al metge. Aquest ho pot extirpar i realitzar una biòpsia per a comprovar si és maligne o no. Aquest tipus de càncer pot prevenir-se més que molts altres. Seguint les anteriors instruccions poden disminuir-se o anul·lar-se gran part dels factors de risc, amb el que les possibilitats de patir un càncer de pell també disminueixen.

Principals mesures abans i durant l' exposició solar

- Evitar l'ús de productes que continguin alcohol i perfums.
- Triar el protector solar adequat, en funció del tipus de pell, del lloc d'aplicació i de les condicions ambientals.
- Aplicar una bona quantitat del producte solar 30 minuts abans de prendre el sol, sobre la pell seca.
- Evitar prendre el sol entre les dotze del matí i les quatre de la tarda, en zones de gran altitud, i en llocs pròxims a l'equador.
- L'aigua, la neu i la sorra actuen reflectint els raigs solars i augmentant la seva intensitat. És per aquest motiu, pel que poden produir-se cremades fins i tot en l'ombra.
- Les primeres vegades que es prengui el sol, s'haurà d'emprar un factor de protecció més elevat.
- Encara en els dies nuvolosos, cal utilitzar el protector solar.
- Cal ingerir molts líquids per a compensar la pèrdua de líquids a causa de l'exposició solar.
- Després d'un bany, quan s'hagi suat molt, o després de passar dues hores de l'anterior aplicació, caldrà tornar a aplicar la crema protectora.
- Després de l'exposició al sol, cal ingerir molts líquids a causa de la pèrdua d'aquests.

2.5 SÍMPTOMES

El càncer de pell no melanoma pot tenir l'aspecte de diverses marques en la pell. Els carcinomes de cèl·lules basals apareixen en forma d'àrees de color vermell, planes o escamoses, o de petites àrees ceroses, brillants i translúcides al relleu que poden sagnar. Pot haver algun got sanguini irregular visible, o mostrar àrees de color blau, cafè o negre. Els carcinomes de cèl·lules escamoses poden tenir forma de protuberàncies creixents, de superfície aspra, o plans com taques vermelloses de la pell que creixen lentament. El melanoma pot aparèixer com un canvi en aquelles taques de la pell. Qualsevol nafra, protuberància, marca, etc. que sigui sospitosa pogués ser un melanoma. La pell pot tornar-se aspra o escamosa o pot sagnar o traspuar. Es pot donar un melanoma a partir d'una piga, que canviï d'aspecte, o

textura. En general, una piga és una taca de color uniforme, de color cafè, canyella o negre en la pell. Tenen menys de sis mil·límetres de diàmetre i pot estar present des del naixement o pot aparèixer en la infància o joventut. La majoria de les persones tenen pigues que són benignes. És important reconèixer els seus canvis. La regla del ABCD* pot ajudar a reconèixer les característiques d'un melanoma:

(A)simetria: La meitat de la piga no es correspon amb l'altra meitat.

(B)ordes irregulars: Els bordes de la piga són desiguals.

(C)olor: El color de la piga no és uniforme, les seves tonalitats varien des d'un marró a un vermell o blau.

(D)jàmetre: La piga té més de 6 mil·límetres d'ample.

2.6 DIAGNÒSTIC

El metge es basarà en els símptomes que presenti la pell del pacient i la història clínica. Quan se sospiti l'existència d'alguna anomalia, es realitzarà una biòpsia. Amb la biòpsia s'analitza el teixit sota el microscopi per a examinar el tipus de cèl·lules que presenta. Depenent d'on es trobi l'alteració de la pell i del tipus que sigui, així es realitzarà un tipus de biòpsia o altre.

- **Biòpsia:** Es l'extracció d'una petita part de la pell per diagnosticar o descartar una malaltia.
 - Biòpsia per raspat: S'insensibilitza la zona de pell a biopsiar amb anestèsia local i es raspen les capes superiors de la pell amb una fulla de bisturí.
 - Biòpsia per punció: S'extreu una mostra de pell més profunda.
 - Biòpsies incisionals i excisionals: S'extreu un tascó de pell. Es realitza per a tumors més profunds. Amb la biòpsia excisional s'extirpa tot el tumor.
 - Biòpsia per aspiració amb agulla fina: S'utilitza una xeringa amb una agulla fina per a extreure petites partícules del tumor.

Desgraciadament el diagnòstic clínic, fins i tot en mans d'experts no pot aconseguir una gran precisió diagnòstica degut al fet que moltes lesions poden ser difícils d'identificar especialment en fases inicials. Això comporta la realització de gran número de biòpsies innecessàries en pacients amb múltiples nevus melanocítics. Per aquest motiu resulta fonamental aconseguir la màxima precisió amb nous sistemes de diagnòstic no invasiu.

Aquests nous sistemes són:

- **Confocal:** La diagnosi amb Microscopía Confocal permet l'estudi de mostres amb marcatge fluorescent, fent seccions òptiques de les mateixes. S'excita la mostra punt a punt per mitjà d'un làser. La longitud d'ona d'emissió d'aquesta mostra és major a la d'excitació, i és aquesta última la qual al passar per un petit diafragma (pinhole) permet la detecció d'un sol plànol focal. La Figura 2.8 mostra un exemple amb diagnosi de MCF.

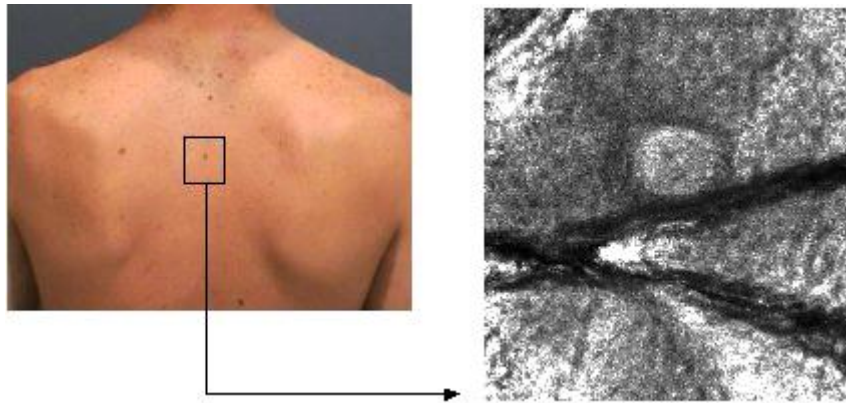


Figura 2.8. Imatge procedent d'un diagnosi de MCF

- **Dermatoscòpia:**

La Dermatoscòpia és una tècnica de diagnòstic per imatge no invasiva que permet millorar de forma significativa la precisió diagnòstica en les lesions melanocítiques. Els nous sistemes informàtics de diagnòstic permeten obtenir mapes corporals totals, localitzar les lesions i arxivar electrònicament les imatges de les lesions amb microscòpia d'epiluminiscència digital. Això fa possible realitzar controls seriatos molt precisos del pacient i detectar canvis mínims suggestius de fer-se malignes fins i tot quan existeix un gran nombre de lesions. La Figura 2.9 ens mostra un diagnòstic amb dermatoscòpia.

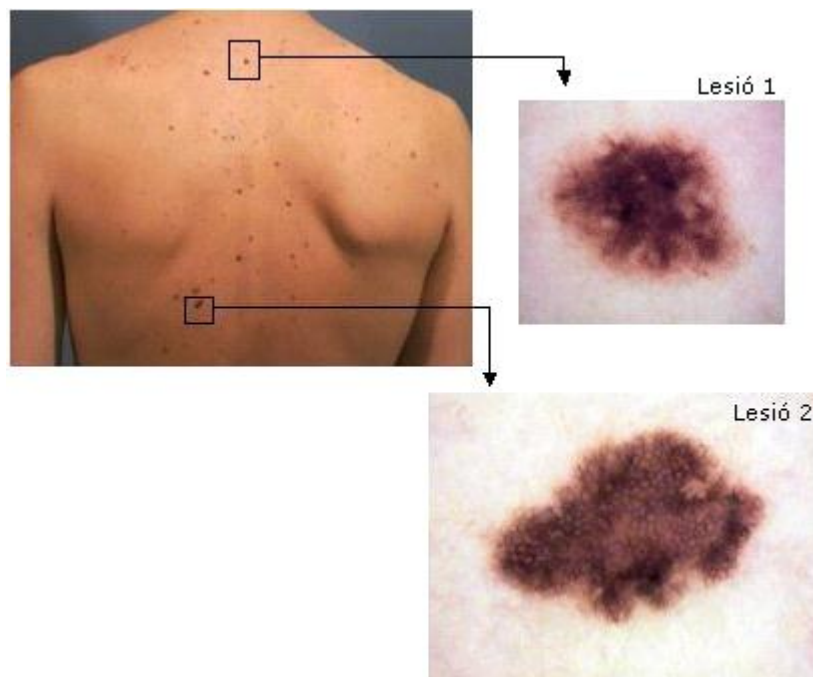


Figura 2.9. Imatges procedents d'un diagnòstic amb dermatoscòpia

2.7 ESTATS DEL CÀNCER

Càncer no melanoma

- Etapa 0: el carcinoma de cèl·lules escamoses in situ, o malaltia de Bowen, és l'etapa inicial del carcinoma de cèl·lules escamoses. Aquest càncer només afecta a la epidermis.
- Etapa I: el càncer no té més de dos centímetres i no hi ha ganglis limfàtics ni altres òrgans afectats.
- Etapa II: el càncer té més de dos centímetres però no s'ha estès a ganglis limfàtics ni a altres òrgans.
- Etapa III: el càncer s'ha propagat a teixits profunds, com són ossos, músculs o cartílags, i/o als ganglis limfàtics regionals. No hi ha afectació d'òrgans distants.
- Etapa IV: el càncer pot tenir qualsevol grandària i s'ha estès a òrgans com el cervell o els pulmons.

Càncer melanoma

- Etapa 0: el melanoma és in situ, és a dir, es troba en la epidermis.
- Etapa I: el melanoma encara és un tumor de baix risc. Té menys d'un mil·límetre i mig i es troba localitzat en la pell sense afectació ganglionar.
- Etapa II: el seu grossor és major d'un mil·límetre i mig. Encara està localitzat en la pell, sense afectació ganglionar.
- Etapa III: el melanoma s'ha estès als ganglis limfàtics propers a la pell afectada.
- Etapa IV: el melanoma s'ha propagat a zones distants de la pell afectada, a òrgans i/o ganglis distants.

Pronòstic per etapes

Per a l'etapa I l'índex de supervivència per a cinc anys és de més del 90%.

Per a les etapes II i III, els índexs són al voltant del 80% i del 50% respectivament.

En l'etapa IV, l'índex de supervivència a cinc anys disminueix entre un 20% i 30%.

Tal i com mostra la següent figura (Figura 2.10), observem els estats del càncer i la seva evolució. Podem veure com el melanoma es va estenen cap a zones més interiors de la pell.

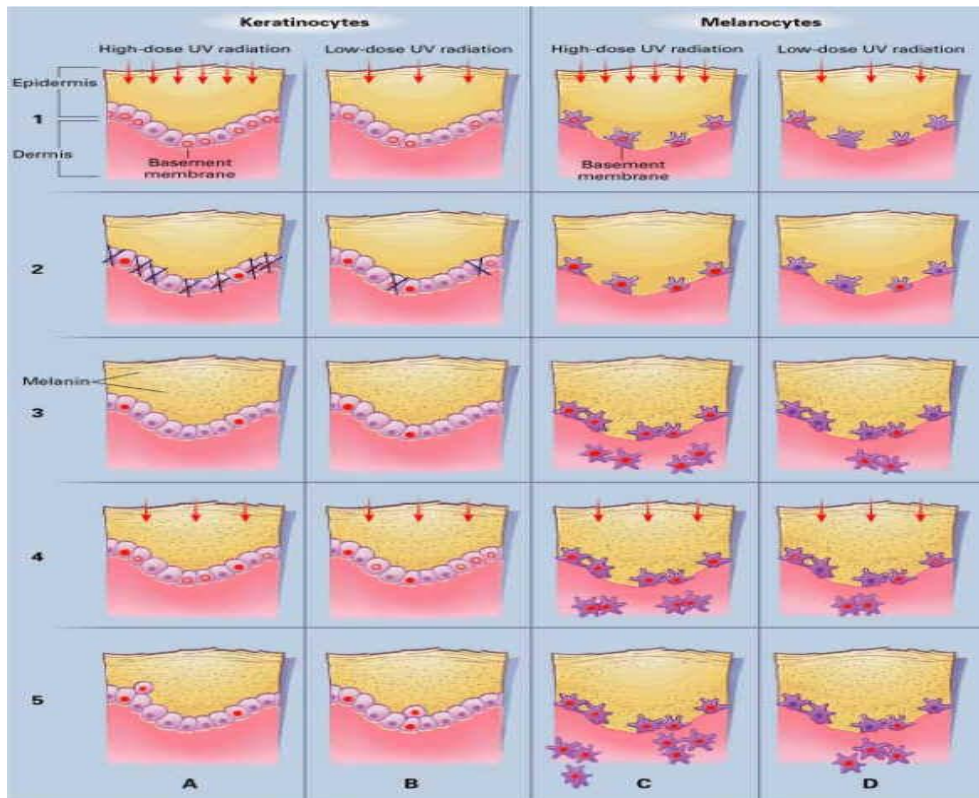


Figura 2.10. Exemple de les fases del càncer

2.8 TRACTAMENT

Els tractaments que es descriuen en aquesta secció s'apliquen en els diferents tipus de càncer de pell. Aquests tractaments els podem dividir en tres parts principals, com són la cirurgia, quimioteràpia i radioteràpia.

Cirurgia

- **Escissió simple:** el tumor i una porció de teixit circumdant s'extirpen mitjançant bisturí.
- **Electrodessecació i curetatge:** el tumor s'extreu raspant el teixit amb un instrument cridat cureta. Després es tracta el teixit on es trobava el tumor amb una agulla elèctrica per a destruir les cèl·lules canceroses que quedin.
- **Criocirurgia:** s'utilitza nitrogen líquid per a congelar i així destruir les cèl·lules canceroses.
- **Cirurgia de Moh:** S'elimina la capa de la pell afectada pel càncer i més tard s'extreu el teixit del voltant (teixit circumdant) comparant ambdós mitjançant proves en el microscopi.
- **Cirurgia per làser:** s'utilitzen els raigs làser per a vaporitzar les cèl·lules canceroses. Aquesta tècnica s'empra en el carcinoma de cèl·lules basals molt superficials i en el carcinoma de cèl·lules escamoses in situ.
- **Cirurgia de ganglis limfàtics:** si els ganglis propers al càncer estan inflamats caldrà extirpar-los mitjançant cirurgia, i examinar-se sota un microscopi per a determinar la presència de cèl·lules canceroses.

Quimioteràpia

La quimioteràpia és un tractament que utilitza medicaments per destruir les cèl·lules canceroses del nostre cos. Depenen de com apliquem aquest tractament és separada en:

- **Quimioteràpia tòpica:** Significa que el medicament contra el càncer es col·loca directament en la pell, en lloc d'administrar-se per via oral o injectar-se en una vena. El medicament d'ús més freqüent en el tractament tòpic del càncer de la pell és el fluorouracil·lo (també cridat 5-FU). Quan s'aplica directament en la pell en forma de crema, el fluorouracil·lo arriba a les cèl·lules canceroses properes a la superfície de la pell, però no a les quals han penetrat profundament, ni a les quals s'han estès a altres òrgans. Per aquest motiu, el tractament amb fluorouracil·lo sol utilitzar-se solament en condicions premalignes
- **Quimioteràpia sistèmica:** la quimioteràpia sistèmica utilitza medicaments contra el càncer que s'injecten en una vena o s'administren per via oral. Aquests medicaments passen a través del flux sanguini a totes les parts del cos. A diferència de la quimioteràpia tòpica, la quimioteràpia sistèmica pot atacar les cèl·lules canceroses que s'han propagat als ganglis limfàtics i altres òrgans.

A la Figura 2.11 següent, veiem la imatge del tractament d'un pacient amb quimioteràpia.



Figura 2.11. Tractament de la Quimioteràpia

Radioteràpia

Consisteix en l'ocupació de raigs d'alta energia, com raigs X, per a destruir o disminuir el nombre de cèl·lules canceroses. És un tractament local. Es desenvolupa al llarg d'uns dies (els quals el oncòleg i el radiòleg hagin cregut convenients). En si, el tractament dura uns minuts. No és dolorós sinó que és una mica semblat a una radiografia només que la radiació és major i està concentrada en la zona afectada. La radioteràpia s'utilitza en aquest tipus de càncer en aquells pacients que no poden sotmetre's a cirurgia pel seu estat general. La Figura 2.12 mostra un exemple de tractament amb radioteràpia.



Figura 2.12. Tractament de la Radioteràpia

Petites conclusions:

- La majoria dels carcinomes de cèl·lules basals i de cèl·lules escamoses es curen mitjançant cirurgia menor. Depenent de la localització i el tipus de tumor s'empraran diversos tipus de cirurgia.
- La cirurgia làser s'empra bàsicament en el carcinoma de cèl·lules basals molt superficials i en cèl·lules escamoses in situ.
- El tractament de la radioteràpia pot guarir aquells tipus de càncer que es trobi en estats baixos. També pot utilitzar-se com complement a la cirurgia.

2.9 Sumari

En aquest capítol hem presentat informació per conèixer amb més detall el nostre marc de treball, el càncer de pell. En els pròxims capítols desenvoluparem les diferents tasques del projecte com ara investigar sobre la necessitat de registre d'imatges confocals i la reconstrucció del volum tridimensional.

3. ADI de l'algoritme de Registre 2D

Jaume Delgado Fernandez
Enginyeria Tècnica en Informàtica de Sistemes.

Universitat de Girona, 2006

3. Anàlisi, Disseny i Implementació de l'Algoritme de Registre 2D

3.1 Introducció

En aquest capítol parlarem d'una de les tasques més importants del nostre projecte: el registre d'imatges 2D. A partir d'un conjunt de talls desalineats corresponents a les imatges de la pell amb diferents profunditats i temps de captura, l'objectiu de poder tenir els talls obtinguts del microscopi confocal perfectament alineats.

3.2 Registre d'imatges 2D

L'objectiu d'aquest capítol és proposar i aplicar diferents mètodes de registre per tal d'avaluar quin és el millor mètode per a minimitzar el possible moviment entre talls i poder obtenir una reconstrucció 3D més acurada.



Figura 3.1. Pas de les imatges desalineades obtingudes amb el microscopi confocal on mitjançant un registre 2D/2D de cada tall obtindrem l'alineat dels talls.

El registre de les imatges és el procés que determina les transformacions espacials necessàries per tal de situar mapejats de punts homogenis d'una imatge a una segona imatge. En aquest projecte s'avaluaran 4 mètodes de registre diferents, combinant la utilització de mètriques de mitjanes quadrades i informació mútua i les transformacions rígides i afins. [Insight into Image,2004]

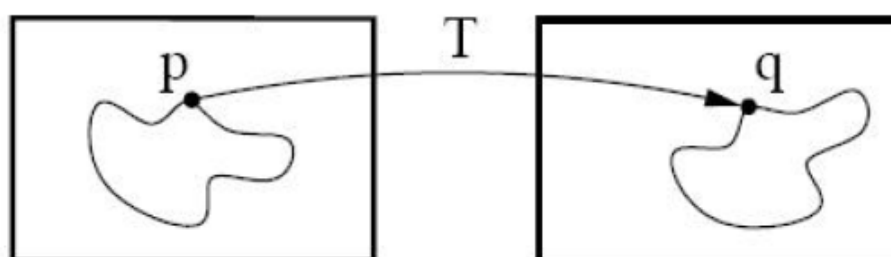


Figura 3.2. El registre d'imatges és el procés de trobar un mapa de punts d'una imatge a una altra.

Per tal de registrar, necessitem dues imatges, una definida com la imatge fixa ($f(x)$) i una altra denominada la imatge moguda ($m(x)$). On x representa les coordenades de la imatge.

El registre és tractat com un problema d'optimització amb l'objectiu de poder trobar una traça adequada que permeti alinear la imatge fixa amb la imatge moguda. El registre es basa en la utilització de diferents components bàsics que són: la mètrica, l'interpolador, l'optimitzador i el transformador.

El transformador representa al conjunt de transformacions (ex. escalats), translacions i rotacions que s'aplicaran a la imatge fixa respecte la imatge moguda per poder trobar un mapejat de punts que estigui i concordi a les dues imatges. L'interpolador s'utilitza per avaluar la intensitat dels mapejats obtinguts de la imatge moguda respecte la imatge fixa de les possibles posicions que pot agafar. La mètrica mesura com de bé es pot aparellar la imatge fixa respecte la imatge moguda transformada, el grau de similitud entre la imatge fixa i la moguda. Un exemple és la diferència entre imatges. L'optimitzador es calcula bàsicament com una funció de cost, que optimitza la mètrica respecte els paràmetres del transformador. La Figura 3.3 mostra com es relacionen cadascun dels components bàsics del registre.

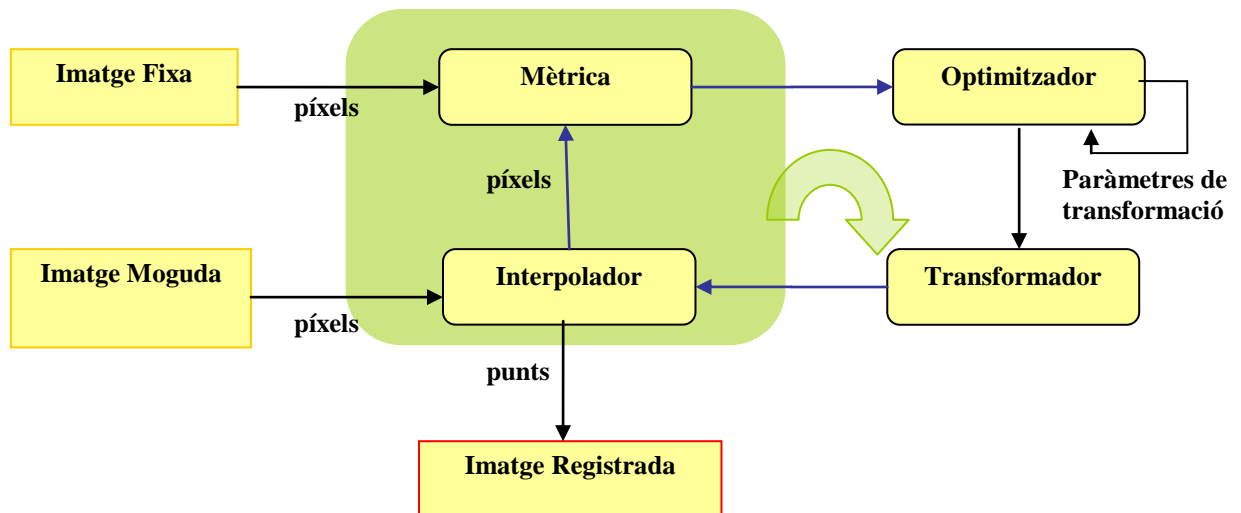


Figura 3.3 Aquests són els components bàsics d'un registre. Dues imatges d'entrada, el transformador, la mètrica, l'interpolador i l'optimitzador.

De manera comú a tots els mètodes s'utilitza interpolació lineal i el mètode d'optimització de gradient descendent. La mètrica i el transformador provem amb dues opcions.

Els mètodes finalment utilitzats en cada component bàsica són els següents:

- Mètrica → mètrica de mitjanes quadrades i Informació Mútua.
- Interpolador → interpolador lineal
- Optimitzador → gradient descendent (gradient descent)
- Transformador → transformador rígid i transformació Afí.

3.2.1 Mètriques utilitzades

3.2.1.1 Mètrica de mitjanes quadrades

La mètrica tal com s'ha explicat anteriorment mesura la coincidència entre la imatge moguda transformada amb la imatge fixa, comparant la intensitat de grisos de les imatges. La mètrica s'avalua amb la imatge transformada (on hi intervé l'interpolador i el transformador). Mitjançant el transformador podem intentar correspondre mapejats

de punts en les dues imatges i mitjançant l'interpolador podrem avaluar la intensitat a la imatge moguda del mapejat obtingut i tot això donarà com a resultat un valor de mètrica.

La mètrica de mitjanes quadrades mesura la diferència de grisos de les dues imatges al quadrat d'una regió determinada seguint la fórmula de la següent equació:

$$MS(A, B) = \frac{1}{N} \sum_{i=1}^N (A_i - B_i)^2$$

A_i és el píxel i -éssim de la imatge A

B_i és el píxel i -éssim de la imatge B

N és el número de píxels de la regió

Quan més petit sigui el valor de la funció, vol dir que major serà la correspondència entre imatges i millor registre tindrem. Depenent de l'optimitzador que escollim, tindrem pitjors o millors resultats depenent també de la configuració dels paràmetres inicials. El valor òptim d'aquest tipus de mètrica és 0.

3.2.1.2 Mètrica amb Informació Mútua

La Informació Mútua (IM) mesura la entropia (aleatorietat) entre dues imatges, és a dir, la probabilitat que el valor d'un píxel qualsevol d'una imatge tingui un valor concret. L'entropia de cada imatge es calcula segons l'equació següent:

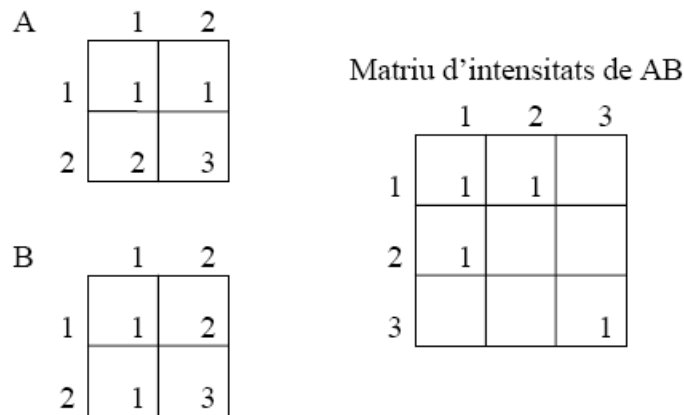
$$H(A) = - \sum_{a=1}^M p_A(a) \log p_A(a)$$

on :

M: número.

$P_A(x)$: probabilitat de trobar un píxel d'intensitat x en la imatge. S'obté a partir de normalitzar l'histograma.

Per calcular la informació mútua cal saber també el valor de l'entropia conjunta entre les dues imatges A i B. Per saber aquest valor tindrem una matriu MxM on les files representaran la intensitat dels píxels de la imatge A i les columnes les de la imatge B i M és el valor d'intensitat màxim de A i de B. D'aquesta forma podrem observar a on la distribució conjunta d'intensitats de les dues imatges. A la Figura 3.4 es pot veure un exemple de dues imatges de 4 píxels.



$$\begin{aligned}
A(1,1) = 1 \text{ i } B(1,1) = 1 &\rightarrow AB(1,1)++ \\
A(1,2) = 1 \text{ i } B(1,2) = 2 &\rightarrow AB(1,2)++ \\
A(2,1) = 2 \text{ i } B(2,1) = 1 &\rightarrow AB(2,1)++ \\
A(2,2) = 3 \text{ i } B(2,2) = 3 &\rightarrow AB(3,3)++
\end{aligned}$$

Figura 3.4. Exemple de càlcul de la taula de l'histograma per calcular l'entropia conjunta de A i B

Un cop coneguem la taula de l'histograma podrem calcular el valor de l'entropia de A i B segons la següent equació:

$$H(A, B) = \sum_{a=1}^M \sum_{b=1}^M p_{AB}(a, b) \log p_{AB}(a, b)$$

Un cop tenim coneguts el valor de l'entropia de A, l'entropia de B i l'entropia conjunta de A i B coneixerem el valor de la informació mútua aplicant la següent equació:

$$MI = H(A) + H(B) - H(A, B)$$

El valor de IM que obtindrem quan més aproximat sigui a 1 vol dir que les imatges són més semblants (si és igual a 1 vol dir que són exactament iguals) i quan menor sigui el valor de IM vol dir que més diferents són (si el valor és 0 vol dir que són completament diferents).

3.2.2 Interpolador lineal

Quan estem registrant, la mètrica és l'encarregada de comparar el nivell d'intensitat de la imatge fixa respecte la imatge moguda transformada i l'interpolador és l'encarregat de poder ajustar la quadrícula del mapejat de punts després d'haver-hi hagut transformacions. Per tant serà l'encarregat d'intentar correlacionar el millor possible el mapejat amb la imatge fixa.

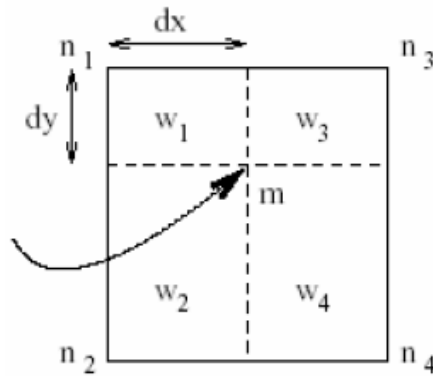


Figura 3.5. Interpolació d'un punt m de la imatge moguda transformada que cau sobre la imatge fixa.

L'interpolador lineal considerarà que la intensitat variarà linealment entre les posicions de la quadrícula i per tant el valor de píxel que associarà correspondrà al pes específic cada valor d'intensitat de píxel (n_i) on hagi caigut el punt.

L'equació següent mostra com es calcula el valor que obtindrà:

$$B(m) = \sum_i w_i B(n_i)$$

on w_i és el pes de cada un dels quatre punts de la quadrícula que es calcularà de la següent manera:

$$w_1 = (1-dx)(1-dy) \quad w_2 = (1-dx)(dy) \quad w_3 = dx(1-dy) \quad w_4 = dx dy$$

Hi han d'altres tipus d'interpoladors més senzills com per exemple l'interpolador del veí més proper que agafa el valor d'intensitat del punt que té més a prop. A l'exemple el punt m hauria agafat com a valor d'intensitat n_1 .

3.2.3 Optimitzador: Gradient descent (gradient descent)

Els optimitzadors en el marc del registre d'imatges, són utilitzats per optimitzar el criteri de la mètrica respecte els paràmetres del transformador. L'objectiu doncs, és minimitzar la similitud (veure Figura 3.6) en funció dels paràmetres del transformador durant cada iteració. Mesura el canvi de la funció de la similitud al voltant dels paràmetres actuals i modificar els paràmetres del transformador cap on aquest canvi (gradient) té un sentit descendent (gradient negatiu).

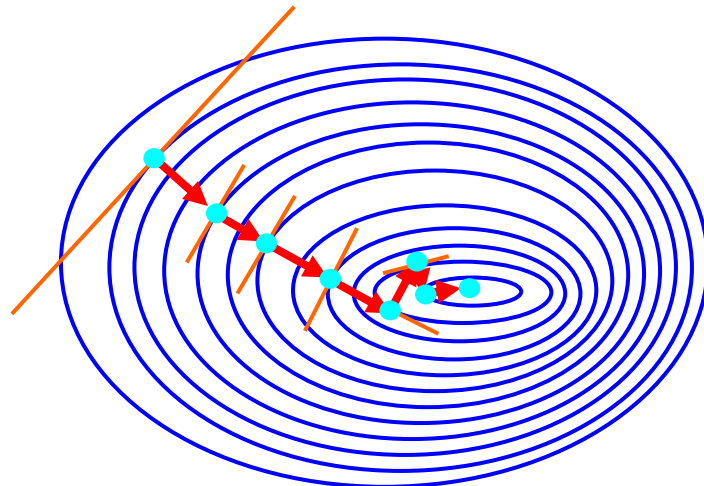


Figura 3.6. Recorregut de l'optimitzador buscant la trajectòria més eficient.

L'optimitzador s'atura quan arriba a un punt màxim d'iteracions o quan no pot trobar valors de gradient negatius.

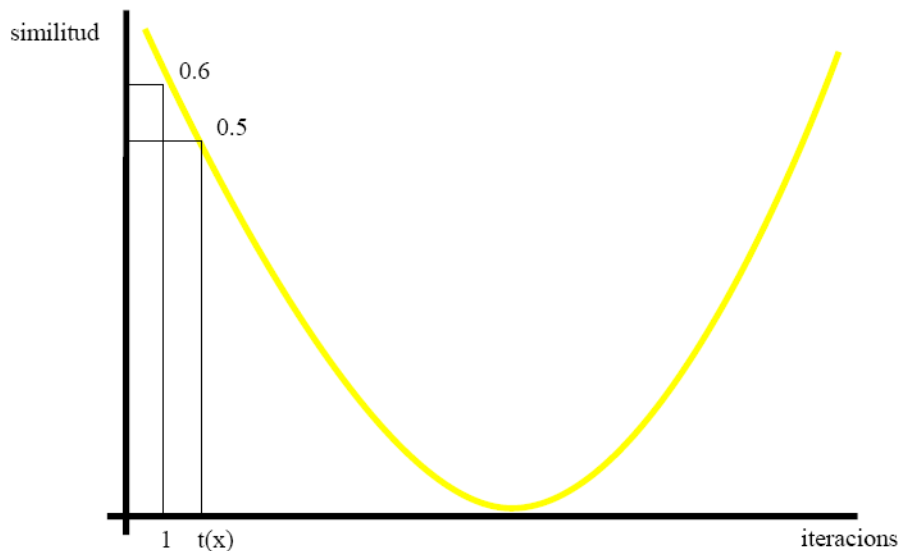


Figura 3.7. Gràfic que mostra un altre exemple del funcionament de l'optimitzador de gradient descendent.

Per exemple, si partim que el valor inicial de similitud és 0.7 el primer pas serà per exemple que $t(x)$ pren per valor 1 i -1. Si el valor de similitud del punt $t(1)$ és menor que el del punt $t(-1)$ agafarem el que tingui el valor de similitud valor més petit (gradient descendent). Seguirem iterant fins que arribem a un punt on finalitzi el nombre d'iteracions establert o fins a un punt on el valor de similitud torni a créixer (com mostra la Figura 3.7).

Tot i que existeixen altres optimitzadors, l'objectiu d'aquest capítol és trobar el transformador i la mètrica que millor s'adapten al nostre problema, i no avaluar els diferents optimitzadors existents.

3.2.4 Transformació rígida 2D i Transformació Afí

El transformador el que farà serà aplicar translacions, rotacions i escalats (segons el mètode escollit) per tal de trobar un mapejat de punts de l'espai de la imatge fixa respecte l'espai de la imatge moguda.

Per obtenir millors resultats s'han utilitzat tipus diferents de transformacions per tal d'obtenir un millor alineat. Els millors resultats que s'han donat han estat utilitzant una transformació rígida 2D centrada i una transformació de similitud 2D centrada. Tot seguit s'explicarà les principals característiques de cadascuna d'elles.

3.2.4.1 Transformació rígida 2D centrada

Representa rotacions en dos dimensions amb un centre arbitrari seguit de translacions en dos dimensions (la transformació rígida no fa escalats). Aquest tipus de transformació ens ha permès poder especificar de forma arbitrària el centre de rotació a diferència d'altres tipus de transformacions que fan les rotacions partint de l'origen de coordenades.

A la pràctica l'usuari pot determinar el centre de rotació, l'angle de rotació i la translació que serà aplicada després de la rotació. Seguidament es mostra l'equació de la transformació d'un punt (x,y).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x - C_x \\ y - C_y \end{bmatrix} + \begin{bmatrix} T_x + C_x \\ T_y + C_y \end{bmatrix}$$

θ és l'angle de rotació
(Cx,Cy) coordenades del centre de rotació
(Tx,Ty) components de la translació

Aquest tipus de transformador només es pot aplicar a imatges de dos dimensions.

3.2.4.2 Transformació Afí

Aquest tipus de transformador representa rotacions, translacions, escalats i retallats. Es diferencia de la transformació rígida comentada al punt anterior, en que conté el factor escalat i pot fer deformacions com estirats. Aquesta transformació és una de les transformacions més utilitzades pel que fa al registre d'imatges.

L'equació següent mostra l'efecte de la transformació d'un punt (x,y).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x - C_x \\ y - C_y \end{bmatrix} + \begin{bmatrix} T_x + C_x \\ T_y + C_y \end{bmatrix}$$

λ és el factor escalar
 θ és l'angle de rotació
(Cx,Cy) coordenades del centre de rotació
(Tx,Ty) components de la translació

La transformació a realitzar s'especifica mitjançant una matriu $N \times N$ (on tindrem les rotacions, escalats i retallats i tindrem un vector $N \times 1$ que representaran les translacions). Aquest tipus de transformador es pot aplicar a imatges de dos o tres dimensions.

3.2.5 Exemple de Registre

En aquesta secció mostrem dos exemples de registre utilitzant dos tipus d'imatges diferents. L'avaluació dels diferents mètodes de registre presentats fins ara es realitzarà en el següent capítol per tal establir quin és el més eficient per la nostra aplicació.

La Figura 3.8 mostra el resultat de registrar imatges confocals utilitzant el mètode AMI.

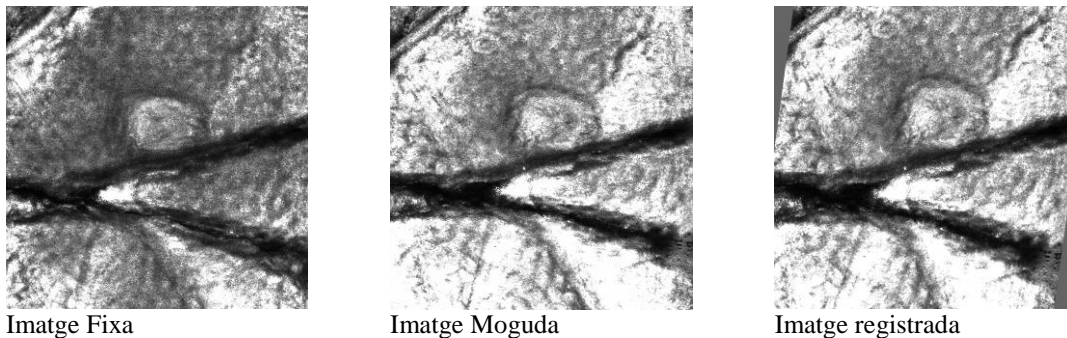


Figura 3.8. Resultats de registrar imatges confocals.

La Figura 3.9 mostra un altre resultat de registrar imatges de ressonància magnètica del cervell utilitzant l'algoritme AMI. Tot i no ser l'objectiu d'aquest treball aquest tipus d'imatges ens ajudaran a avaluar els diferents mètodes de registre.

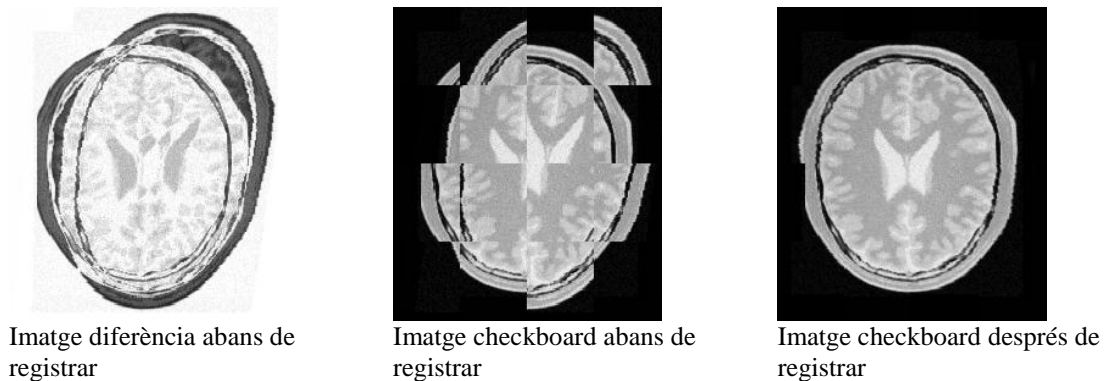


Figura 3.9. Resultats de registrar imatges MRI del cervell.

3.3 Sumari

En aquest capítol hem donat a conèixer els mètodes que utilitzarem per assolir l'objectiu de registrar les imatges 2D, tant en definició com en comportament sobre l'algorisme de registre.

Seguidament veurem les proves de registre realitzades amb tots els mètodes definits i avaluarem quin és el sistema més eficient i que ens dona millors resultats per poder-lo aplicar en el nostre treball amb plenes garanties.

4. Avaluació dels mètodes de registre

Jaume Delgado Fernandez
Enginyeria Tècnica en Informàtica de Sistemes.

Universitat de Girona, 2006

4. Avaluació dels mètodes de registre

4.1 Introducció

En aquest capítol avaluarem els resultats obtinguts en el registre 2D de les imatges de la pell obtingudes amb el microscopi confocal.

S'explicaran els conceptes bàsics referents a l'avaluació del registre d'imatges 2D i sobre els mètodes que ens aportaran informació visual i numèrica sobre l'alineació dels talls de pell. S'avaluaran els diferents mètodes utilitzant diferents tipus d'imatges.

4.2 Mètode per l'avaluació

4.2.1 Diferència d'imatges

La diferència d'imatges ens permetrà saber si dos imatges coincideixen. Per calcular la diferència simplement es restarà els dos talls. Figura 4.1 mostra un exemple de la diferència entre dos imatges.

La diferència s'ha utilitzat com una eina visual per poder comparar la imatge original amb la imatge registrada que s'ha obtingut al executar el registre. Si les dues imatges són idèntiques la imatge diferència serà en blanc. En el cas contrari ressaltarà les zones que no coincideixen.

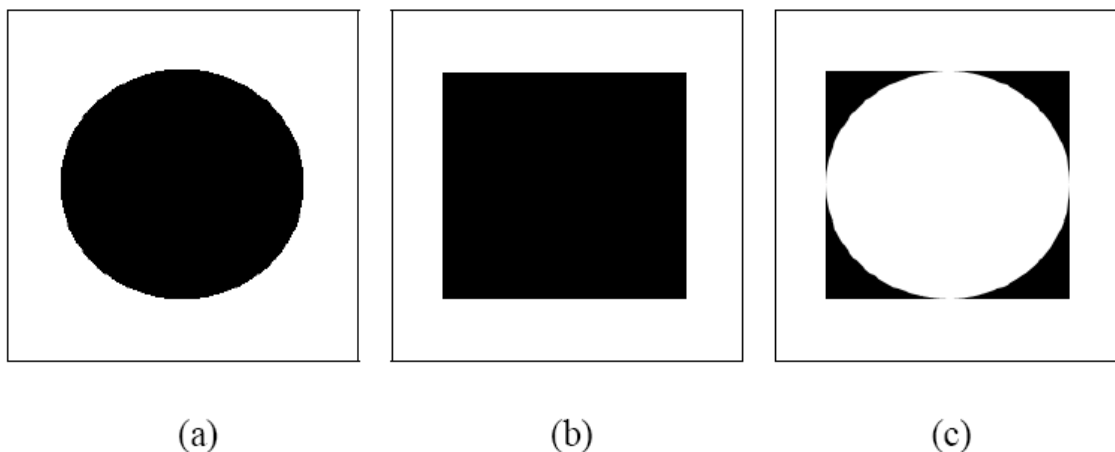


Figura 4.1. (c) és la imatge diferència (a)–(b).

L'algoritme s'ha dissenyat per poder obtenir la imatge diferència entre la imatge original i la imatge capturada en la profunditat següent (imatge moguda).

4.2.2 Checkerboard

El checkerboard ens permet obtenir noves imatges dividides en regions on cada regió és una part d'una de les dos imatges. D'aquesta forma es pot veure la unió de les dues imatges i per tant veure si s'han alineat correctament tal com mostra la Figura 4.2. El checkerboard s'ha calculat abans d'aplicar el registre i després d'aplicar el registre.

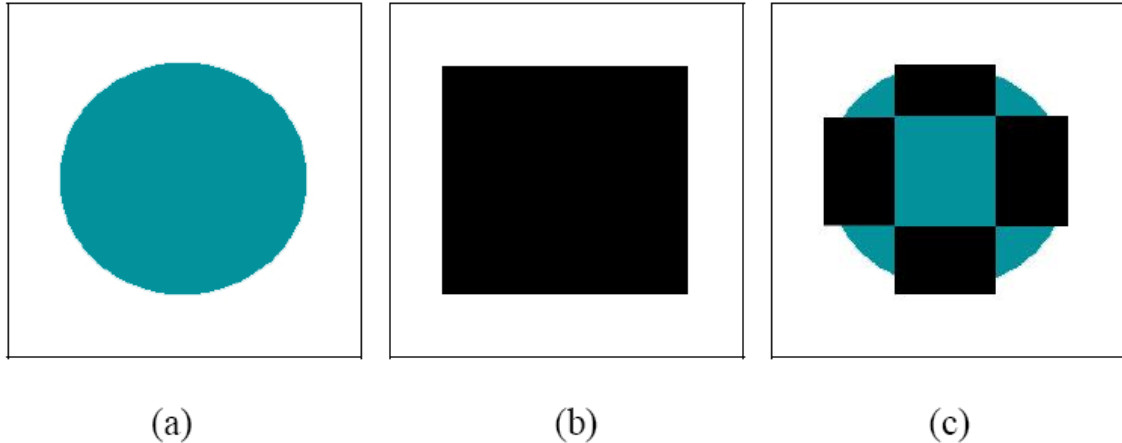


Figura 4.2. (c) és la imatge després d'aplicar-hi al checkerboard composta per quadrants d'(a) i de (b).

4.2.3 Informació Mútua

Aquest algoritme mesurarà la Informació Mútua (MI) entre dues imatges utilitzant el marc de les classes estadístiques de les ITKs. La MI mesura la entropia (desordre) entre dues imatges amb les equacions següents:

$$H(A) = -\sum_{a=1}^M p_A(a) \log p_A(a)$$

$$H(A, B) = \sum_{a=1}^M \sum_{b=1}^M p_{AB}(a, b) \log p_{AB}(a, b)$$

$$MI = H(A) + H(B) - H(A, B)$$

Per més informació sobre la Informació mútua veure l'apartat 3.2.1.2 d'aquesta memòria.

4.2.4 Diferència de quadrats

La mètrica de mitjanes quadrades mesura la diferència de grisos de les dues imatges al quadrat d'una regió determinada.

$$MS(A, B) = \frac{1}{N} \sum_{i=1}^N (A_i - B_i)^2$$

Per més informació sobre la diferència de quadrats veure l'apartat 3.2.1.1 de la memòria.

4.3 PROVES I AVALUACIÓ DELS RESULTATS

Un cop presentades les diferents maneres d'avaluar el registre s'explicaran les proves realitzades en el projecte i els resultats a què s'han arribat. Les proves han estat realitzades amb imatges confocals de la pell i amb imatges de ressonància magnètica del cervell (MRI).

En aquestes proves, s'avaluarà el registre 2D/2D dels talls de pell desalineats que rebem del microscopi amb diferents profunditats, dels quals obtindrem informació com ara, les imatges diferència abans de registrar, els talls diferència després de registrar, varis algoritmes de mesura de similitud (imatges d'informació mútua i de mitjanes quadrades) i el checkerboard d'abans i després del registre. Aquesta avaluació ens permetrà determinar el millor algoritme per aplicar a les imatges confocals.

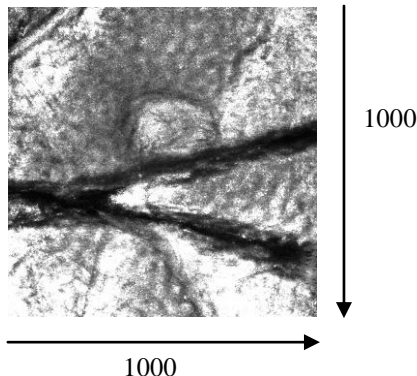
4.3.1 Imatges Utilitzades

El projecte s'ha provat amb diferents tipus d'imatges. Imatges confocals reals i MRI del cervell reals. Les imatges de MRI, tot i no ser objectiu directe d'aquest treball, ens permetran saber si el registre ha estat implementat correctament i ens ajudarà a entendre el funcionament de cada algorisme.

4.3.1.1 Imatges confocals i MRI del cervell

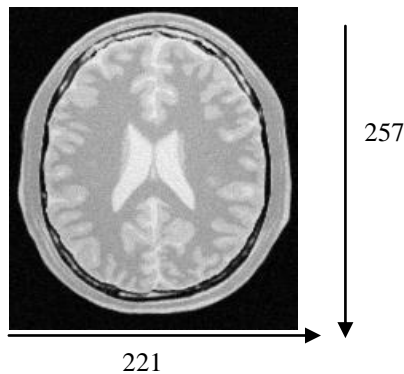
Imatges Pell

- Dimensions: 1000 x 1000
- Tipus: Imatges PNG i BMP
- Mida: aproximadament 978Kb



Imatges MRI

- Dimensions: 221 x 257
- Tipus: Imatges PNG i BMP
- Mida: aproximadament 20Kb



4.3.2 Registre

Tot seguit es mostraran algunes de les imatges més significatives de les diferents proves que s'han utilitzat. Aquest apartat s'ha dividit en dues parts, la part on tindrem els talls de pell del microscopi confocal, la segona els talls reals del MRI del cervell.

La Figura 4.3 els diferents tipus de talls amb els que realitzarem totes les proves sobre el registre 2D/2D.

La primera fila de la Figura 4.3, correspon als talls confocals originals(Talls Oi). La segona fila de la mateixa taula ens mostra els talls originals de MRI del cervell (Talls Ti).

Es a dir, s'agafarà com a imatge fixa el tall F0 i es registrarà amb la resta (M1 - M4). Igualment amb les imatges de MRI .

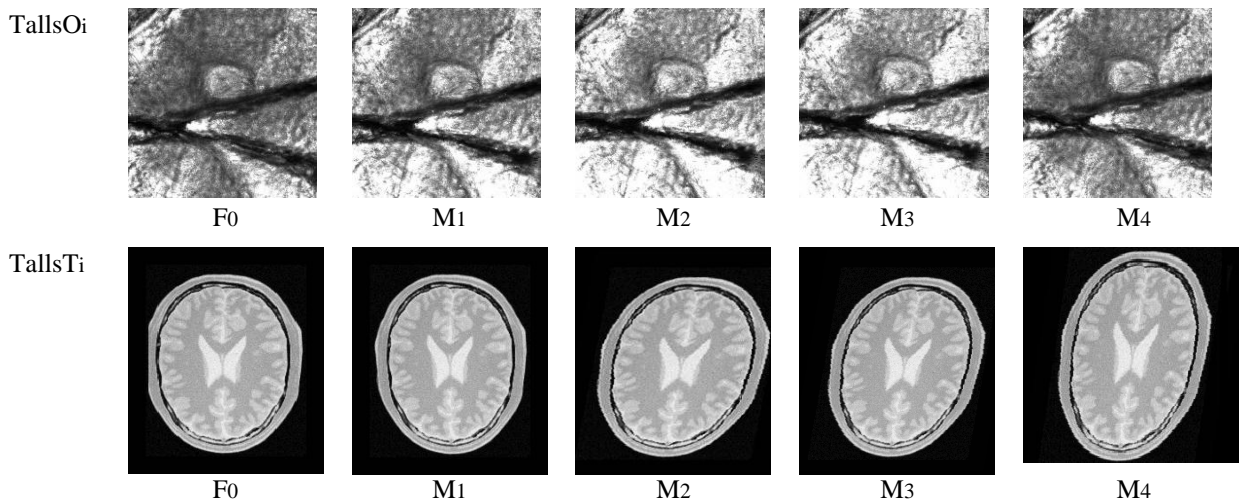


Figura 4.3. Talls confocals i MRI originals.

En les següents figures, les Figures 4.4, 4.5, 4.6 4.7, correspon als talls obtinguts després d'aplicar-hi el registre (talls Ri). Separem cada tipus de registre i mostrem a la primera columna la imatge fixa. La segona columna correspon a la imatge moguda amb la seva profunditat corresponent i la tercera columna hi ha la imatge resultant registrada entre el tall Oi i el tall Mi corresponent.

Les següents figures (Figura 4.8, Figura 4.9, Figura 4.10 i Figura 4.11) mostraran també els resultats amb els talls MRI del cervell

Registre per AMI(Transformació Afí + Informació Mútua)

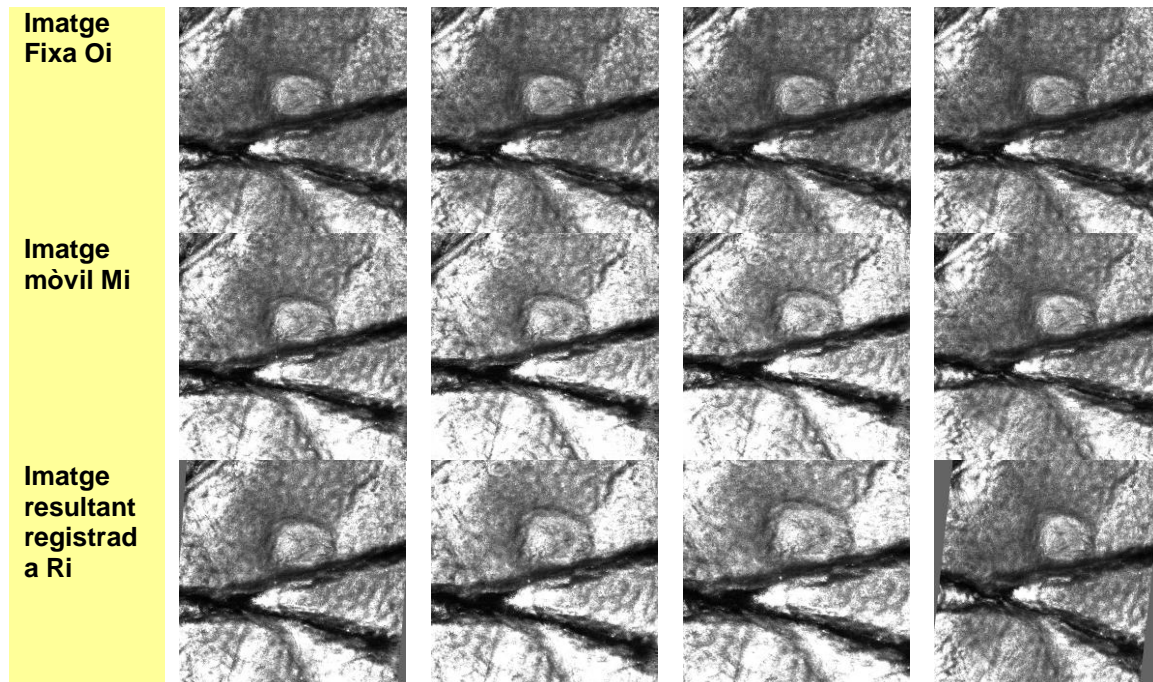


Figura 4.4. Imatges resultants del registre AMI.

Registre per AMSE(Transformació Afí + Mitjanes Quadrades)

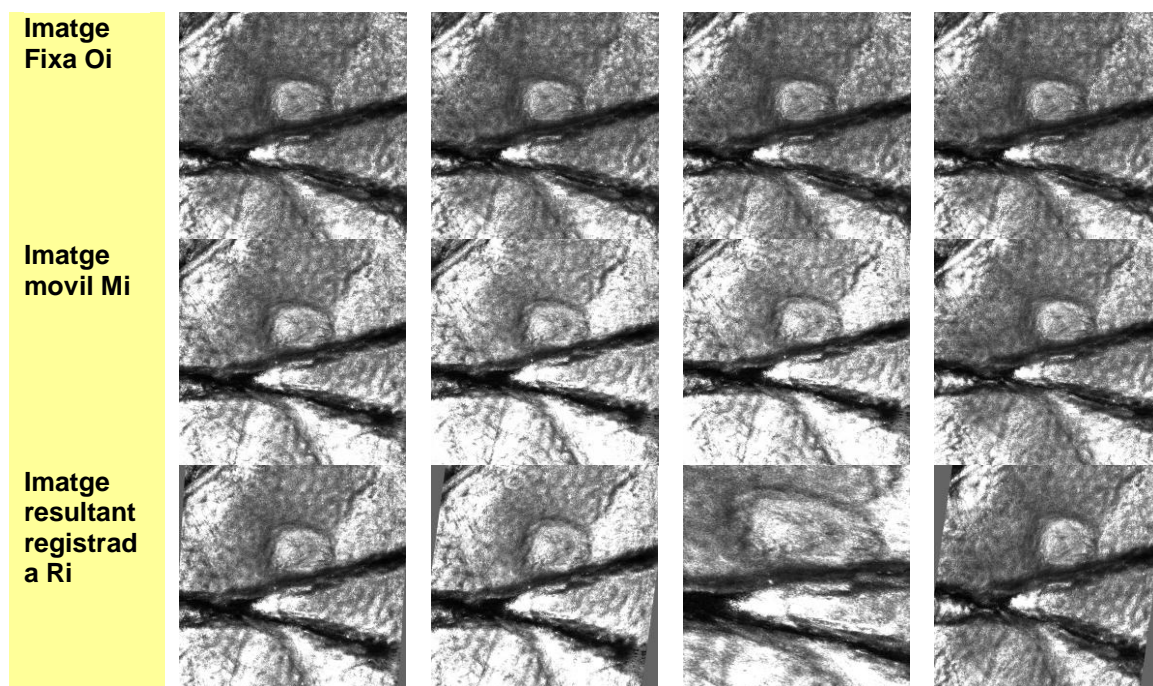


Figura 4.5. Imatges resultants del registre AMSE

Registre RMI(Transformació rígida 2D + Informació Mútua)

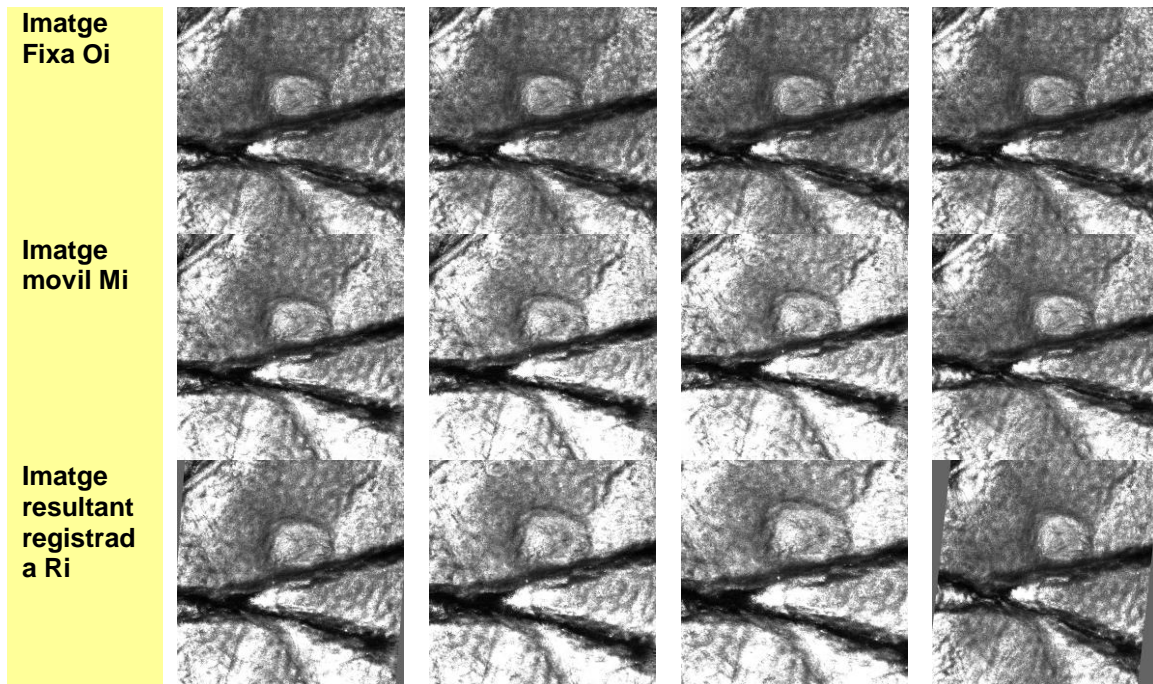


Figura 4.6. Imatges resultants del registre RMI

Registre RMSE(Transformació rígida 2D + Mitjanes quadrades)

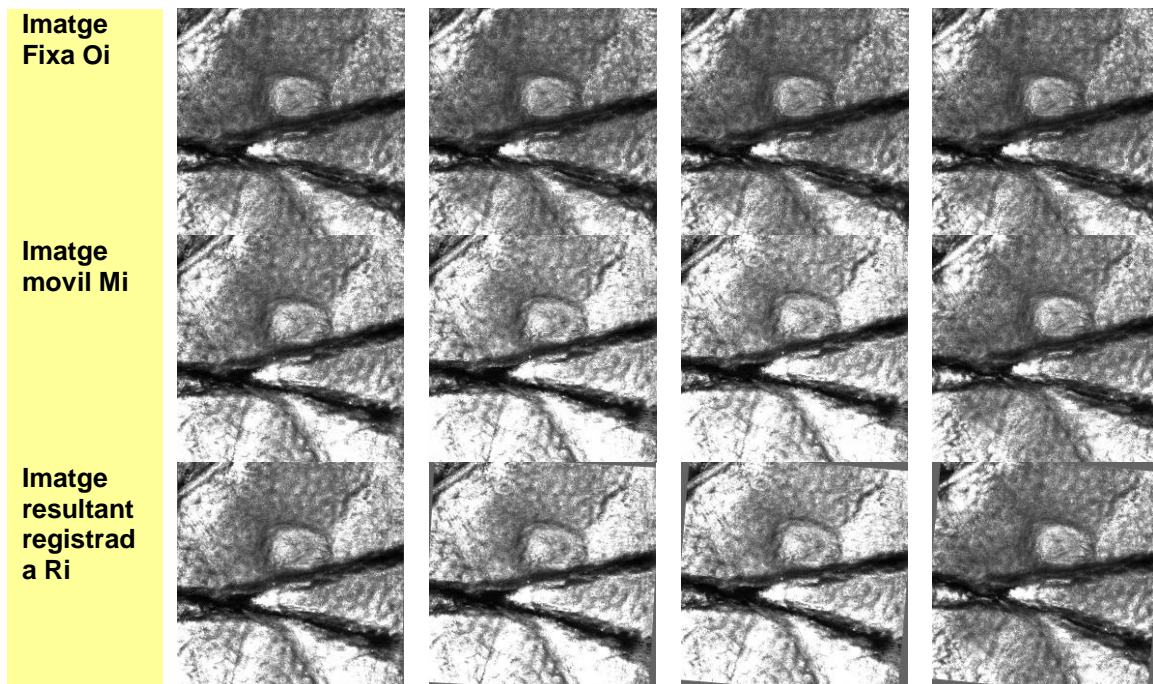


Figura 4.7. Imatges resultants del registre RMSE

Registre per AMI(Transformació Afí + Informació Mútua)

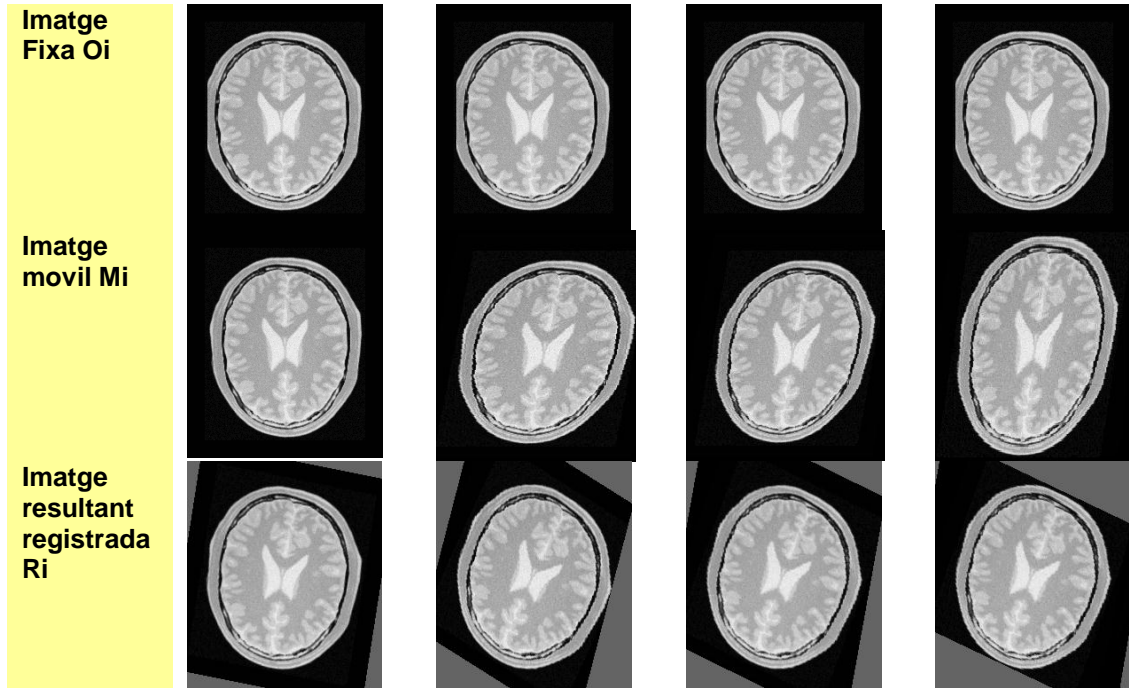


Figura 4.8. Imatges resultants del registre AMI.

Registre per AMSE(Transformació Afí + Mitjanes Quadrades)

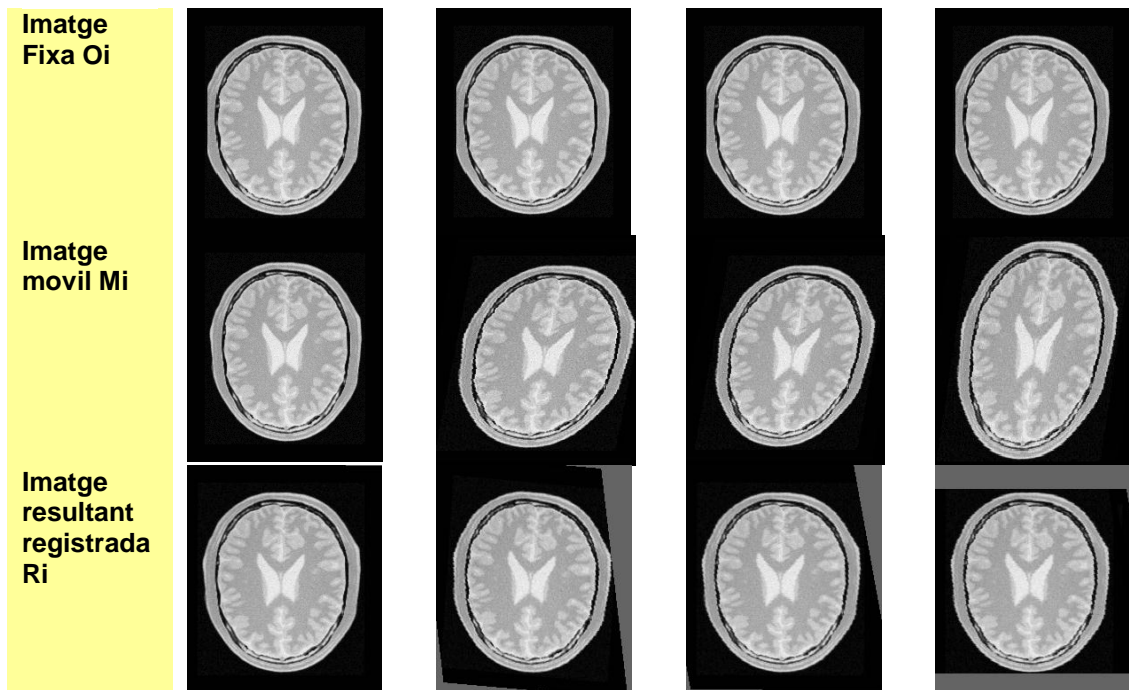


Figura 4.9. Imatges resultants del registre AMSE

Registre RMI(Transformació rígida 2D + Informació Mútua)

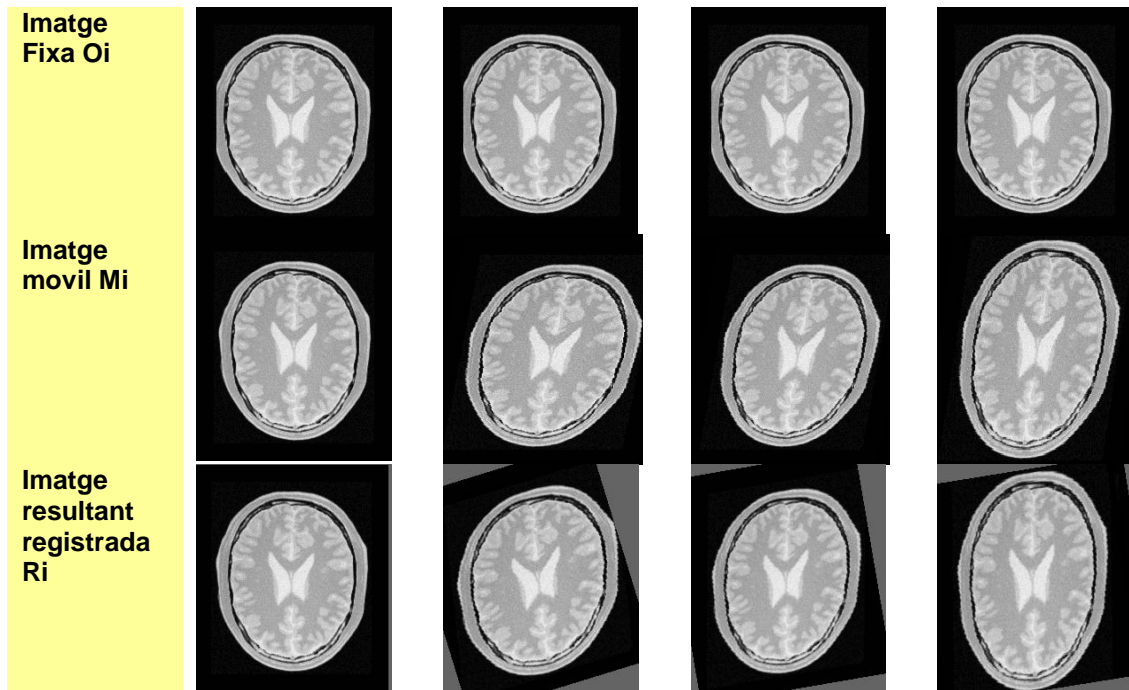


Figura 4.10. Imatges resultants del registre RMI

Registre RMSE(Transformació rígida 2D + Mitjanes quadrades)

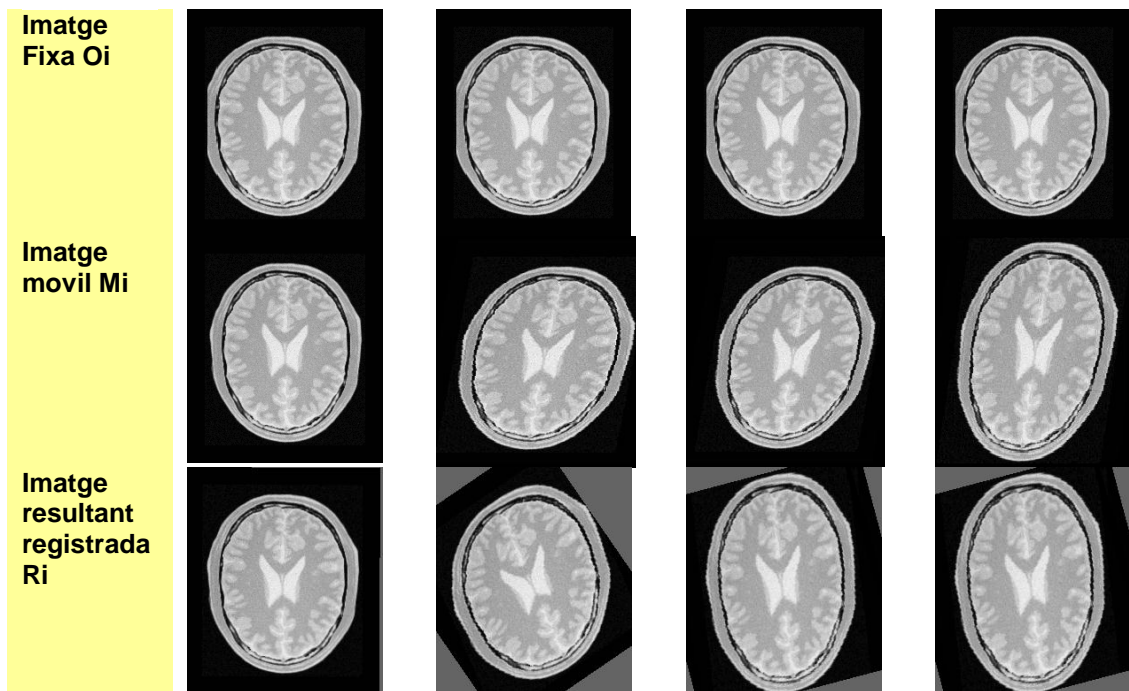


Figura 4.11. Imatges resultants del registre RMSE

4.3.3 Avaluació

Tal com s'ha explicat anteriorment s'han utilitzat diferents algoritmes de registre per alinear (registre per transformació Afí, transformació rígida 2D, mètrica Informació mútua i Mitjanes quadrades). Tot seguit es mostraran resultats quantitius i qualitius de les imatges que s'han obtingut un cop s'ha alineat.

Cal destacar que en aquesta tasca el nostre objectiu és registrar els talls el millor possible amb els diferents mètodes, per poder assolir el següent objectiu de poder reconstruir el volum el més semblant a l'original.

Per saber si s'han registrat correctament, les proves que s'han realitzat han estat la diferència i el checkboard entre el tall original (Tall O_0) amb el tall posterior en profunditat (Tall O_{i+1}) i dades numèriques com la informació mútua (MI) i diferència de quadrats (MSE). Tot seguit hi ha els resultats d'haver aplicat l'alineació entre els talls. Està dividit en taules cadascuna corresponent al tipus de registre aplicat.

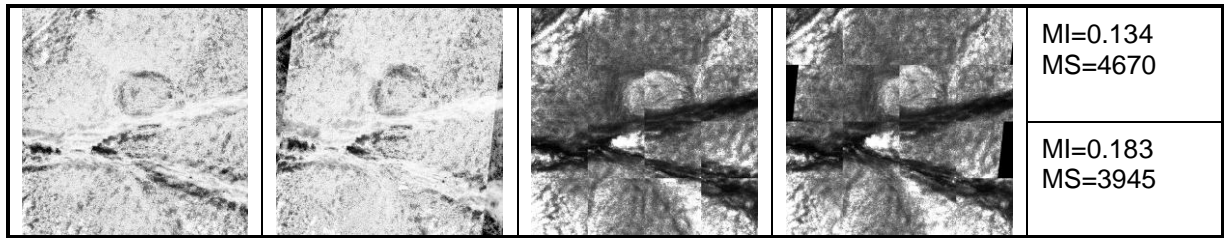
En tots els casos s'observa que la mesura similitud (MI o MS) obté millors resultats (imatges més similars) després del registre, tot i que en alguns casos aquesta millora no és del tot significativa donat que les imatges inicials estan molt ben alineades. Analitzant la imatge diferència també apreciem millores comparat a no registrar. Veieu l'apartat de l'estudi comparatiu per una millor comparació dels resultats en cada mètode.

4.3.3.1 Registre per AMI(Transformació Afí + Informació Mútua)

Calcularem el registre per AMI amb els talls confocals i MRI del cervell.

4.3.3.1.1 Imatges Confocals

Diferència abans	Diferència després	Checkboard abans	Checkboard després	Similitud abans/ després
				MI=0.322 MS=4128
				MI=0.345 MS=3757
				MI=0.201 MS=6863
				MI=0.233 MS=5921
				MI=0.138 MS=7568
				MI=0.180 MS=6253



MI=0.134
MS=4670

MI=0.183
MS=3945

Figura 4.12. Resum del registre AMI amb imatges confocals

Talls	AMI			
	MI		MSE	
	Valor inicial	Valor Final	Valor inicial	Valor Final
1	0.322	0.345	4128	3757
2	0.201	0.233	6863	5921
3	0.138	0.180	7568	6253
4	0.134	0.183	4670	3945

Taula 4.1. Taula resum dels resultats mètrics abans i després d'alinejar amb registre AMI

4.3.3.1.2 MRI Cervell

Diferència abans	Diferència després	Checkboard abans	Checkboard després	Similitud abans/ després
				MI=0.652 MS=1448
				MI=0.691 MS=1093
				MI=0.382 MS=3693
				MI=0.549 MS=1972
				MI=0.413 MS=3273
				MI=0.601 MS=1659
				MI=0.238 MS=6047
				MI=0.589 MS=1763

Figura 4.13. Resum del registre AMI amb imatges MRI del cervell

Talls	AMI			
	MI		MSE	
	Valor inicial	Valor Final	Valor inicial	Valor Final
1	0.652	0.691	1448	1093
2	0.382	0.549	3693	1972
3	0.413	0.601	3273	1659
4	0.238	0.589	6047	1763

Taula 4.2. Taula resum dels resultats mètrics abans i després d'alinejar amb registre AMI

4.3.3.2 Registre per AMSE(Transformació Afí + Mitjanes Quadrades)

Calcularem el registre per AMSE amb els talls confocals i MRI del cervell.

4.3.3.2.1 Imatges Confocals

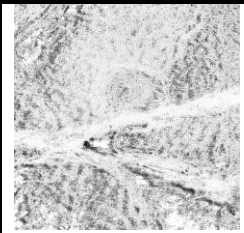
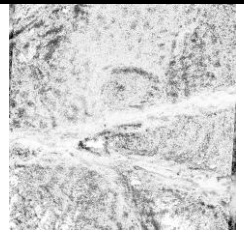
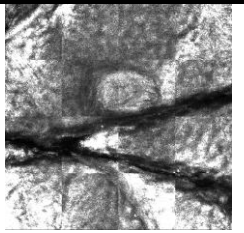
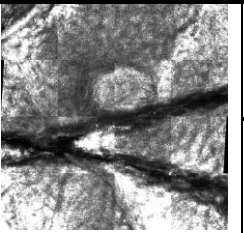
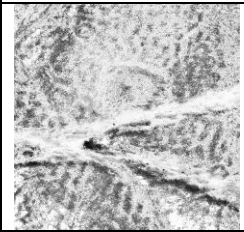
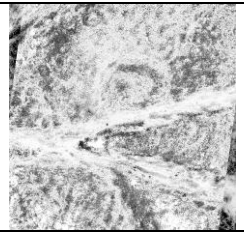
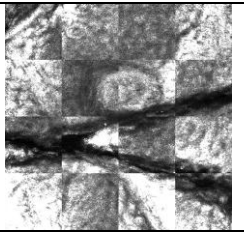
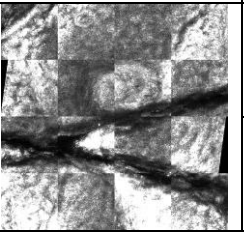
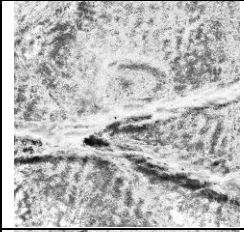
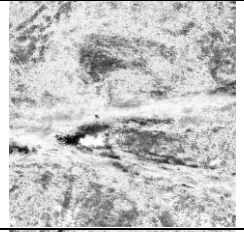
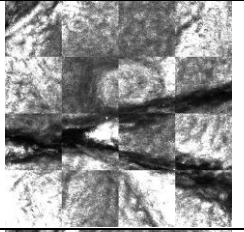
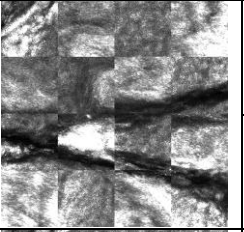
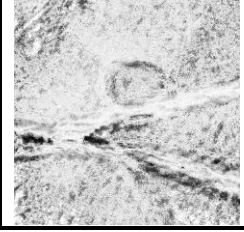
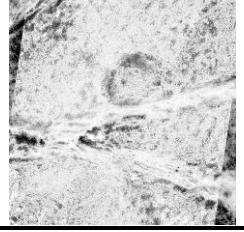
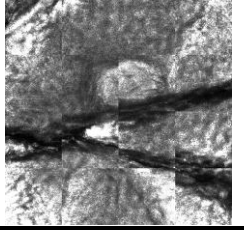
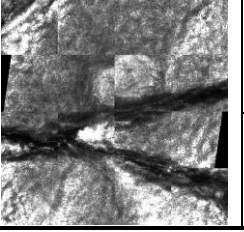

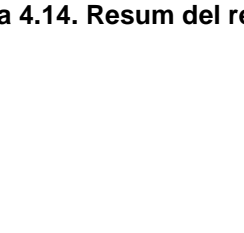
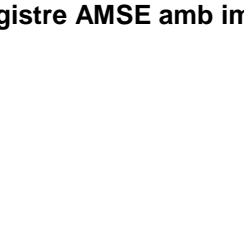





Diferència abans	Diferència després	Checkboard abans	Checkboard després	Similitud abans/ després
				MI=0.330 MS=4128
				MI=0.372 MS=3602
				MI=0.204 MS=6863
				MI=0.273 MS=5726
				MI=0.139 MS=7568
				MI=0.118 MS=6063
				MI=0.135 MS=4670
				MI=0.201 MS=3617

Figura 4.14. Resum del registre AMSE amb imatges confocals

Talls	AMSE			
	MI		MSE	
	Valor inicial	Valor Final	Valor inicial	Valor Final
1	0.330	0.372	4128	3602
2	0.204	0.273	6863	5726
3	0.139	0.118	7568	6063
4	0.135	0.201	4670	3617

Taula 4.3. Taula resum dels resultats mètrics abans i després d'alinejar amb registre AMSE

4.3.3.2.2 MRI Cervell



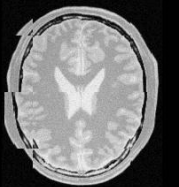
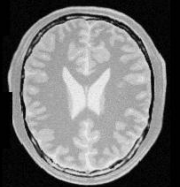


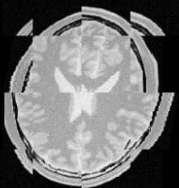
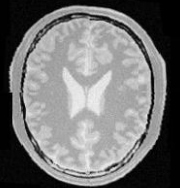
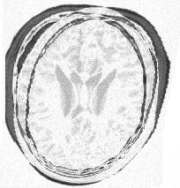
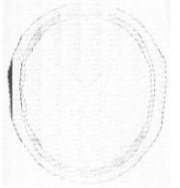
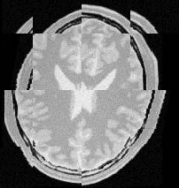
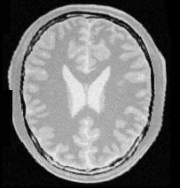
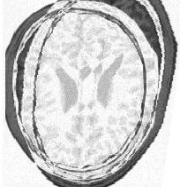
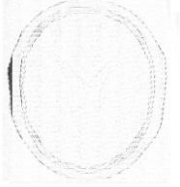

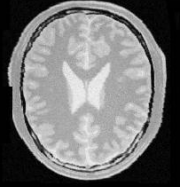












Diferència abans	Diferència després	Checkboard abans	Checkboard després	Similitud abans/ després
				MI=0.648 MS=1448
				MI=1.120 MS=144.2
				MI=0.381 MS=3693
				MI=0.806 MS=581.3
				MI=0.414 MS=3273
				MI=1.095 MS=185.4
				MI=0.237 MS=6047
				MI=1.120 MS=224.6

Figura 4.15. Resum del registre AMSE amb imatges MRI del cervell

Talls	AMSE			
	MI		MSE	
	Valor inicial	Valor Final	Valor inicial	Valor Final
1	0.648	1.120	1448	144.2
2	0.381	0.806	3693	581.3
3	0.414	1.095	3273	185.4
4	0.237	1.120	6047	224.6

Taula 4.4. Taula resum dels resultats mètrics abans i després d'alinejar amb registre AMSE

4.3.3.3 Registre RMI(Transformació rígida 2D + Informació Mútua)

Calcularem el registre RMI tant en el talls confocals com en els talls MRI del cervell.

4.3.3.3.1 Imatges Confocals

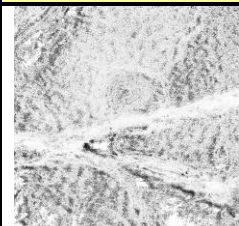
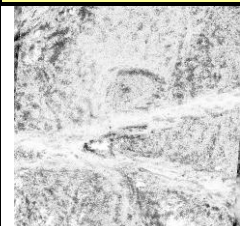
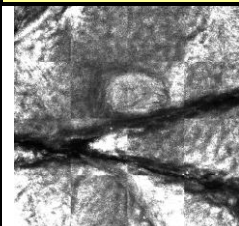
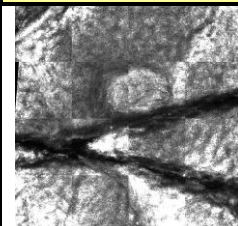
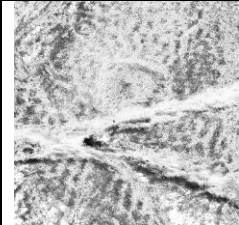
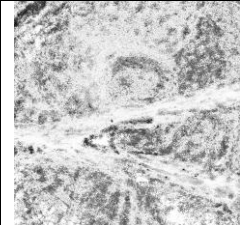
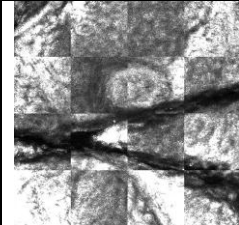
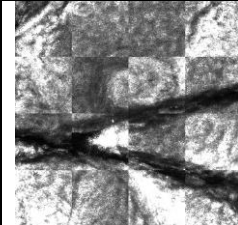
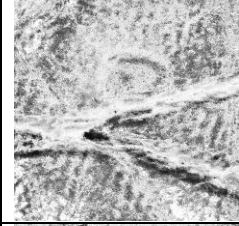
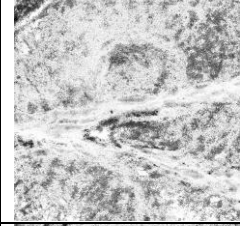
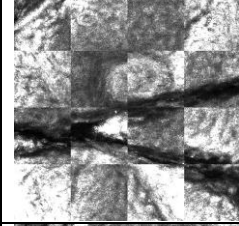
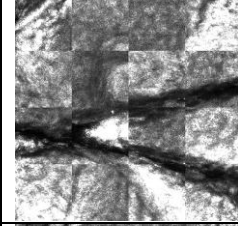
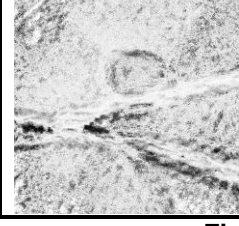
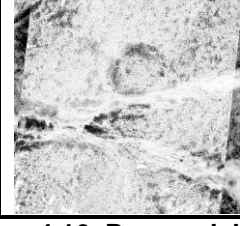
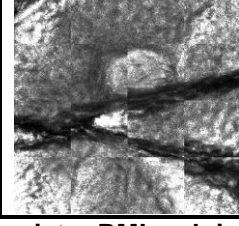
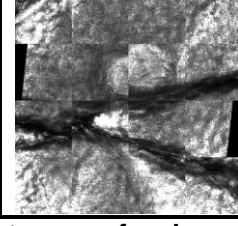

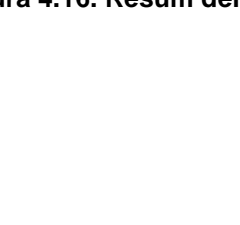
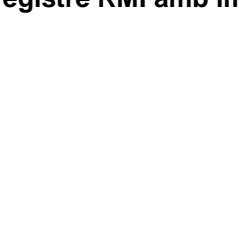
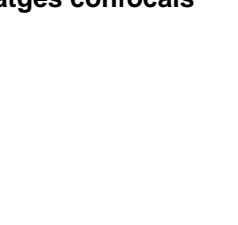




Diferència abans	Diferència després	Checkboard abans	Checkboard després	Similitud abans/ després
				MI=0.322 MS=4128
				MI=0.307 MS=4084
				MI=0.201 MS=6863
				MI=0.205 MS=6624
				MI=0.138 MS=7568
				MI=0.164 MS=7005
				MI=0.134 MS=4670
				MI=0.115 MS=4810

Figura 4.16. Resum del registre RMI amb imatges confocals

RMI				
	MI		MSE	
Talls	Valor inicial	Valor Final	Valor inicial	Valor Final
1	0.322	0.307	4128	4084
2	0.201	0.205	6863	6624
3	0.138	0.164	7568	7005
4	0.134	0.115	4670	4810

Taula 4.5. Taula resum dels resultats mètrics abans i després d'alinejar amb registre RMI

4.3.3.3.2 MRI Cervell


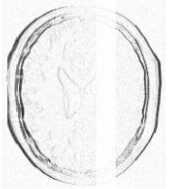
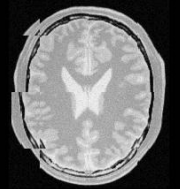
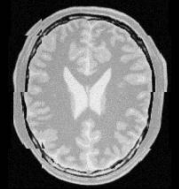


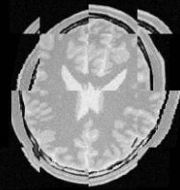



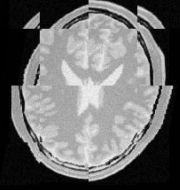
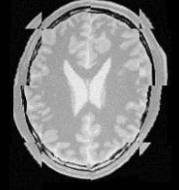
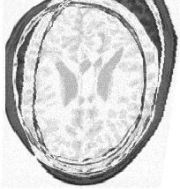


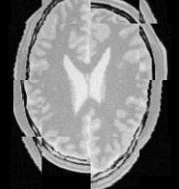
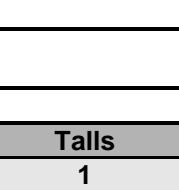
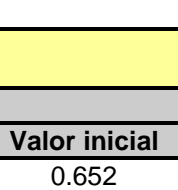
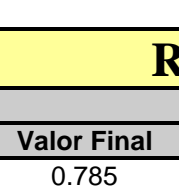
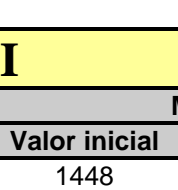
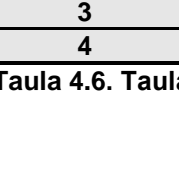
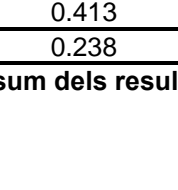
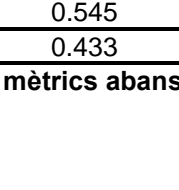
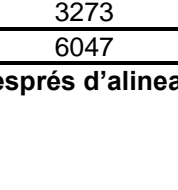




Diferència abans	Diferència després	Checkboard abans	Checkboard després	Similitud abans/ després
				MI=0.652 MS=1448
				MI=0.785 MS=1135
				MI=0.382 MS=3693
				MI=0.785 MS=1135
				MI=0.413 MS=3273
				MI=0.545 MS=2740
				MI=0.238 MS=6047
				MI=0.433 MS=3932

Figura 4.17. Resum del registre RMI amb imatges MRI del cervell

RMI				
	MI		MSE	
Talls	Valor inicial	Valor Final	Valor inicial	Valor Final
1	0.652	0.785	1448	1135
2	0.382	0.490	3693	2911
3	0.413	0.545	3273	2740
4	0.238	0.433	6047	3932

Taula 4.6. Taula resum dels resultats mètrics abans i després d'alinejar amb registre RMI

4.3.3.4 Registre RMSE(Transformació rígida 2D + Mitjanes quadrades)

Calcularem el registre RMSE tant en els talls confocals com en els talls MRI del cervell.

4.3.3.4.1 Imatges Confocals

Diferència abans	Diferència després	Checkboard abans	Checkboard després	Similitud abans/després
				MI=0.330 MS=4128
				MI=0.311 MS=4088
				MI=0.204 MS=6863
				MI=0.202 MS=6581
				MI=0.139 MS=7568
				MI=0.166 MS=7005
				MI=0.135 MS=4670
				MI=0.162 MS=4047

Figura 4.18. Resum del registre RMSE amb imatges confocals

Talls	RMSE			
	MI		MSE	
	Valor inicial	Valor Final	Valor inicial	Valor Final
1	0.330	0.311	4128	4088
2	0.204	0.202	6863	6581
3	0.139	0.166	7568	7005
4	0.135	0.162	4670	4047

Taula 4.7. Taula resum dels resultats mètrics abans i després d'alinejar amb registre RMSE

4.3.3.4.2 MRI Cervell


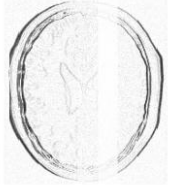

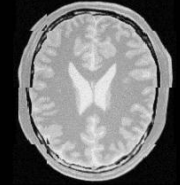

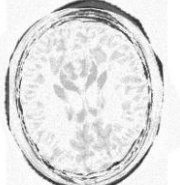
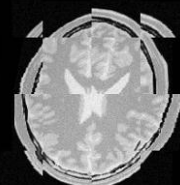
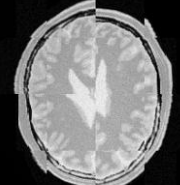
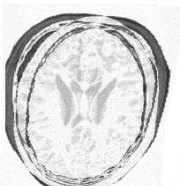

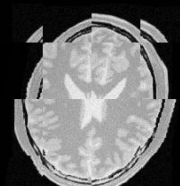
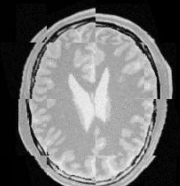
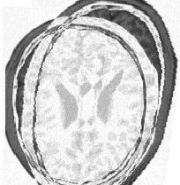

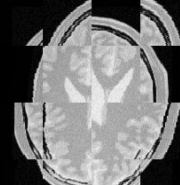
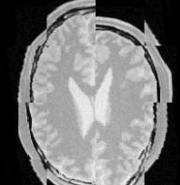
Diferència abans	Diferència després	Checkboard abans	Checkboard després	Similitud abans/ després
				MI=0.648 MS=1448
				MI=0.784 MS=1122
				MI=0.381 MS=3693
				MI=0.495 MS=2506
				MI=0.414 MS=3273
				MI=0.514 MS=2522
				MI=0.237 MS=6047
				MI=0.410 MS=3993

Figura 4.19. . Resum del registre RMSE amb imatges MRI del cervell

Talls	RMSE			
	MI		MSE	
	Valor inicial	Valor Final	Valor inicial	Valor Final
1	0.648	0.784	1448	1122
2	0.381	0.495	3693	2506
3	0.414	0.514	3273	2522
4	0.237	0.410	6047	3993

Taula 4.8. Taula resum dels resultats mètrics abans i després d'alinejar amb registre RMSE

Nota: (AMI) Informació mútua amb Transformació afí.
 (AMSE) Mean Squares amb Transformació afí.
 (RMI) Informació mútua amb Transformació rígida.
 (RMSE) Mean Squares amb Transformació rígida.

4.4 Estudi comparatiu

Donada la quantitat de resultats obtinguts realitzarem un estudi comparatiu per determinar quin és el mètode més adient en imatges confocals. Els estudis realitzats amb les imatges MRI del cervell ens han servit per comprovar que els algorismes creats funcionen correctament, no tan sols amb imatges confocals. En aquest apartat ens centrarem a les imatges de confocal.

Tall	AMI	AMSE	RMI	RMSE
1	MI=0.322 MSE=4128	MI=0.331 MSE=4128	MI=0.322 MSE=4128	MI=0.331 MSE=4128
2	MI=0.202 MSE=6863	MI=0.204 MSE=6863	MI=0.202 MSE=6863	MI=0.204 MSE=6863
3	MI=0.138 MSE=7568	MI=0.139 MSE=7568	MI=0.138 MSE=7568	MI=0.139 MSE=7568
4	MI=0.135 MSE=4670	MI=0.135 MSE=4670	MI=0.135 MSE=4670.4678	MI=0.135 MSE=4670

Taula 4.9. Taula resum dels resultats mètrics abans d'alineat

Tall	AMI	AMSE	RMI	RMSE
1	MI=0.345 MSE=3757	MI=0.372 MSE=3602	MI=0.307 MSE=4084	MI=0.311 MSE=4088
2	MI=0.234 MSE=5921	MI=0.273 MSE=5726	MI=0.206 MSE=6624	MI=0.202 MSE=6581
3	MI=0.180 MSE=6253	MI=0.119 MSE=6063	MI=0.164 MSE=7005	MI=0.166 MSE=7005
4	MI=0.183 MSE=3945	MI=0.201 MSE=3617	MI=0.115 MSE=4810	MI=0.163 MSE=4047

Taula 4.10. Taula resum dels resultats mètrics després d'alineat

Com podem veure en les taules comparatives anteriors les dades de després d'alineat donen resultats satisfactoris. La MI després d'alineat és superior al valor de la MI d'abans d'alineat indicant així que hi ha major relació entre les imatges. El valor serà millor quan més pròxim a 1 sigui.

La diferència de quadrats (MSE) és on més clar queda que s'obtenen molt bons resultats ja que el valor de la MSE d'abans d'alineat és bastant gran i un cop alineat el tall aquest valor queda molt reduït en comparació. Això vol dir que el valor d'error disminueix, com més disminueix millors resultats dona.

Així doncs, el procés d'alineament finalment s'ha realitzat correctament en tots dos tipus d'alineat, proporcionant talls alineats a partir dels originals. Aquesta millora en l'alineació s'observa també qualitativament, tal i com mostrarem a continuació.

RESULTATS AMI vs RMI

Els resultats obtinguts són molt similars, amb imatges de sortida quasi idèntiques. Hem agafat unes quantes imatges de les proves realitzades per veure amb més exactitud, les diferències de registrar o no aplicar el registre.

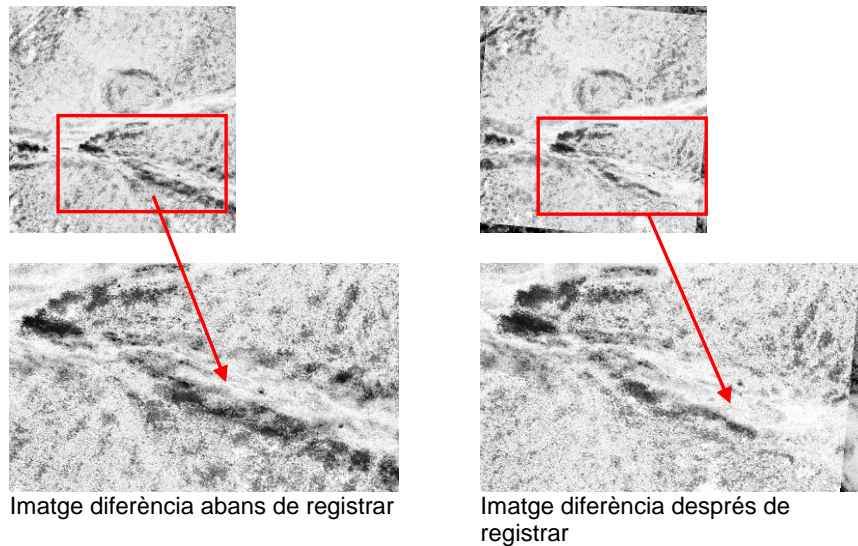


Figura 4.20. Imatges diferència abans i després de registrar amb registre RMI

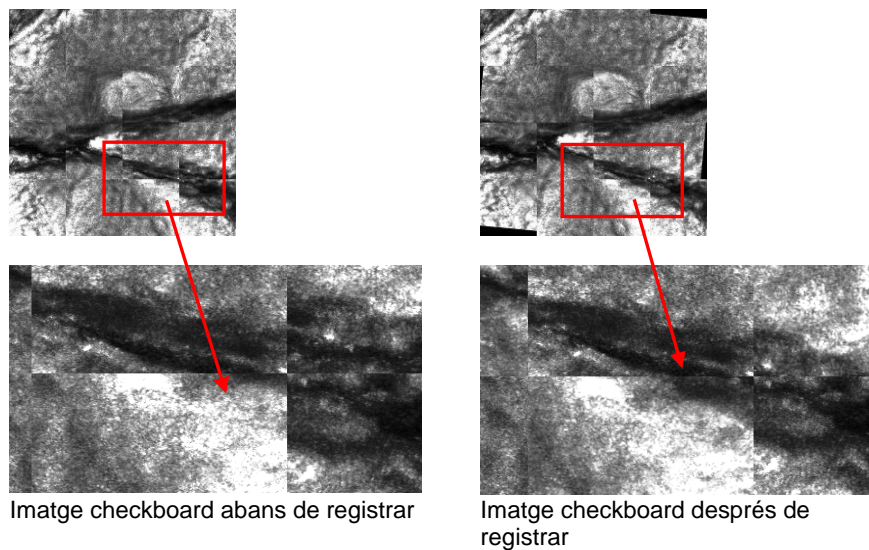


Figura 4.21. Imatges checkboard abans i després de registrar amb registre RMI

Veiem com en les imatges diferència (Figura 4.20) podem veure com després de registrar la imatge és molt més similar a la imatge original (com més similar és la imatge més blanca es veu la zona).

I en les imatges checkboard (Figura 4.21) es pot observar com al registrar la imatge està molt més bé definida que sense registrar. Veiem un bon exemple de la necessitat de registrar, ja que obtenim resultats més òptims que sense registrar.

Inconvenients:

Amb el registre AMI, degut a la transformació Afí, produeix escalats de parts de la imatge, fent que surtin resultats que no són reals. A continuació veiem un exemple (Figura 4.22), com podem veure com la part arrodonida després de registrar és més gran que abans de registrar, fent que surtin resultats que distorsionen la realitat. Això és degut als escalats que produeix les transformacions Afins.

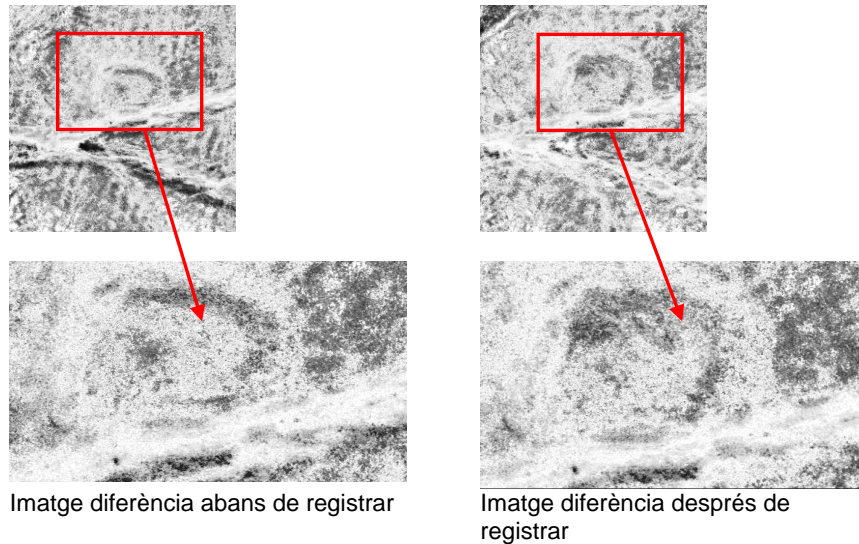


Figura 4.22. Imatges checkboard abans i després de registrar amb registre RMI

RESULTATS RMI vs RMSE

Els resultats obtinguts són molt similars, millora lleugerament RMSE. Observem que la reconstrucció del checkboard són quasi idèntiques, fent que l'apreciació de quina està més definida sigui veritablement difícil.

Gràcies al zoom podem intuir que el registre RMSE és més eficient en la definició dels contorns dels cossos negres, i ajudant-nos de les dades numèriques obtingudes, afirmem aquesta suposició.

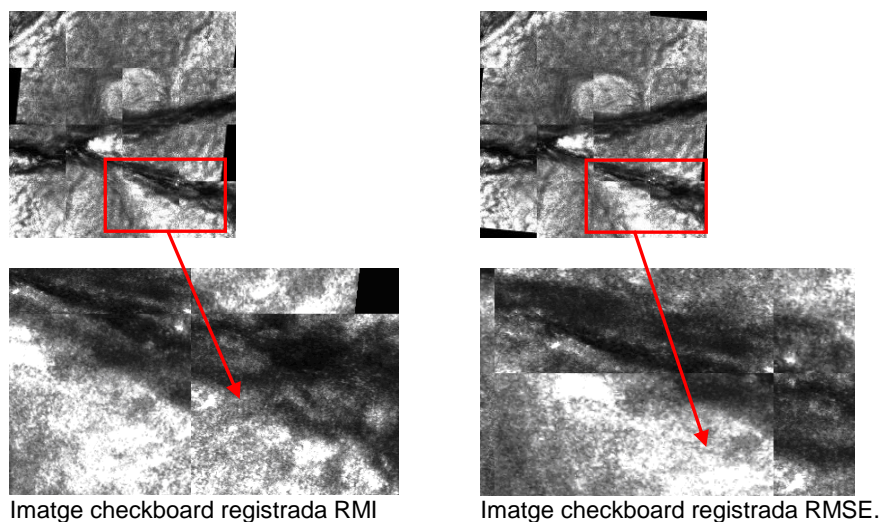


Figura 4.23. Imatges checkboard registrades amb RMI i RMSE

Juntament amb totes aquestes comparatives amb imatges obtingudes del registre, també hem fet un estudi amb gràfics per observar els resultats dels valors obtinguts d'una altra manera, com es veurà seguidament.

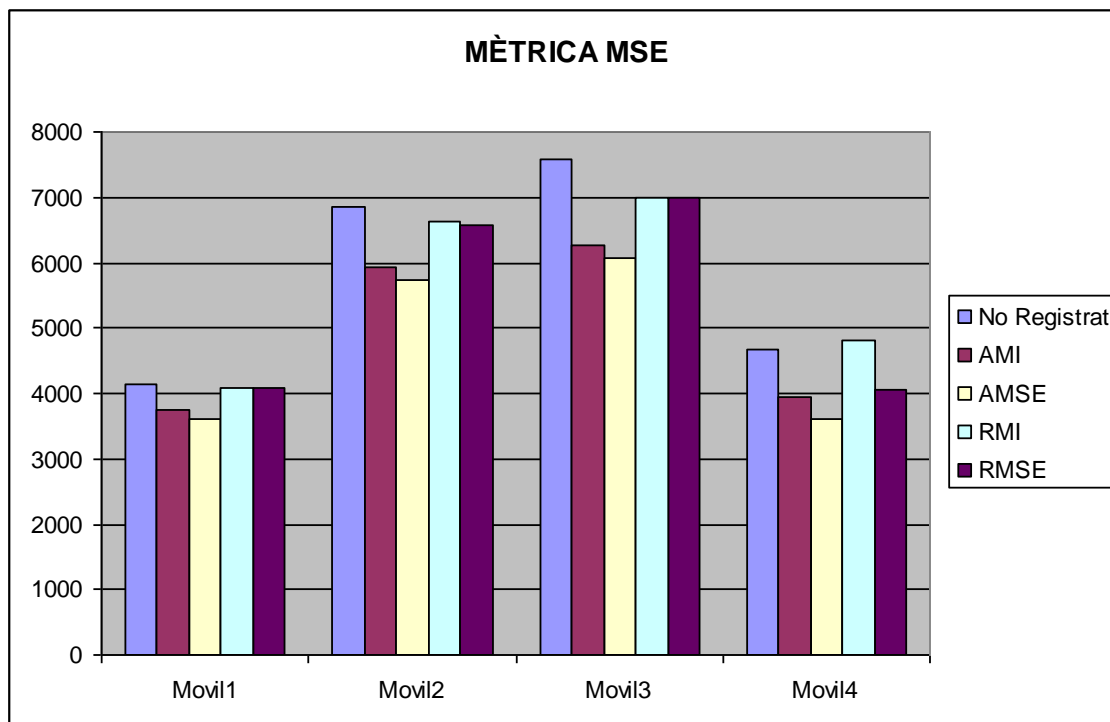


Figura 4.24. Gràfic de les dades obtingudes en el registre utilitzant mètrica MSE.

Aquest gràfic ens mostra quin és l'error al registrar la imatge fixa i les mòbils utilitzant els diferents tipus de registre. Com més elevat sigui el valor, pitjor registrades estaran les imatges.

Amb les dades que ens representa aquest gràfic, les nostres conclusions són que registrar les imatges ens dona menys error (valor mètric de MSE menor) que No registrar els talls. Podem observar que la primera columna "No Registrar" sempre té un valor (Error) més elevat que les altres columnes. Després entre els tipus de registre veiem que la metodologia que ens aporta millors resultats amb aquesta mètrica és el registre AMSE. Pel nostre registre, aquesta dada no és molt fiable ja que al utilitzar transformació afí ens escala i deforma zones de la imatge, llavors en el nostre cas, no ens millora el registre. Aquest fet ens porta a pensar en la utilització de transformacions rígides com a compromís entre millora del registre i eliminació de transformacions no desitjades.

Amb les dades d'aquest gràfic i la informació del següent, obtindrem la informació visual i numèrica necessària per tenir una idea més adequada de la tècnica més eficient.

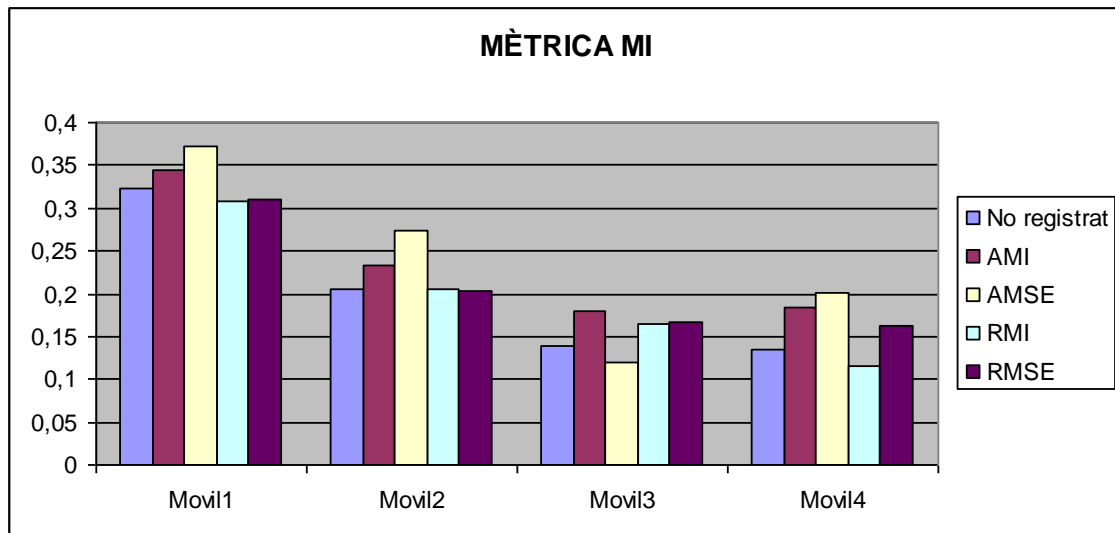


Figura 4.25. Gràfic de les dades obtingudes en el registre utilitzant mètrica AMI

Aquest gràfic ens mostra quina és la similitud entre la imatge fixa i les mòvils utilitzant els diferents tipus de registre i No registrar. Com més elevat sigui el valor, millor registrades estaran les imatges. Amb les dades que ens representa aquest gràfic, les nostres conclusions són que registrar les imatges ens dona millors valors de similitud (valor mètric de AMI major) que No registrar els talls. Podem observar que la primera columna "No Registrar" sempre té un valor més petit que les altre columnes.

Després entre els tipus de registre veiem que la metodologia que ens aporta millors resultats amb aquesta mètrica és el registre AMSE, juntament amb els mètodes rígids.

Però després, comparant les dades dels gràfics amb les dades visuals que ens dona les imatges diferència i checkboard la conclusió que extrèiem és que el registre AMSE és el que ens dona millors resultats numèrics, però quan observem els resultats visuals ens donem compte que el registre AMSE distorsions les imatges fent que el registre no sigui interessant pel nostre projecte. Així, la millor opció que tenim és el registre Rígid RMSE, que és el que ens aporta la millor mitjana de resultats entre visuals i numèrics.

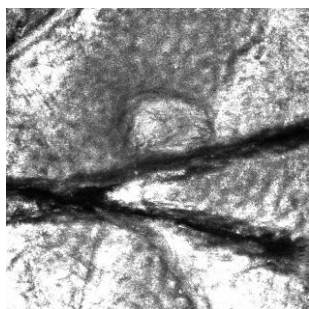
4.5 CONCLUSIONS

Tan els valor visuals (checkerboard i diferència) i els valors numèrics milloren en tots els registres. La imatge checkerboard de després d'haver-se alineat queda més ben definida respecte el checkerboard d'abans d'alineat-se.

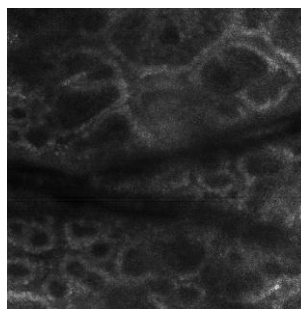
El registre que s'ha fet servir amb transformació Afí s'han obtingut pitjors resultats degut a que l'optimització ha obtingut paràmetres de transformació no gaire realistes (escalats i deformacions). Aquest fet ens porta a concloure que la transformació Afí té més graus de llibertat del que són necessaris per el nostre problema de registre. Utilitzant el registre rígid és on s'han obtingut millors resultats, mantenint la dimensió i el volum del tall com es pot veure i alineant correctament a cada tall.

Amb el tema de registrar o no registrar, proposem que el registrar les imatges ens dóna millors resultats que no alineat els talls.

Inicialment els talls inicials i del mig s'alineaven correctament, en canvi, els talls més exteriors de la imatge (últims talls) s'han anat alineant de forma no gaire adequada, ja que són imatges molt diferents amb els talls del principi. Al estar en profunditats elevades la quantitat d'informació que es pot observar és mínima, i això fa que la similitud amb els altres talls sigui quasi nul·la, fent del registre una tasca complicada. La Figura 4.26 il·lustra aquest fet amb imatges a diferents profunditats de la mateixa regió d'interès. Aquest problema es resoldrà en el capítol de reconstrucció, on proposem realitzar el registre entre grups de talls 2D propers entre ells.



Imatge original amb poca profunditat



Imatge original amb profunditat elevada.

Figura 4.26. Imatges originals amb diferents profunditats .

Després d'haver fet totes les proves anteriors amb diferents mètodes de registre, i per tal de deduir si era millor registrar o no registrar per reconstruir el volum 3D de la pell, vam arribar a la conclusió que l'opció més eficient pel nostre PFC era utilitzant el registre RMSE (Registre Rígid amb mètrica de mitjanes quadrades), ja que no degrada les imatges amb deformacions o esclats irrealistes que farien obtenir resultats incorrectes per l'estudi de les imatges de la pell i obté un millor resultat que no registrant.

4.6 Sumari

En aquest capítol hem avaluat diferents mètodes de registre i extret conclusions sobre quin és el més eficient per la nostra tasca. Ara ja podem estudiar el procés de reconstrucció del volum tridimensional, que serà l'objectiu del capítol següent.

5. Reconstrucció del volum 3D

Jaume Delgado Fernandez
Enginyeria Tècnica en Informàtica de Sistemes.

Universitat de Girona, 2006

5.Reconstrucció del volum 3D

5.1 Objectius

La informació del microscopi confocal sobre la profunditat dels diferents talls i el procés de registre d'aquests permeten obtenir les dades necessàries per tal de reconstruir-ne el volum. La representació del volum de la pell en 3 dimensions no només serà útil per l'ajuda en la interpretació de les imatges MCF, sinó que permetrà ser utilitzat per tal de millorar el diagnòstic del càncer de pell per als dermatòlegs.

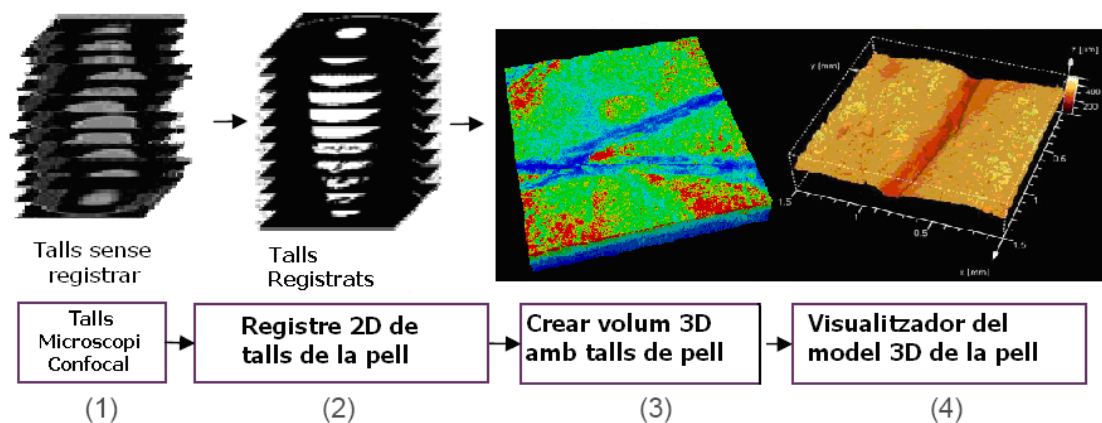


Figura 5.1. Esquema del procés de Visualització.

Així doncs, com podem veure a la Figura 5.1, la fase de reconstrucció volumètric pren com a entrada el resultat obtingut a la fase de registre 2D i les dades provinents de l'escàner MCF.

5.2 Adquisició i Microscopi Confocal

En aquest apartat narrarem com adquirim les imatges mèdiques de la pell a través del microscopi confocal juntament amb la seva informació que ve donada en un arxiu de text adjuntat a cada imatge, i explicarem el funcionament d'aquesta eina en l'estudi del càncer de pell.

5.2.1 Microscopi Confocal

Comunament es coneix per LSCM (laser scanning confocal microscopy) i s'ha establert com a una eina per obtenir imatges d'alta resolució i reconstruccions 3-D d'una gran varietat de mostres biològiques. [NCI, 2005]

La font d'excitació consisteix en un làser el qual s'enfoca en un petit punt d'una mostra fluorescent. La llum reflectida i la fluorescència emesa són capturades pel mateix objectiu i dirigides cap un fotodetector que convertirà el senyal lluminós en un senyal digital. Abans del fotodetector existeix un petit diafragma (pinhole) que evitarà que la fluorescència emesa per punts de la mostra que no estan en el pla focal quedi obstruïda, de tal manera que obtindrem la informació d'un sol pla focal. Això és de gran utilitat especialment quan es treballa amb mostres gruixudes (fins 150 μm). La Figura 5.2 mostra una imatge d'un microscopi confocal.



Figura 5.2. Imatge d'un microscopi Confocal.

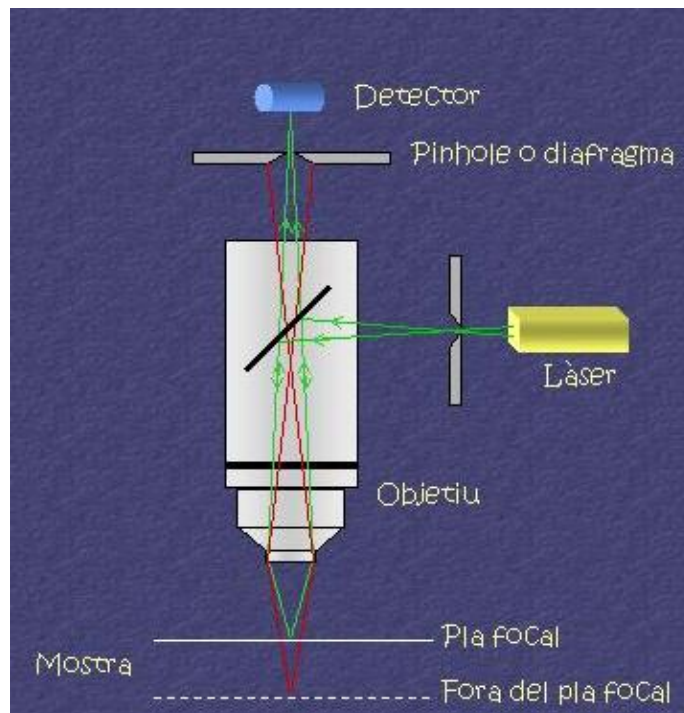


Figura 5.3. Imatge del funcionament del microscopi confocal

Com podem veure a la Figura 5.3, el feix del làser és dirigit a un punt concret del pla focal. La llum reflectida per la mostra, o la fluorescència emesa, és dirigida i captada per un fotodetector, una vegada travessa el diafragma. La imatge focal s'obté després de rastrejar la mostra (scanning) amb el làser. Les imatges obtingudes són sempre imatges digitals: els fotodetectors (PMTs) transformen el senyal lumínic en un senyal elèctric, que mitjançant el sistema informàtic acoblat, es tradueix en un píxel. Cada píxel proporciona informació referent a la localització tridimensional del punt excitat, així com la intensitat lumínica d'aquest punt.

El microscopi confocal és capaç d'escombrar diferents plans en l'eix Z mitjançant un motor que es mou cap amunt i a baix (fins seccions de 0.1 μm). Posteriorment es podrà generar una reconstrucció 3-D apilant una sèrie de seccions òptiques 2-D.

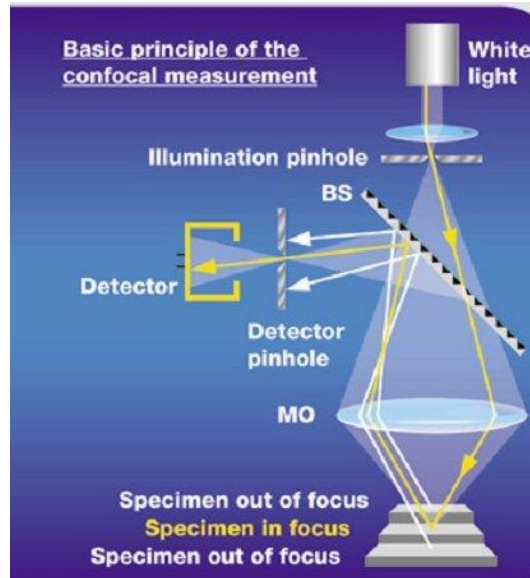


Figura 5.4. Mesura del microscopi confocal

Les principals aplicacions de la microscòpia confocal són, entre altres: estudis d'expressió i localització de molècules en 2 o 3 dimensions, reconstruccions tridimensionals, estudis de localització, estudis de dinamisme de molècules in vivo, mesures de concentracions de ions intracel·lulars in vivo, estudis de desenvolupament, etc. Podem veure un exemple de reconstrucció tridimensional en la següent figura (Figura 5.5) utilitzant el programa de visualització Paraview. [Paraview]

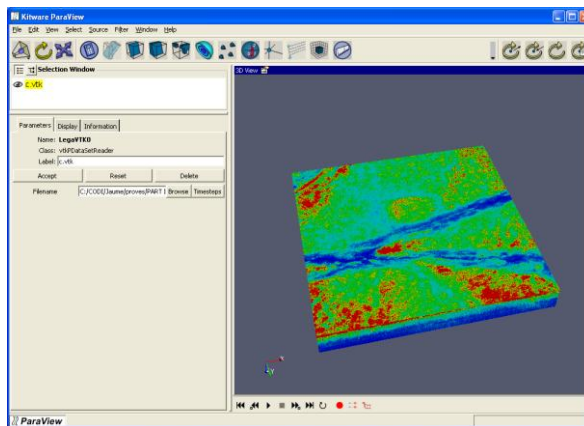


Figura 5.5. Exemple de reconstruccions tridimensionals

5.2.2 Talls confocals i el seu format.

Gràcies al microscopi confocal podem adquirir una col·lecció d'imatges d'una mostra, com pot ser el cas de la pell. Aquestes són imatges en 2 dimensions de zones molt fines del cos que estiguem estudiant a diferents profunditats. A més podem obtenir un gran nombre d'imatges amb el que la zona d'interès quedi totalment definit per al seu estudi. Aquestes imatges les podem utilitzar per a obtenir una representació en 3 dimensions.

5.2.2.1 Imatges amb arxius *.txt

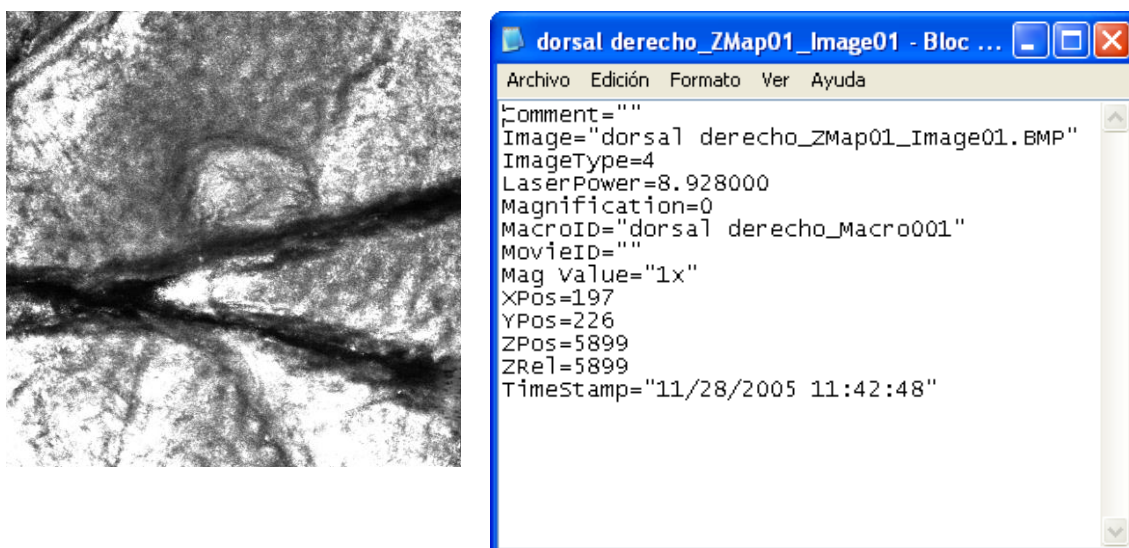


Figura 5.6. Imatge Original de la pell i contingut del fitxer de text.

Les imatges que ens proporciona el microscopi confocal són imatges 2D en format BMP o PNG, acompanyat del corresponent arxiu de text on ens dona tota la informació referent al tall obtingut.

Aquests fitxers normalment es componen en tretze parts, però per el nostre treball les parts més importants són el nom i format de la imatge, la posició en X,Y i sobretot en Z (que ens dirà la profunditat de la pell). El format d'aquesta informació es mostra en la següent taula (Taula 5.1).

Files	Contingut
Comentari	Comment=""
Nom imatge	Image="dorsal derecho_Zmap01_Image14.BMP"
Tipus imatge	ImageType=4
Potència Làser	LaserPower=14.976000
	Magnification=0
	MacroID="dorsal derecho_Macro001"
Identificador	MovieID=""
	Mag Value="1x"
Posició en X	Xpos=197
Posició en Y	Ypos=226
Posició en Z	Zpos=5984
Posició relativa en Z	Zrel=5984
Data i hora de captació de la imatge	TimeStamp="11/28/2005 11:42:48"

Taula 5.1. Taula descriptiva del contingut del fitxer de text.

5.3 Registre d'imatges 2D

En aquesta primera tasca, el que obtindrem serà un conjunt de talls desalineats corresponents a les imatges mèdiques de la pell, i per tant, el que ha de fer aquesta tasca, és alinear cadascun d'aquests talls de forma que puguem alinear/transformar el volum tal com mostren les imatges de la Figura 5.7 amb l'objectiu de poder tenir els talls obtinguts del microscopi confocal amb diferents profunditats i temps de captura perfectament alineats.

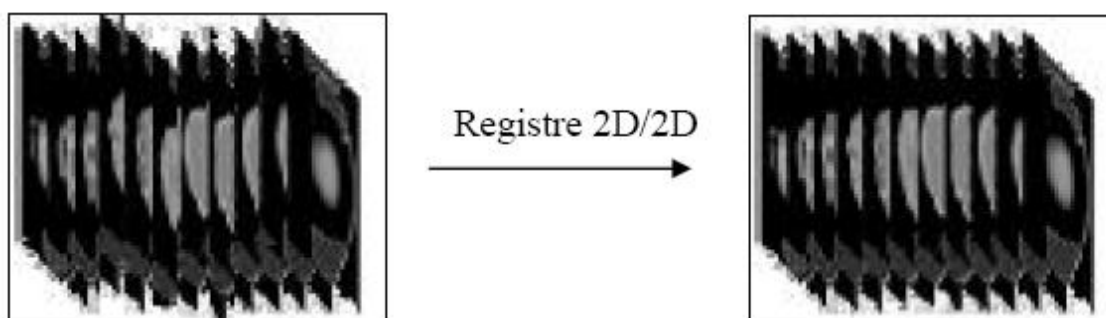


Figura 5.7. Pas de les imatges desalineades obtingudes amb el microscopi confocal on mitjançant un registre 2D/2D de cada tall obtindrem l'alineament dels talls.

Després d'haver fet totes les proves necessàries i programades en quina opció de registre era la més adequada pel nostre registre de imatges de la pell, i per deduir si era millor registrar o no registrar per reconstruir el volum 3D de la pell, varem arribar a la conclusió que l'opció més eficient pel nostre PFC era utilitzant el registre RMSE (Registre Rígid amb mètrica de mitjanes quadrades), tal com s'ha mostrat en el capítol 4. Llavors totes aquestes variants de registre que us mostrarem en aquests apartats seran a partir del registre RMSE. Seguidament es descriuen diferents maneres de fer el registre en els talls 2D.

5.3.1 Registre amb varies imatges referència

Explicarem la metodologia d'aquest algorisme per registrar cada tall 2D de pell. El supòsit inicial és que a cada iteració tindrem una nova imatge de referència. Inicialment agafarem el tall inicial (tall 0) com el fixe, el registrarem amb el tall posterior. Un cop tinguem aquestes imatges registrades, la imatge resultant l'agafarem com de referència i l'utilitzarem per registrar el tall adjacent. Això seguirem fent fins al tall final. Quan tots els talls estiguin registrats podrem dir que tenim els talls alineats i per tant, podrem fer la reconstrucció tridimensional.

En la següent figura podem veure com seria el funcionament d'aquest algorisme (Figura 5.8).

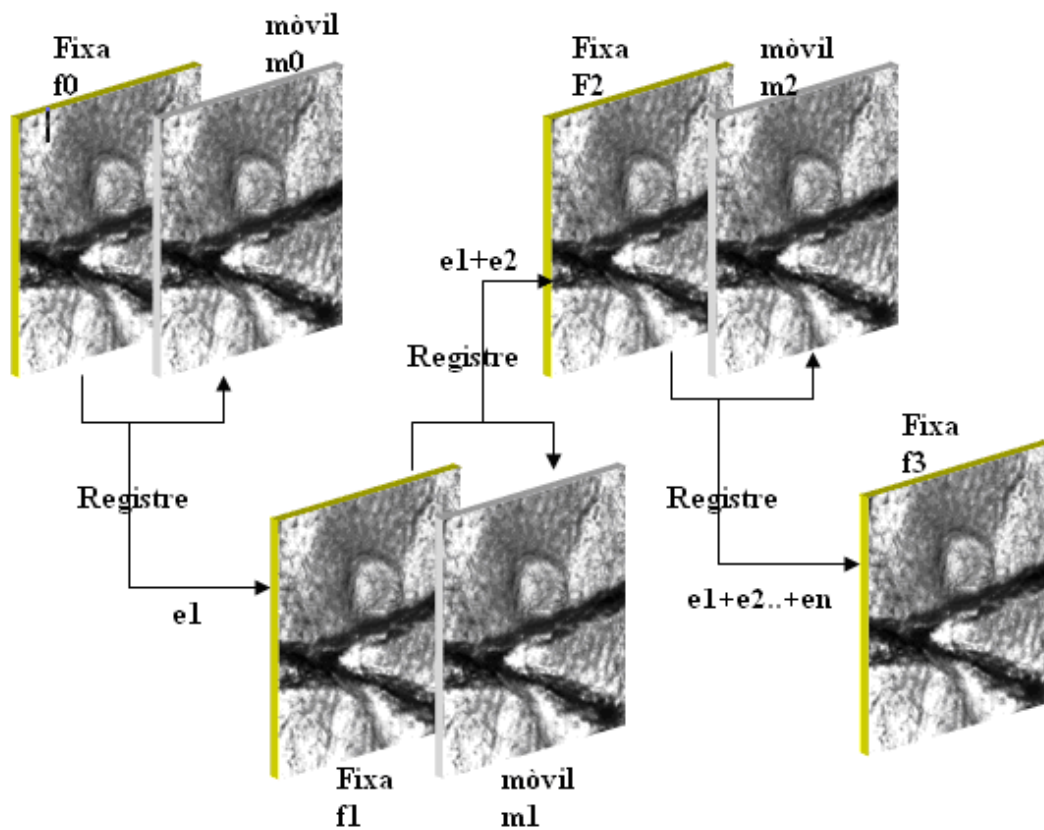


Figura 5.8. Funcionament del registre amb varies imatges de referència

Inconvenient: Acumulació d'errors

Al registrar imatges consecutives, i agafant com a imatge referència la imatge resultant, tots els errors que anem fent en cada registre (a la Figura 5.8, podem veure els errors e_1 , e_2 , .. e_n) es van acumulant fins al final.

5.3.2 Registre amb una imatge de referència (tall del mig)

Inicialment posarem com a imatge referència el tall del mig (núm. talls /2). Llavors començarem l'alineació registrant el tall referència amb el tall anterior (T_{i-1}) i posterior (T_{i+1}). Un cop tinguem aquestes imatges registrades, registrarem els talls adjacents a aquestes (T_{i-2} i T_{i+2}) amb la imatge referència que sempre serà la mateixa. Això es realitzarà fins al tall inicial i fins al tall final. En la següent figura podem veure com seria el funcionament d'aquest algorisme. (Figura 5.9).

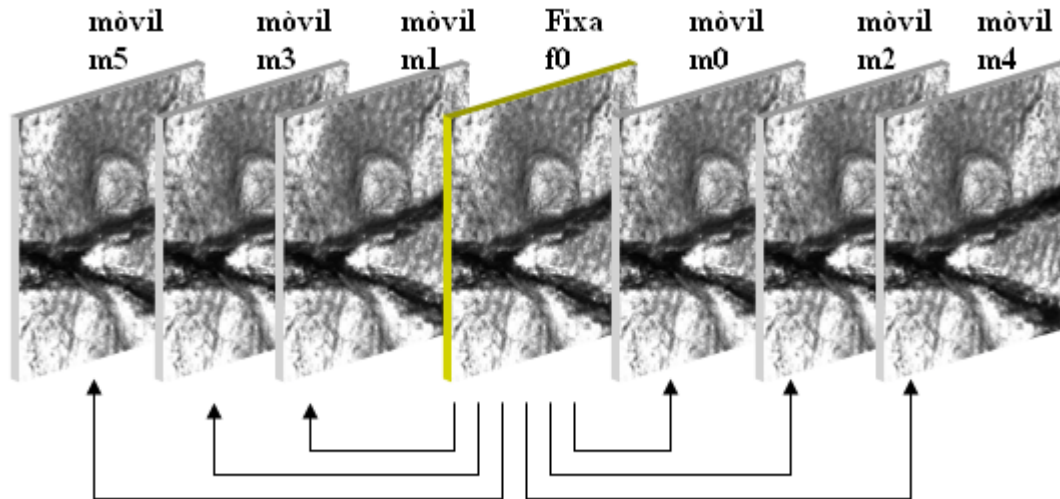


Figura 5.9. Funcionament del registre amb una imatge de referència

Inconvenient: No funciona correctament si les imatges són molt diferents

Aquest algorisme té l'avantatge en que al partir d'una imatge referència i sempre registrar amb aquesta no va acumulant errors, però en canvi si les imatges del principi són molt diferents de les finals (Fo de Mn), les imatges resultants del registre no s'adeqüen molt a la realitat.

5.3.3 Registre mixte (Agafar cada x talls una imatge de referència i aplicar en aquests talls el registre anterior)

En aquest registre s'utilitzarà un híbrid entre els dos registres anteriors. Tot seguit el registre es durà a terme de la següent manera:

Primerament, dividim el número total de talls en X per obtenir X talls de referència per fer el registre. A partir d'aquí, es registrarà des del tall inicial (T0) fins al primer tall de referència (Tr1).

El passos seran registrar el tall de referència (Tr1) amb tots els seus talls anteriors. Quan aquests ja estan alineats, llavors passem la referència al pròxim tall de referència (Tr2). Comencem a registrar tots els seus anteriors començant des de el tall que abans havíem fet servir de referència (Tr1).

Aquest procediment s'haurà d'aplicar fins arribar al últim tall de referència. Cal tenir en compte, que si l'últim tall de referència no coincideix amb el darrer tall (la divisió no és sencera), el nostre algorisme fa un registre de l'últim tall referència amb els seus posteriors fins trobar el tall final.

Escollint correctament el nombre d'imatges fixes aquest mètode minimitza els inconvenients trobats amb els altres dos mètodes.

Tenim la avantatge en que no acumulem errors constantment, ja que no agafem tots els talls com referència, i no acumulem errors a cada pas.

També tenim la millora de treballar amb no només una imatges de referència, això ens permet que si les imatges fossin molt diferents no ens donin resultats molt irreal al fer registres amb varis talls de referència que van variant.

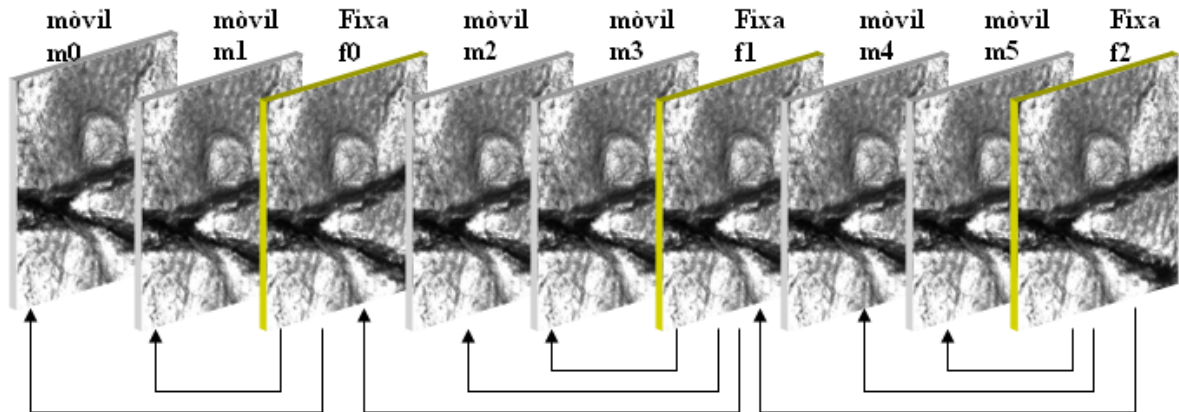


Figura 5.10. Funcionament del registre mixte

5.4 Reconstrucció 3D del isovolum.

En aquest capítol explicarem com hem construït un volum en 3 dimensions a partir de imatges 2D, juntament amb la informació recollida dels arxius de text que contenen cada tall de la pell. En aquests arxius, bàsicament el que ens interessa, és la posició relativa en l'eix Z (o sigui, la profunditat del tall en la pell) per poder reconstruir el més ajustat possible a la realitat. Partirem com a entrada, del resultat obtingut a la fase de registre 2D, que seran tots els talls confocals alineats.

5.4.1 Metodologia

El supòsit inicial és que es començarà partint des del tall inicial (considerant que es un tall perfectament alineat). Tot seguit la construcció del volum es durà a terme de la següent manera:

- A partir del tall inicial, llegim el seu arxiu de text adjunt per aconseguir la informació sobre la profunditat.
(Al trobar-nos que els talls inicials no començaven amb profunditat 0, sinó que ja tenien un valor en l'eix Z més elevat, varem considerar fer una normalització dels valors, restant-li a tots els valors la menor profunditat trobada).
- Obrir l'arxiu corresponent a la imatge, extreure la informació del valor en l'eix Z, i guardar aquest valor per utilitzar-lo en crear el volum.
- Si el tall corresponent té la mateixa profunditat que el comptador que tenim amb la posició en Z, llavors posem la imatge en aquesta posició Z en el volum 3D.
En canvi, si la profunditat del tall és superior a la que tenim al comptador, llavors el que fem és introduir un tall obtingut de la mitjana del tall anterior i posterior.
- Augmentar el comptador de la posició en Z en una unitat. La profunditat entre tall i tall serà unitària, pera aconseguir un volum uniforme.
- Repetir aquestes operacions fins trobar la màxima profunditat.

En la següent figura (Figura 5.11) mostrem la seqüència de la reconstrucció de l'isovolum, introduint a cada pas el tall registrat amb la metodologia explicada abans amb la seva corresponent profunditat recollida del fitxer de text.

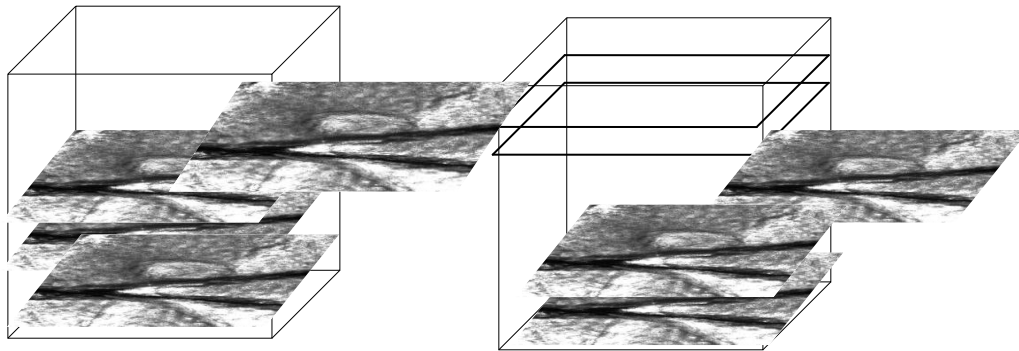


Figura 5.11. Seqüència de reconstrucció de l'isovolum introduint tall per tall registrat.

Una opció més elaborada, però que no s'ha pogut dur a terme és la interpolació dels talls intermedis en funció de la seva distància al tall més proper.

5.5 Exemples de reconstrucció

En quan a la reconstrucció, en les següents imatges (Figura 5.12) mostrem uns exemples del isovolum construït a partir dels talls de la pell aplicant els mètodes estudiats anteriorment.

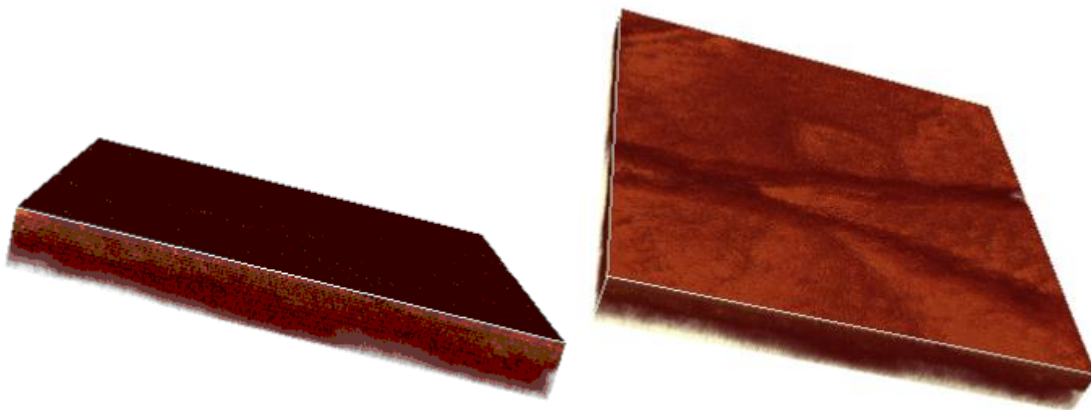


Figura 5.12. Exemples de isovolum aplicant metodologia mixta.

5.6 Sumari

En aquest capítol hem explicat com reconstruïm el isovolum a partir dels talls registrats en el capítol anterior juntament amb la seva informació extreta del fitxer de text adjunt a cada tall amb la metodologia estudiada que ens ha aportat millor resultats. En el capítol següent veurem més exemples de visualització del volum i com lligar-ho amb la eina visual per interactuar amb tota la informació.

6. Aplicació visual per l'anàlisi d'imatges Confocals

Jaume Delgado Fernandez
Enginyeria Tècnica en Informàtica de Sistemes.

Universitat de Girona, 2006

6. Aplicació visual per l'anàlisi d'imatges Confocals

6.1 Objectius

L'objectiu en aquest projecte era la construcció d'una eina visual per a ajudar als especialistes mèdics a la interpretació, interacció i visualització de les imatges MCF. Aquesta eina hem intentat que fos el més pràctica possible per als dermatòlegs i que el seu funcionament sigui fàcil i entenedor.

6.2 Requeriments

El desenvolupament de l'aplicació visual ha de complir uns certs requeriments que s'han establert conjuntament amb els metges dermatòlegs. Cal tenir en compte que aquesta és una primera versió de l'aplicació i que segurament caldrà incorporar d'altres requeriments en un futur, però que ja no formaran part d'aquest projecte. Inicialment es va establir com a prioritat la necessitat de poder visualitzar la zona d'interès reconstruïda 3D i al mateix temps permetre la visualització de talls d'aquesta regió simulant talls histològics (transversals) en els diferents eixos de coordenades. A continuació es llista els requeriments més importants de l'aplicació:

- Obrir volums reconstruïts de les imatges confocals provinents d'aplicar la metodologia descrita en els capítols anteriors.
- Permetre la visualització del volum en 3D.
- Permetre la interacció amb el volum per poder observar-lo en diferents vistes
- Permetre seleccionar plans d'aquest volum i visualitzar-lo com una imatge 2D addicional.
- Dins d'un pla seleccionat permetre la navegació en profunditat d'aquest pla.

6.3 Desenvolupament

Per desenvolupar l'eina visual hem treballat amb totes les classes aplicades en totes les tasques anteriors, com han estat les llibreries itk i vtk, juntament amb la utilització de les Qt per dissenyar la interfície de l'eina.

El nostre programa per veure els resultats del registre amb el volum tridimensional i els talls histològics de cada eix.

Hem creat una interfície amb un fàcil funcionament. Està dividit en zones diferenciades. La part de dalt és on tenim les pantalles per visualitzar. A la esquerra tenim la finestra on es veurà el volum 3D que carreguem, escollint-lo amb el botó "Carregar Volum". Al costat, la finestra de la dreta ens visualitzarà el tall histològic en l'eix que nosaltres prèviament haurem escollit (eix X, Y o Z). La Figura 6.1 mostra el disseny d'aquestes finestres.

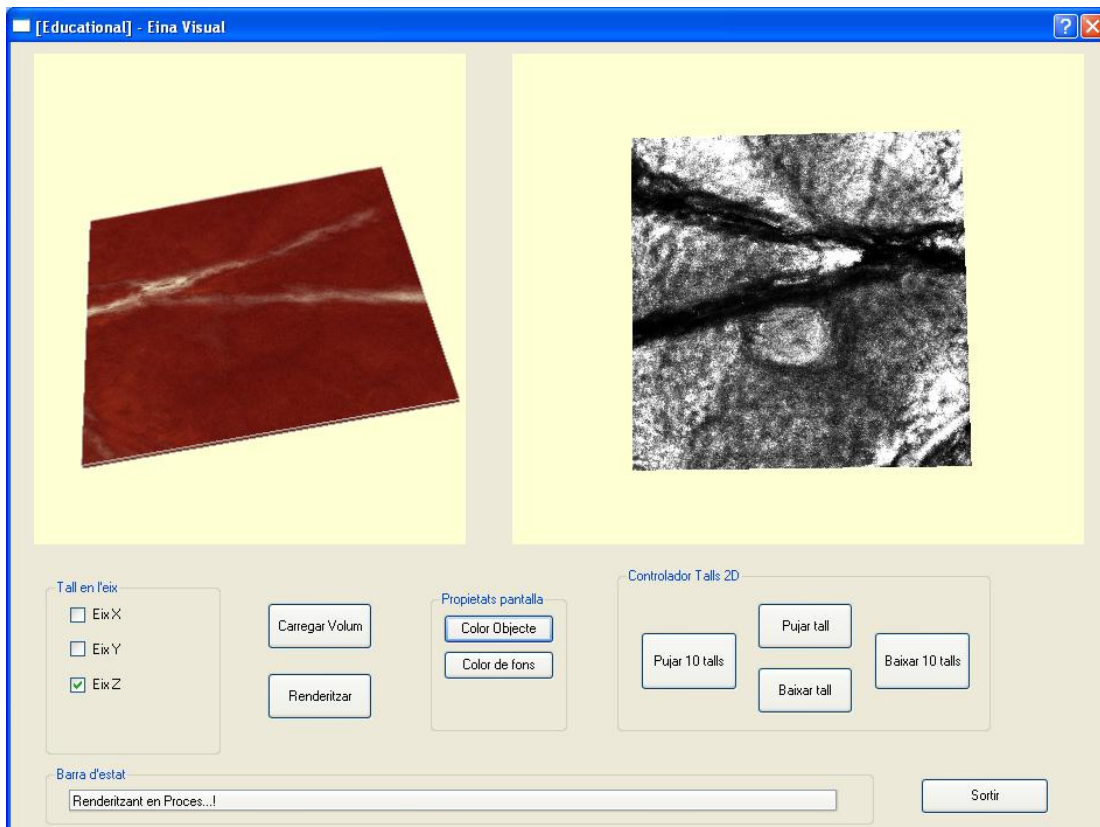


Figura 6.1. Disseny de la interfície visual

6.3.1 Obrir el volum.

Per obrir el volum registrat el que fem és agafar el volum registrat amb itk, el passem al ImageReader i després fem la connexió de itk-vtk. Llavors després creem el volum vtk i li connectem el volum 3D. En el següent diagrama de classes (Figura 6.2) podem veure com fem l'acció de obrir el volum amb les classes itk i vtk.

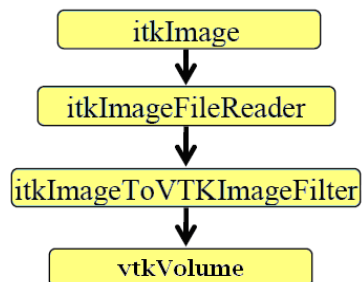


Figura 6.2. Diagrama de classes per obrir el volum.

Tal i com mostra la Figura 6.3, el diagrama funcional per obrir el volum tridimensional corresponent a la pell.

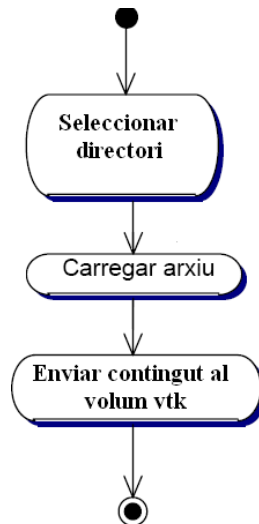


Figura 6.3. Diagrama funcional per obrir el volum.

6.3.2 Visualització i interacció amb el volum 3D

En la part de visualització i interacció s'ha utilitzat la classe `vtkVolume` per treballar amb el volum 3D, juntament amb el `vtkOutlineFilter` els passarem a la classe `vtkPolyDataMapper`.

Amb aquest objecte creat amb tota la informació del volum registrat, ja podrem posar-ho en un `vtkActor` i ja podrà renderitzar per ser visualitzat i poder interactuar amb ell.

La interacció de la interfície s'ha assolit utilitzant la classe `vtkRenderWindowInteractor`.

La Figura 6.4 ens mostra el diagrama de classes corresponent a aquesta tasca.

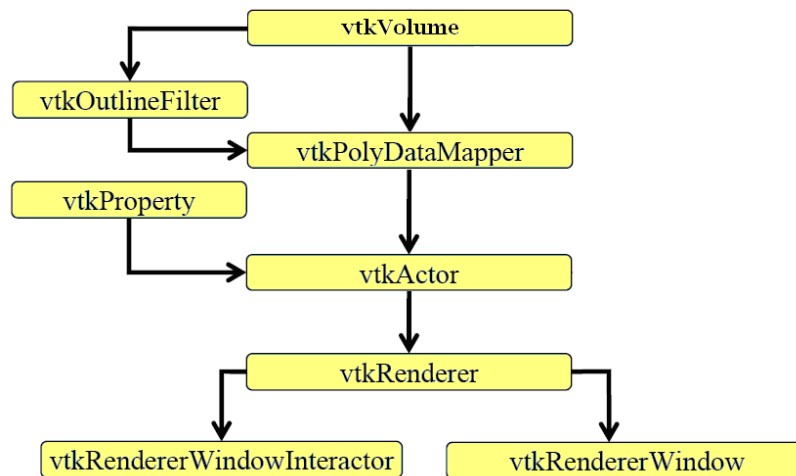


Figura 6.4. Diagrama de classes de la visualització i interacció.

La Figura 6.5 mostra el diagrama funcional de la visualització del volum.

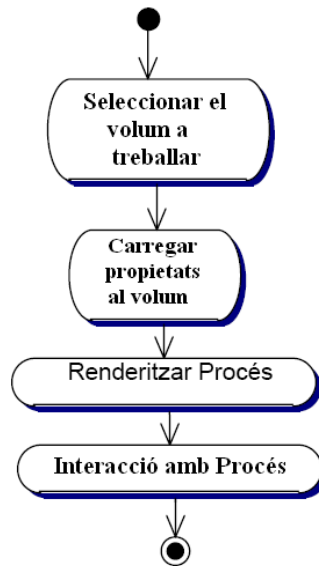


Figura 6.5. Diagrama funcional de la visualització.

6.3.3 Selecció dels plans

A partir de la classe `vtkImagePlaneWidget` que conté un pla 2D de l'eix escollit, li passarem els paràmetres necessaris amb la classe `vtkImageColorMap`. Així ja podrem crear l'actor i renderitzar el pla en el render corresponent.

En el diagrama de classes de la Figura 6.6 veiem que tenim dos `vtkRenderers`. Això és perquè tindrem un per el volum 3D i altre per visualitzar el tall 2D que haurem escollit.

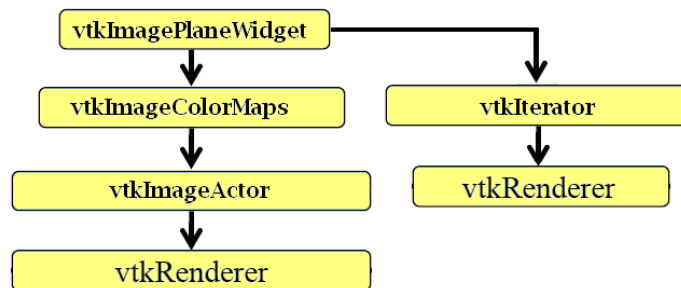


Figura 6.6. Diagrama de classes en la selecció de plans.

En la Figura 6.7 podem veure el diagrama de casos per seleccionar un pla de coordenades concret.

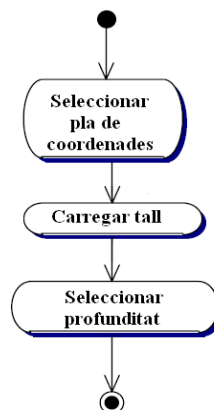


Figura 6.7. Diagrama funcional per seleccionar un pla.

La selecció dels plans serà escollit per nosaltres amb l'opció que ens dona la interfície amb "Talls en l'eix".(Veure Figura 6.8). Immediatament que activem una de les opcions, el tall es visualitzarà en la pantalla de la dreta.

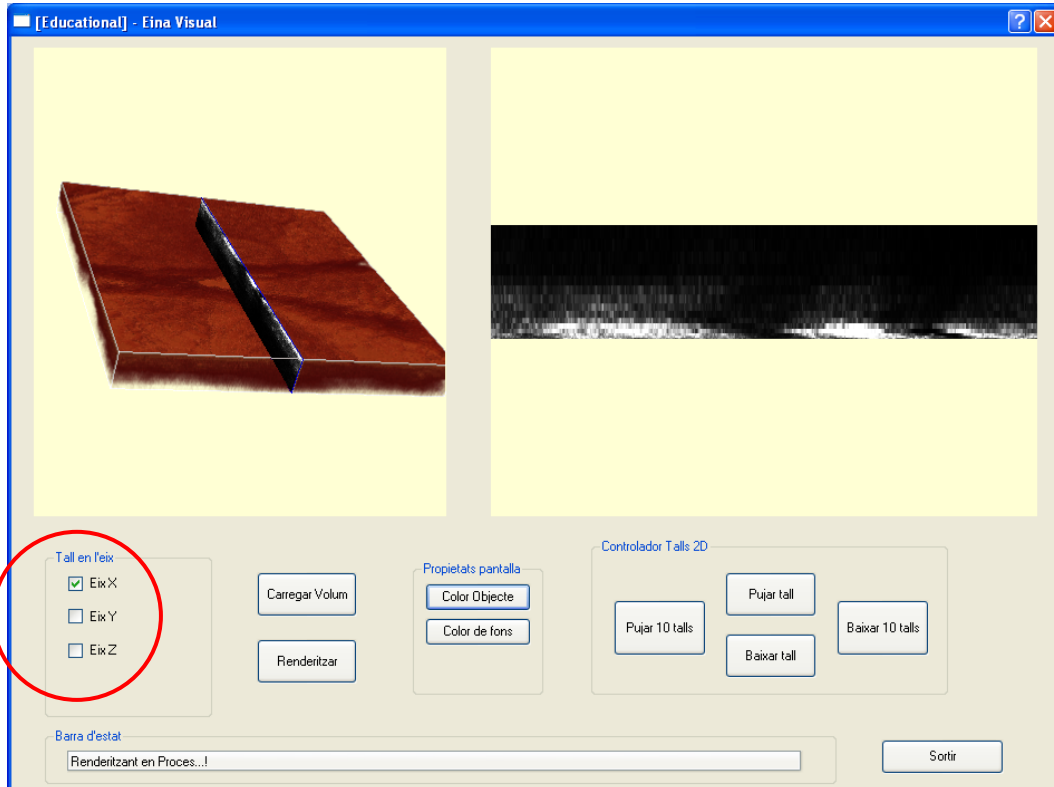


Figura 6.8. Exemple de com es veu el pla seleccionat en el volum 3D.

6.3.4 Navegació dins del pla

Un cop escollit un tall histològic i ja visualitzat, un dels requeriments que teníem i hem implementat és poder escollir la profunditat del tall per poder veure una regió d'interès exacta a una profunditat puntual. Amb el ratolí podem interactuar amb l'eina poden moure el volum 3D i els talls de la manera que vulguem. A més podrem fer Zoomin i Zoomout amb el botó dret del ratolí, per observar millor la regió d'interès que ens interessa. A continuació veiem els botons per realitzar aquests requeriments (Figura 6.9).

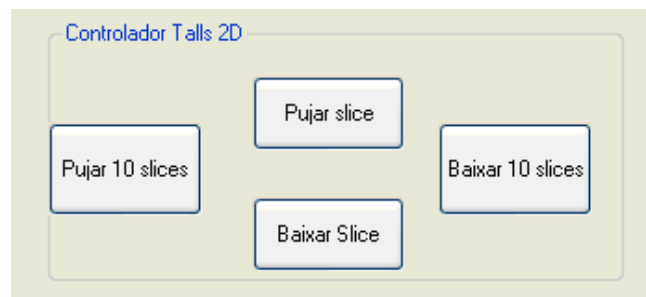


Figura 6.9. Botons preparats per treballar amb la profunditat dels talls.

També hem de tenir en compte que l'eina creada també permet funcionar amb les ordres que li podem enviar des de el teclat.

Llavors prement les tecles x, y o z veuríem els talls histològics en aquest eix de coordenades. I amb les tecles de pujar, baixar, RePAg i AvPag podrem escollir la profunditat d'aquest tall.

A continuació, en la Figura 6.10 veiem com seria el diagrama de classes per poder escollir el tall de pell amb la profunditat que vulguem.

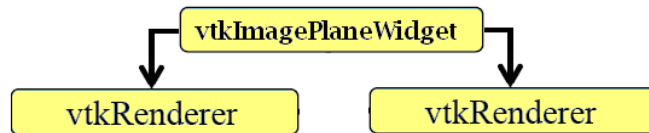


Figura 6.10. Diagrama de classes per escollir profunditat d'un tall.

6.3.5 Altres funcionalitats

Paral·lelament a tots els requeriments de l'eina, hem afegit unes opcions secundàries que poden millorar la visualització. Aquestes són l'opció de canviar el color de fons de les finestres on es veuran els volums i talls i l'altre serà poder canviar el color del contorn del volum 3D. Aquestes millores ens permetran veure més nítidament el contorn del volum si el volum i el fons són de color massa semblants, igualment amb el canvi de color dels fons.

A la Figura 6.11 podem observar com podem utilitzar aquestes opcions.



Figura 6.11. Comandaments per canviar les propietats de la pantalla.

6.4 Exemples

En els següents exemples veurem com hem dissenyat la interfície i resultats d'unes proves realitzades amb l'eina creada.

En la Figura 6.12 podem observar un exemple de l'eina en funcionament, quan s'ha carregat justament el volum 3D registrat.

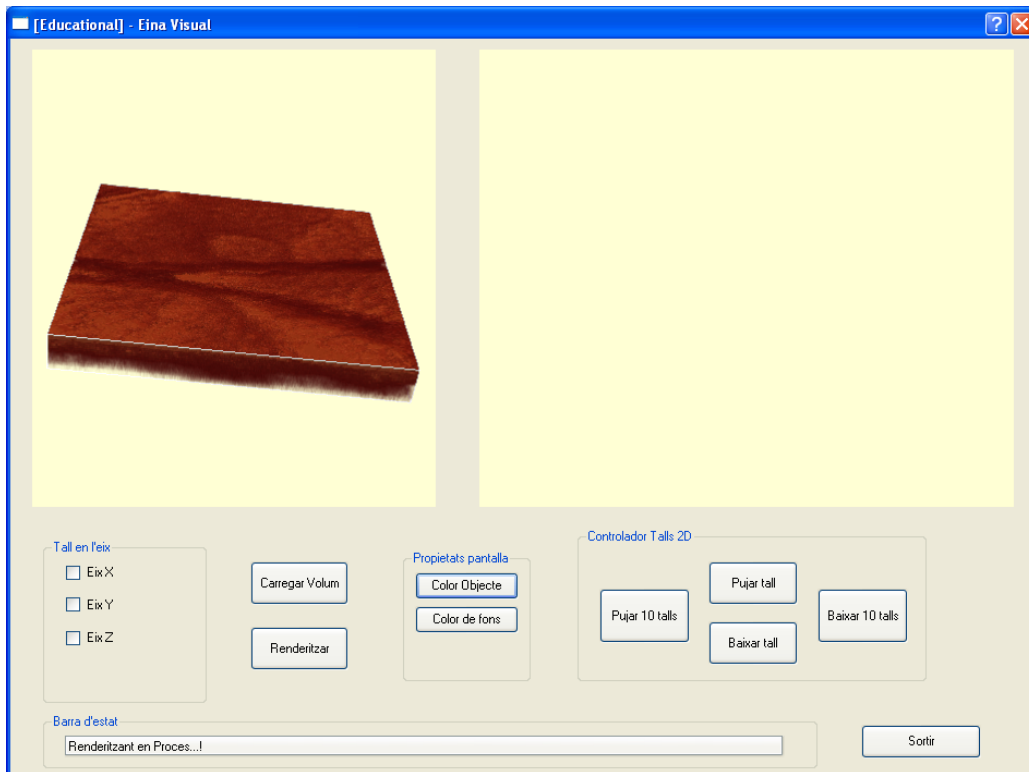


Figura 6.12. Volum carregat i renderitzat.

A continuació veiem la interfície amb el tall Z visualitzat en la finestra dreta, amb el volum corresponent a la part esquerra.(Figura 6.13)

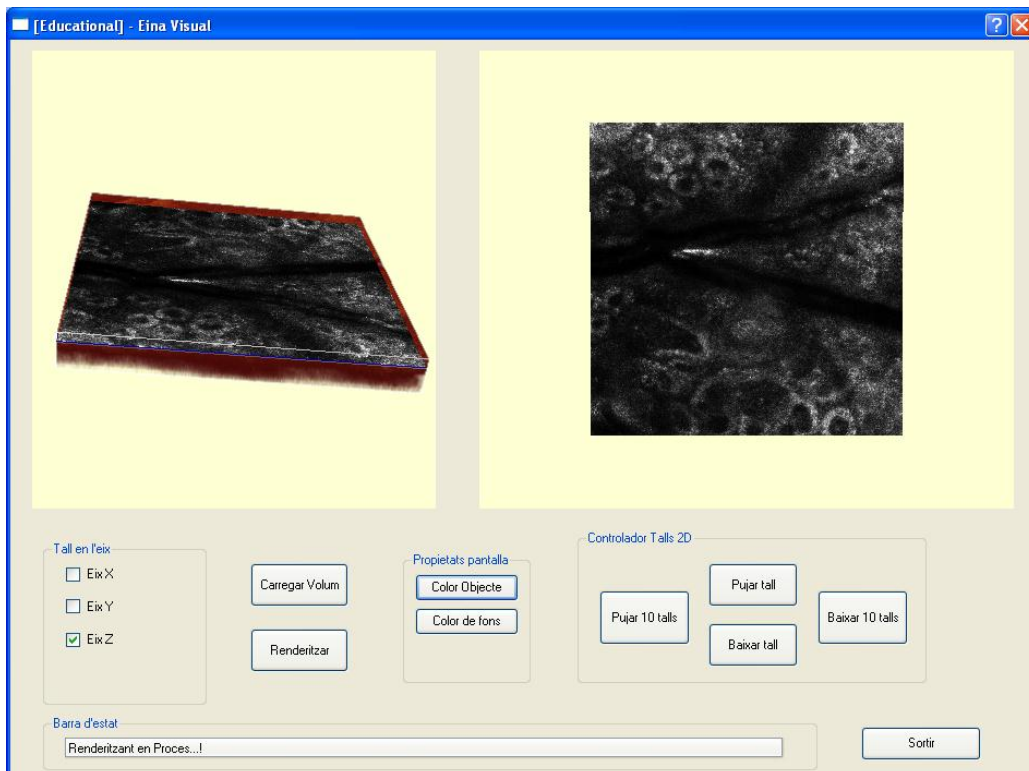


Figura 6.13. Visualització del tall X amb una profunditat escollida per l'usuari.

Com mostra la Figura 6.14, veiem la interfície amb el tall X visualitzat en la finestra dreta amb la profunditat escollida al centre, amb el volum corresponent a la part esquerra.

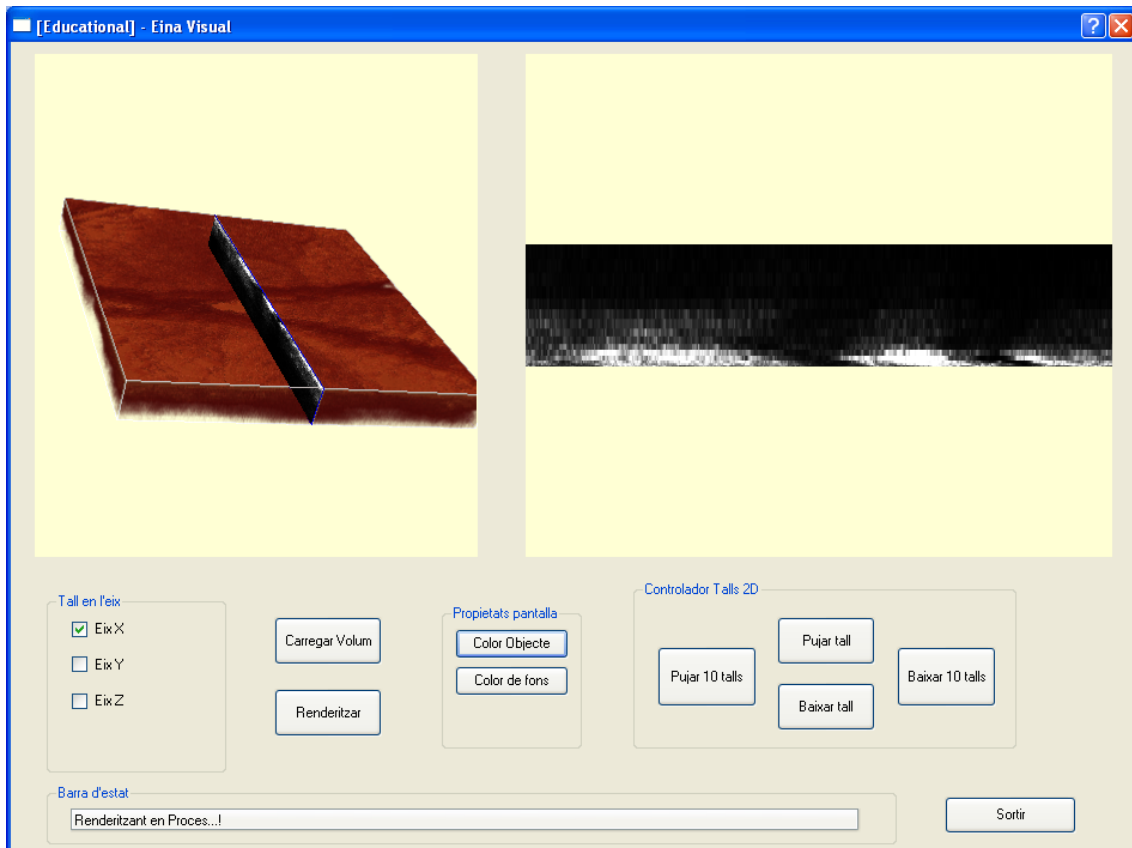


Figura 6.14. Exemple de visualització del tall X amb profunditat a la meitat del volum.

6.5 Sumari

En aquest capítol hem construït l'eina visual pensada per ajudar als especialistes dermatòlegs per interpretar les imatges confocals de la pell.

Amb aquesta tasca, finalment, assolim tots els objectius del projecte. En els pròxims capítols veurem les conclusions trobades juntament amb possibles millores futures del projecte i la seva bibliografia.

7. Conclusions i treballs futurs

Jaume Delgado Fernandez

Enginyeria Tècnica en Informàtica de Sistemes.

Universitat de Girona, 2006

7. Conclusions i treballs futurs

Havent realitzat un conjunt de proves per testar les diferents tasques que ens varem fixar, es poden extreure un seguit de conclusions. Tanmateix, l'experiència adquirida en la realització del projecte ens permet proposar millores i camins que es poden prendre en futurs projectes, que seguidament s'exposen.

7.1 Conclusions

Els objectius fixats al començament del projecte s'han acomplerts satisfactòriament. S'ha pogut avaluar l'anàlisi realitzat i la solució adoptada. D'altra banda s'ha proposat un mètode d'avaluació per avaluar el registre 2D, tant de manera subjectiva (visualment mitjançant diferències d'imatge i checkboards) com objectivament emprant les mètriques utilitzades al procés de registre (Informació mútua i diferència de quadrats).

Els resultats obtinguts amb les imatges confocals han estat molt bons alhora d'intentar el registre 2D, tenint en compte que les imatges obtingudes amb MCF són difícils d'interpretar. Cal remarcar que això un treball de recerca i amb altres tipus de mètriques i registres més avançades els resultats haurien pogut millorar lleugerament però l'objectiu principal en el qual s'ha enfocat el projecte s'ha assolit correctament.

Després d'haver fet totes les proves necessàries i programades en quina opció de registre era la més adequada pel nostre registre d'imatges de la pell, i per deduir si era millor registrar o no registrar per reconstruir el volum 3D de la pell, varem arribar a la conclusió que l'opció més eficient pel nostre PFC era utilitzant el registre RMSE (Registre Rígid amb mètrica de mitjanes quadrades).

La reconstrucció del volum en 3 dimensions a partir dels talls d'imatge registrats s'ha assolit satisfactòriament. El nostre sistema d'anar emplenant el volum entre profunditats que no tenen tall amb talls resultats de fer mitjanes entre talls anterior i posterior ens dona un resultat molt real del volum.

L'eina de visualització ens va aportar una sèrie de problemes que no ens esperàvem. El traspàs dels mètodes sobre la visualització dels talls histològics de ITK i VTK sobre Qt no va ser fàcil degut a incompatibilitats entre les diferents eines i a la seva dificultat d'aprenentatge. Finalment, es va poder enllestir a temps.

Les proves realitzades amb les imatges confocals han evidenciat la dificultat d'aquest tipus de dades. El registre del volum s'ha assolit correctament. Tot i així, la dificultat que comporta que els talls inicials siguin tant diferents amb els més profunds han donat lloc a un registre més complicat en aquestes regions.

7.2 Treballs Futurs

En aquest tema de treballs futurs, podríem entrar en molts de detalls en possibles millores, ja que al ser un treball final de carrera que comença una nova línia d'investigació mèdica amb imatges MCF de la pell, ens hem trobat moltes noves idees que anaven sortint en el dia a dia. Les més interessants les descriurem seguidament.

Aquest registre de les imatges de la pell en ha suposat un procés lent i costós per el nostre sistema d'eines que tenim. Interessaria, com una feina futura, aconseguir un registre 2D molt més ràpid i en temps mínim, tot això tenint en compte les nostres possibilitats. Una opció, pensem que podria ser treballant amb el hardware, que podria accelerar els càlculs tan costosos.

La mètrica és un factor molt important al procés de registre, per tant, és inevitable remarcar que un dels treballs futurs anirà dirigit a estudiar-ne altres mètriques i possiblement a proposar-ne de noves que tinguin en compte la informació addicional de que disposem.

La metodologia emprada per l'apartat de la reconstrucció del volum tridimensional, varem escollir un entre tres estudis que varem realitzar, degut als millors resultats obtinguts. Una proposta de millora seria estudiar altres metodologies i comparar-les amb les proposades en aquest treball. Això també es podria aplicar a l'algorisme d'emplenar el isovolum amb els talls registrats.

La necessitat de realitzar proves amb una base de dades més extensa és evident. Tot i que ha mostrat bons resultats, la incorporació d'una gran variabilitat podria fer investigar amb un camp més extens de proves i així, possiblement, obtenir millors resultats.

La interfície desenvolupada ha estat dissenyada tenint en compte les necessitats inicials dels dermatòlegs. Tanmateix una possible feina futura seria millorar la interfície visual, juntament amb la opinió dels dermatòlegs per possibilitar la seva utilització en la pràctica clínica habitual.

8. Bibliografia

Jaume Delgado Fernandez
Enginyeria Tècnica en Informàtica de Sistemes.

Universitat de Girona, 2006

8. Bibliografia

[Bauer, 2005], "Processing and 3D visualization of biological volumes of confocal microscopy", Grupo de Visió Artificial (GVA), Escuela Universitaria de Ingenier? Tècnica Industrial de Madrid, www.elai.upm.es

[NCI, 2005] "National Cancer Institute" Website.
<http://www.cancer.gov.html> accedit el 20 de Febrer de 2006.

[Mundo Salud, 2006] Especialistes dermatòlegs i oncòlegs.
<http://www.elmundo.es/elmundosalud/especiales/cancer/piel.html> accedit el 2 d'abril de 2006.

[Cohoon, 2000] J. P. Cohoon, J. W. Davidson; "Programación y diseño en C++";
Mc Graw Hill 2ª Edició, 2000

[Schroeder, 2005] W. Schroeder, K. Martin, B. Lorensen "VTK The
Visualization Toolkit",. Kitware. web: <http://www.vtk.org>

[Ibáñez, 2005] ITK Insight Toolkit, Kitware. Luis Ibáñez, Will Schroeder, Lydia
Ng, Josh Cates, 2nd Edition, 2005. web: <http://www.itk.org>

[Insight into Image,2004] Terry S. Yoo "Insight into Image" Principles and practice for
segmentation, registration and image analysis, 2004.

[NLM 2005] "National Library of Medicine and National Institutes of Health".
<http://www.nlm.nih.gov.html> accdeit el 14 de Març de 2006.

[PFC Torramilans, 2005] Marc Torramilans Privat.Sistema informàtic de gestió,
processament i anàlisi d'imatges de ressonància magnètica de la pròstata per l'ajuda
en el diagnosi del càncer. PFC Enginyeria Tècnica en Informàtica de Gestió. Juny
2005.

[PFC Acuña, 2006] Ivan Acuña. Registre intra i intermodal d'imatge prostàtica: Imatge
de ressonància magnètica i ultrasó de la pròstata.
PFC Enginyeria Tècnica en Informàtica de Gestió. Juliol 2006.

[Visual2005] Visual C++ 2005:
<http://msdn.microsoft.com/vstudio/express/visualc/>

[CMake] Generació de projectes genèrics multiplataforma:
<http://www.cmake.org>

[Paraview] Programa de visualització: <http://www.paraview.org>

[LlistesDistribució] Llistes de distribució ITK i VTK:
<http://public.kitware.com/pipermail/insight-users/>
<http://www.vtk.org/mailman/listinfo/vtkusers>

A. Annex. Disseny i implementació

Jaume Delgado Fernandez
Enginyeria Tècnica en Informàtica de Sistemes.

Universitat de Girona, 2006

9. Annex. Disseny i implementació

En aquest apartat s'explicaran les classes ITK i VTK utilitzades pel desenvolupar els algorismes necessaris per poder desenvolupar les tasques definides durant el projecte.

9.1 Disseny

Les classes que s'utilitzaran per desenvolupar el registre d'imatges són les que podem veure a la Figura 9.1. La zona grisa del diagrama serà la part que s'haurà d'iterar des del tall inicial fins al tall final per tal de poder obtenir cada tall correctament alineat.

La classe `itk::ImageSeriesFileNames` ens permetrà llegir des del primer tall fins al tall final. Després li aplicarem el mètode escollit per registrar amb totes les seves característiques (mètrica, transformador...) i li aplicarem uns filtres com són `itkResampleImageFilter`, `itkCastImageFilter` i `itkRescaleIntensityImageFilter`. A `itkImageFileWriter` posarem la nova imatge registrada. [Ibáñez, 2005]

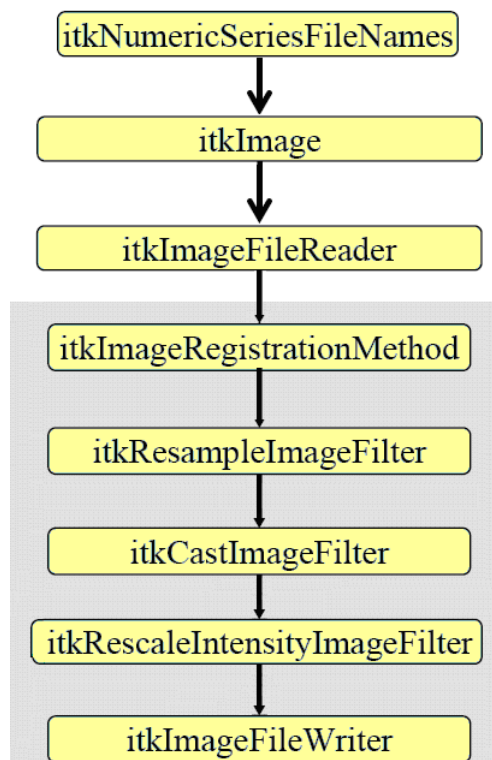


Figura 9.1. Diagrama de classes utilitzades a l'algorisme de registre.

9.1.1 Registrar els talls

Els paràmetres d'entrada de l'algoritme correspondran al tipus d'imatge (png) dels talls, del tall inicial i al tall final. També hi ha un valor opcional que permetrà especificar els talls referència del qual es començarà a registrar.

Tal com s'ha explicat anteriorment, l'algoritme ha estat dissenyat amb el mètode mixte (com varem parlar al capítol 5 i mostrem a la Figura 9.2). La primera part registrarà a partir del tall inicial fins al primer tall de referència. Un cop s'hagi registrat la primera part dels talls, es registrarà des del primer tall referència fins al segon tall de referència. Així fins arribar a l'última imatge.

Un cop tinguem tots els talls alineats, es podrà reconstruir el volum en tres dimensions amb les imatges alineades.

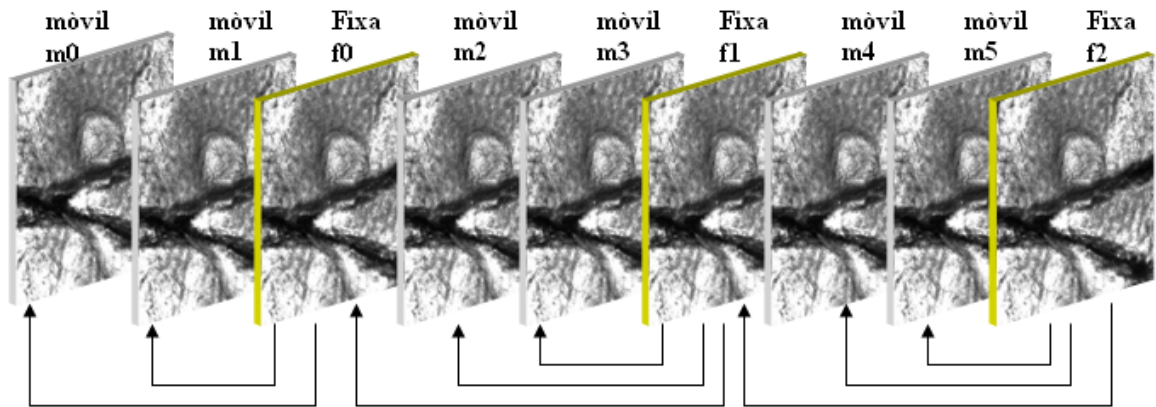


Figura 9.2. Esquema del procés seguit durant la tasca de registre.

Les classes més importants utilitzades de les llibreries ITK són les que ens mostra les següents figures. S'utilitza la classe que fa les transformacions Afins, transformacions rígides 2D, mètrica de mitjanes quadrades i mètrica de informació Mútua, entre d'altres classes.

Registre AMI

A continuació hi han els diagrames de les classes ITK utilitzades pels diferents algorismes implementats per el registre d'imatges 2D. La Figura 9.3 correspon al registre AMI utilitzant com a classes principals:

mètrica → itkMattesMutualInformationImageToImageMetric

interpolador → itkLinearInterpolateImageFunction

optimitzador → itkRegularStepGradientDescent

transformador → itkAffineTransform i itkCenteredTransformInitializer

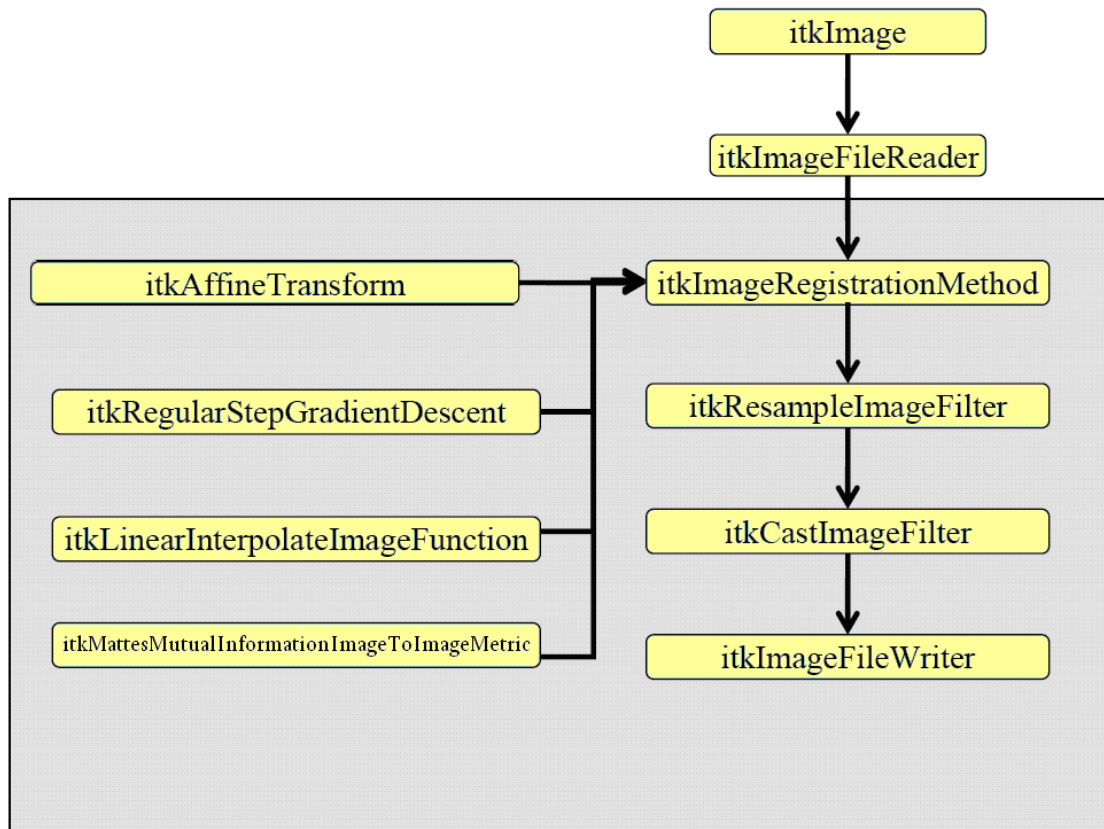


Figura 9.3. Diagrama de classes utilitzades en el registre AMI.

Registre AMSE

La Figura 9.4 correspon al registre AMSE utilitzant com a classes principals:
mètrica → itkMeanSquaresImageToMetric
interpolador → itkLinearInterpolateImageFunction
optimitzador → itkRegularStepGradientDescent
transformador → itkAffineTransform i itkCenteredTransformInitializer

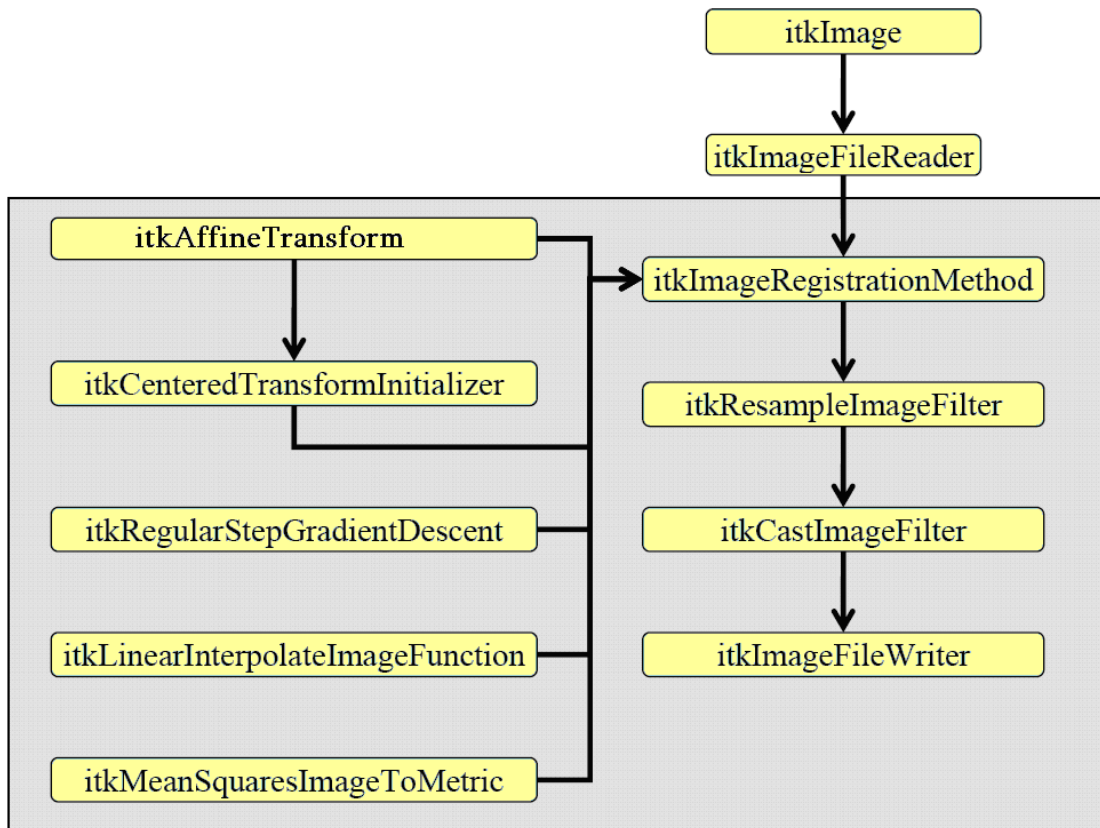


Figura 9.4. Diagrama de classes utilitzades en el registre AMI.

Registre RMI

La Figura 9.5 correspon al registre RMI utilitzant com a classes principals:

mètrica → itkMattesMutualInformationImageToImageMetric

interpolador → itkLinearInterpolateImageFunction

optimitzador → itkRegularStepGradientDescent

transformador → itkCenteredRigid2dTransform i itkCenteredTransformInitializer

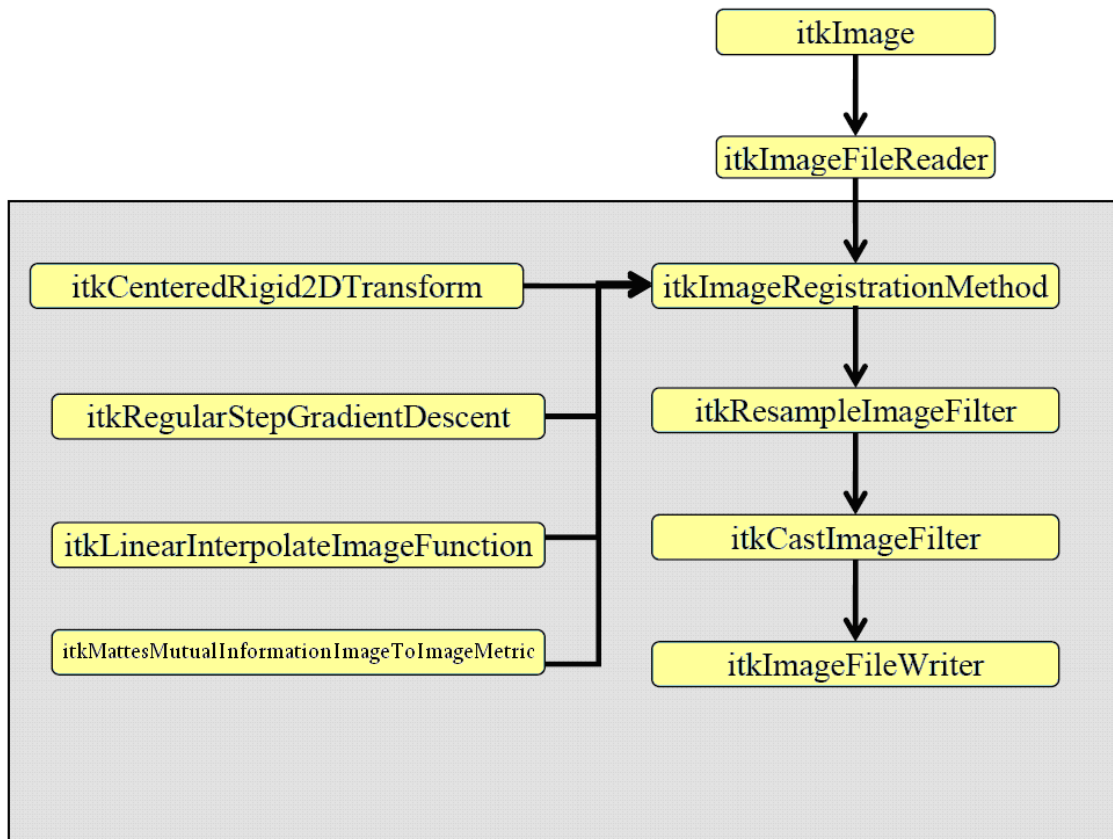


Figura 9.5. Diagrama de classes utilitzant el registre amb transformador Rígid i Informació Mútua.

Registre RMSE

La Figura 9.6 correspon al registre utilitzant com a classes principals:

mètrica → itkMeanSquaresImageToMetric

interpolador → itkLinearInterpolateImageFunction

optimitzador → itkRegularStepGradientDescent

transformador → itkCenteredRigid2DTransform i itkCenteredTransformInitializer

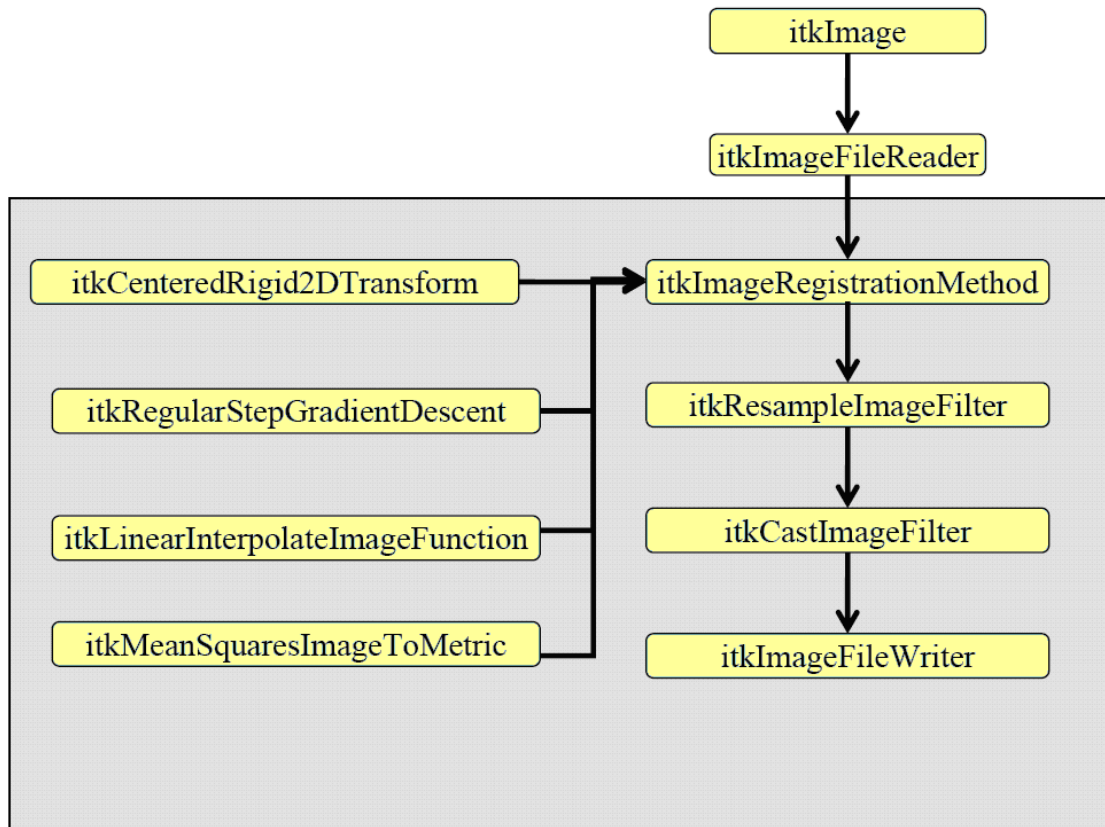


Figura 9.6. Diagrama de classes utilitzant el registre RMSE.

Un cop fet el registre, per avaluar els resultats obtinguts fem servir les classes `itkAbsoluteValueDifferenceImageFilter` i `itkCheckerBoardImageFilter`.

Aquestes classes ens donaran informació visual sobre l'alineament. Una farà la diferència absoluta entre la imatge fixa i la mòvil (veure Figura 9.7). Si les dues imatges són idèntiques la imatge diferència serà en blanc. En el cas contrari ressaltarà les zones que no coincideixen.

La classe `itkCheckerBoardImageFilter` ens permet obtenir noves imatges dividides en regions on cada regió és una part d'una de les dos imatges. D'aquesta forma es pot veure l'unió de les dues imatges i per tant veure si s'han alineat correctament, com mostra el diagrama de classes de la Figura 9.8.

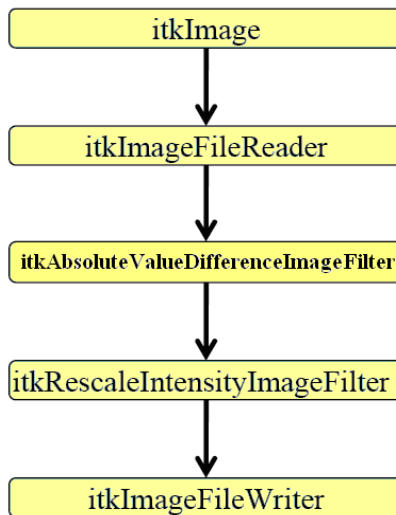


Figura 9.7. Diagrama de classes per calcular la diferència entre imatges.

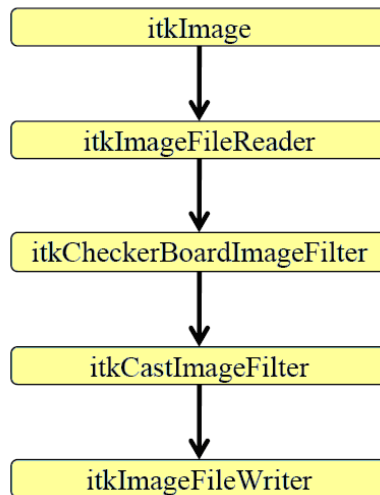


Figura 9.8. Diagrama de classes per calcular el checkerboard entre dues imatges

9.1.2 Reconstrucció del volum 3D.

Les classes que s'utilitzaran per desenvolupar la tasca de reconstrucció del isovolum són les que podem veure ens els diagrames de la classe ITK següents.

9.1.2.1 Adquisició de les dades.

Les dades per començar a reconstruir el volum les extraiem del resultat de la tasca anterior com era el registre d'imatges 2D. Rebrem un volum amb les imatges registrades en format vtk.

Llavors nosaltres llegirem aquest volum registrat i amb una classe anomenada itkExtractImageFilter anirem extraient tall per tall desde l'inicial fins al final. En aquest moment, serà quan ja tindrem tots els talls registrats preparats per reconstruir el volum amb les profunditats corresponents.

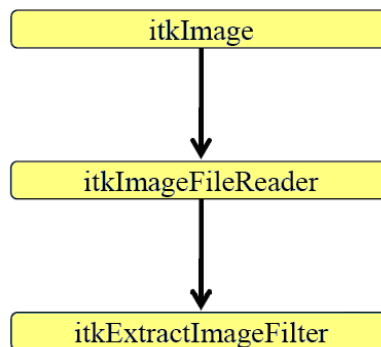


Figura 9.9 Diagrama de classes per l'adquisició de les dades

9.1.2.2 Lectura del fitxer de text de cada tall

A partir del fitxer de text que està adjuntat a cada tall de pell, hem d'obtenir les característiques necessàries per reconstruir el volum. Aquestes dades seràn les posicions relatives en els eixos X, Y i Z.

Aquest procediment el podríem haver realitzat amb les llibreries que ens proporciona Windows per el tractament de fitxers, però per tenir una aplicació multiplataforma aplicable a qualsevol entorn, vam aplicar mètodes de itk.

9.1.2.3 Construcció isovolum

Tenint els talls 2D registrats i el valor de la seva profunditat , passem a reconstruir el volum amb el mètode que hem creat per introduir cada tall al seu lloc corresponent.

Per la reconstrucció utilitzem el mètode SetSlice, que s'utilitza per introduir cada tall a la seva corresponent profunditat. En la Figura 9.10 veiem com el diagrama de classes ens mostra com s'agafa el tall 2D i juntament amb el Punter ho posen al Iterador2D. Mateix sistema per el 3 dimensions.



Figura 9.10. Diagrama de classes en la construcció del volum

9.1.3 Visualització

Per la visualització del isovolum i dels talls 2D hem treballat amb les llibreries vtk. La Figura 9.11 ens mostra les classes utilitzades per poder visualitzar el volum que llegim i poguem visualitzar els talls que el conformen. La zona grisa, igual que la figura anterior, és on haurem d'iterar per poder visualitzar qualsevol dels talls del volum. Per tal de poder renderitzar l'escena i poder visualitzar els objectes que ens facin falta s'han d'utilitzar uns elements bàsics VTK que són els següents: [Schroeder, 2005]

- **Actor:** representa allò que veurem a l'escena. Els actors estan definits per objectes de tipus mapper, property i transform.
- **Light :** il·lumina els actors de l'escena.
- **Camera:** determina la posició de l'observador, control de geometria tridimensional.
- **Property:** representa els atributs del rendering de l'actor (color, textura, espectacularitat, shading,...)
- **Mapper:** representa la definició geomètrica de l'actor
- **Transform:** especifica la posició i orientació d'un actor, càmera o light
- **Renderer:** coordina el rendering dels llums, càmeres i objectes de l'escena
- **RenderWindow:** maneja les finestres en el procés de visualització. Es poden visualitzar més d'un renderer en una escena

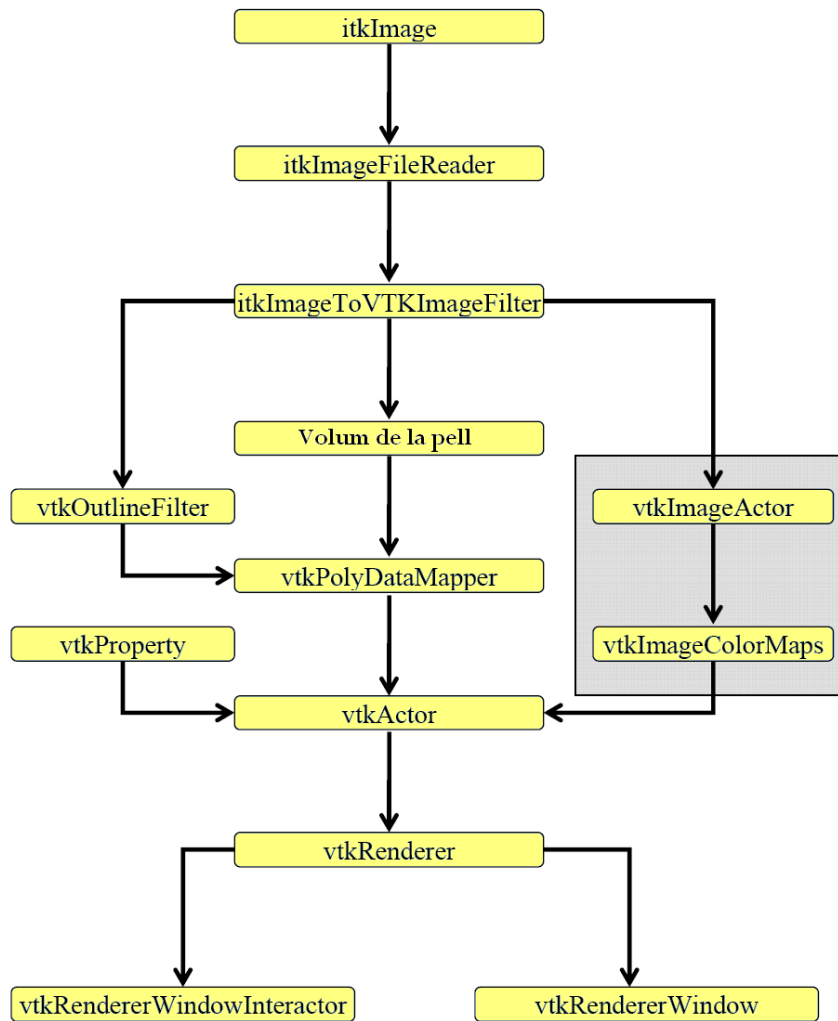


Figura 9.11. Diagrama de classes vtk per la visualització.

9.2 Implementació

En aquest apartat es mostren detalls referents a la implementació realitzada. S'ha inclòs talls de codi rellevants per l'explicació del procés. Això permetrà obtenir una visió més detallada de la implementació així com de la utilització conjunta de les llibreries ITK i VTK.

9.2.1 Registre 2D de les imatges mèdiques.

Primer, declarem les principals llibreries itk que utilitzarem en el registre 2D del nostre codi. [Visual2005]

```

#include "itkImage.h"
#include "itkImageRegistrationMethod.h"
#include "itkCenteredRigid2DTransform.h"
#include "itkMeanSquaresImageToImageMetric.h"
#include "itkLinearInterpolateImageFunction.h"
#include "itkRegularStepGradientDescentOptimizer.h"
  
```

El nostre registre, després d'avaluar totes les proves fetes en el capítol 4, vam arribar a la conclusió que la millor manera de registrar era amb un registre RMSE. Aquest registre estaria format per una mètrica de mitjanes quadrades(MSE) i un transformador Rígid(R).

Sempre un registre està format per quatre elements bàsics imprescindibles per alinear imatges. Podem mirar els elements en la Figura 9.12.

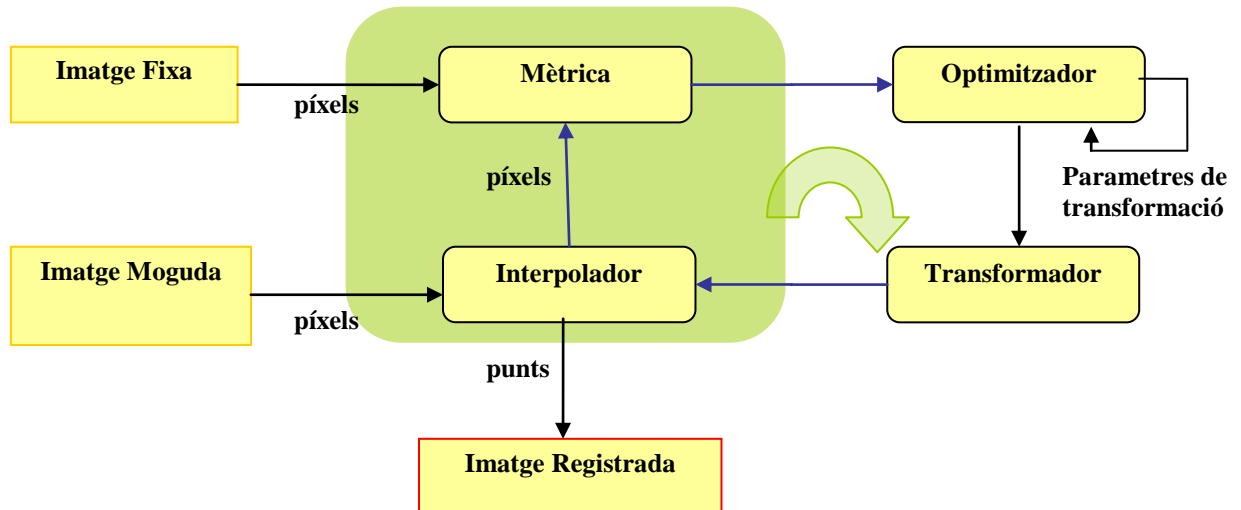


Figura 9.12. Components bàsics d'un registre. Dues imatges d'entrada, el transformador, la mètrica, l'interpolador i l'optimitzador

Aquests elements són la mètrica que ho implementarem amb el mètode `itk::MeanSquaresImageToImageMetric`, el transformador amb `itk::CenteredRigid2DTransform` només utilitzat amb imatges 2D.

L'optimitzador utilitzarem el Gradient descendent que és `itk::RegularStepGradientDescentOptimizer` i l'interpolador estarà implementat per la classe `itk::LinearInterpolateImageFunction`.

```

typedef itk:: itkCenteredRigid2DTransform < double > TransformType;
typedef itk::RegularStepGradientDescentOptimizer OptimizerType;
typedef itk::LinearInterpolateImageFunction< ImageType, double >
InterpolatorType;
typedef itk::MeanSquaresImageToImageMetric< ImageType, ImageType > MetricType;
typedef itk::ImageRegistrationMethod< ImageType, ImageType > RegistrationType;
  
```

Ara ja podem crear els objectes corresponents i després ja ho podem passar tot al mètode del registre.

```

MetricType::Pointer metric = MetricType::New();
OptimizerType::Pointer optimizer = OptimizerType::New();
InterpolatorType::Pointer interpolator = InterpolatorType::New();
RegistrationType::Pointer registration = RegistrationType::New();

registration->SetMetric( metric );
registration->SetOptimizer( optimizer );
registration->SetInterpolator( interpolator );
  
```

També creem l'objecte Inicialitzador del transformador, que prepara tots els paràmetres necessaris per començar el mètode registre.

```

TransformInitializerType::Pointer initializer =
TransformInitializerType::New();
initializer->SetTransform( transform );
initializer->SetFixedImage( fixedImage );
initializer->SetMovingImage( movingImage );
initializer->MomentsOn();
initializer->InitializeTransform();

registration->SetInitialTransformParameters( transform->GetParameters() );

```

En l'optimitzador també s'inicialitzen tots els seus valors per què després faci la seva feina d'optimitzar correctament.

```

OptimizerScalesType optimizerScales( transform->GetNumberOfParameters() );
double translationScale = 1.0 / 1000.0;
optimizerScales[0] = 1.0;
optimizerScales[1] = 1.0;
optimizerScales[2] = 1.0;
optimizerScales[3] = translationScale;
optimizerScales[4] = translationScale;
optimizerScales[5] = translationScale;

optimizer->SetScales( optimizerScales )
optimizer->SetMaximumStepLength( steplength );
optimizer->SetMinimumStepLength( 0.0001 );
optimizer->SetNumberOfIterations( maxNumberOfIterations );

optimizer->MinimizeOn();

```

Instanciem un ResampleFilter

```

ResampleFilterType::Pointer resampler = ResampleFilterType::New();

resampler->SetTransform( finalTransform );
resampler->SetInput( movingImage );

ImageType::Pointer fixedImage2 = fixedImage;
ImageType::Pointer movingImage2 = movingImage;

resampler->SetSize( fixedImage2->GetLargestPossibleRegion().GetSize() );
resampler->SetOutputOrigin( fixedImage2->GetOrigin() );
resampler->SetOutputSpacing( fixedImage2->GetSpacing() );
resampler->SetDefaultPixelValue( 100 );

```

Convertim la imatge registrada amb un caster per enviarla com a Pointer de sortida.

```

CastFilterType::Pointer caster = CastFilterType::New();
caster->SetInput( resampler->GetOutput() );

OutputImageType::Pointer slice2d = OutputImageType::New();
slice2d = caster->GetOutput();
caster->Update();

```

9.2.2 Reconstrucció volum tridimensional

En l'apartat de reconstrucció del volum tenint en compte les profunditats de cada tall, les principals llibreries que utilitzarem seran:

```
#include "itkImage.h"
#include "itkImageRegistrationMethod.h"
#include "itkMeanSquaresImageToImageMetric.h"
#include "itkLinearInterpolateImageFunction.h"
#include "itkRegularStepGradientDescentOptimizer.h"
#include "itkCenteredTransformInitializer.h"
#include "itkCenteredRigid2DTransform.h"
#include "itkNumericSeriesFileNames.h"
#include "itkExtractImageFilter.h"
```

Llegim tots els arxius de text que conté cada imatge per obtenir la informació referent a la profunditat del tall en l'eix Z. Obrirem i llegirem tots els arxius i ens quedarem amb el valor de profunditat que el guardarem en una taula on cada tall té la seva posició en Z. Aquest pas el farem amb aquesta metodologia emprant les eines itk:

```
itk::NumericSeriesFileNames::Pointer fit = itk::NumericSeriesFileNames::New();
fit->SetStartIndex(1);
fit->SetEndIndex(n_talls);
fit->SetIncrementIndex(1);
fit->SetSeriesFormat ("tall%03d.txt");

std::vector<std::string> names = fit->GetFileNames();
std::vector<std::string>::iterator nit;
int x=0;
int taula[n_talls]; //taula de n posicions on guarda el valor de Z

for (nit = names.begin(); nit != names.end(); nit++)
{
    //Cal llegir la profunditat de cada tall confocal
    FILE *file;
    char *pch;
    char *profunditat;
    char string [50];
    int i=0;
    int prof;
    file = fopen ((*nit).c_str(), "r");
    if (file == NULL) perror ("Error opening file");
    else {
        while( i<11)
        {
            fgets (string , 50 , file);
            i++;
        }
        fclose (file);

        pch=strtok(string,"=");
        while(pch != NULL)
        {
            pch = strtok( NULL, "=");
            if (strcmp(pch,"Zrel"))
            {
                profunditat=pch;
                prof= atol(profunditat);
                taula[x]=prof;
            }
        }
    }
}
```



```

        std::cout << "taula " << taula[x] <<std::endl;
        x++;

    }
    pch= NULL;
} //fi while
std::cout << "File: " << (*nit).c_str() << std::endl;
} //fi for
//Normalitzem els valors de la profunditat per que comencin de 0.
//Això ho fem restant el valor mínim de profunditat a totes les altres.
int w=0;
int minim = taula[0];
while(w<n_talls){
    taula[w] = taula[w] - minim;
    w++;
}

```

Ara crearem el volum 3D on anirem insertant els talls de pell en la seva corresponent profunditat. Al crear un volum amb les profunditats reals, entre els espais buits hem aplicat en mètode per introduir talls registrats amb l'última profunditat real.

```

InputImageType::SizeType midaVolumOrg = volumOrg-
>GetBufferedRegion().GetSize();

    InputImageType::SizeType midaVolumResult;

    midaVolumResult = midaVolumOrg; //igualo la mida en x,y del volum
Original al Volum Resultant
    midaVolumResult[2] = profMax; //Possem la profunditat maxima com mida
final de z ([2]) del volum Resultant

    ImageType::Pointer volumeResult = ImageType::New();
    volumeResult->SetRegions(midaVolumResult);
    volumeResult->Allocate();
    volumeResult->FillBuffer(128);

```

9.2.3 Diferència

Les principals llibreries per fer la diferència entre imatges són les següents:

```

#include "itkAbsoluteValueDifferenceImageFilter.h"
#include "InverterIntensityImageFilter.h"
typedef itk::AbsoluteValueDifferenceImageFilter< FixedImageType,
FixedImageType, OutputImageType > DifferenceFilterType;
typedef itk::InvertIntensityImageFilter< OutputImageType, OutputImageType >
InverterType;

```

En el tema de llibreries per fer la diferència vam utilitzar dos tipus, la itkSubtractImageFilter i la itkAbsoluteValueDifferenceImageFilter. Després de fer moltes proves, vam decantar-nos per la segona ja que donava les diferències en valor absolut i és més fàcil observar-ne els resultats.

Un cop tenim llegides les imatges les haurem d'afegir a la pipeline per poder calcular la diferència:

```
DifferenceFilterType::Pointer difference = DifferenceFilterType::New();
difference->SetInput1( fixedImageReader->GetOutput() );
difference->SetInput2( movingImageReader1->GetOutput() );
```

Un cop ho tenim preparat invertim la intensitat de la imatge per que es vegi molt més bé la imatge diferència resultant i la guardem en un nou fitxer png:

```
InverterType::Pointer intensityInverter = InverterType::New();
intensityInverter->SetInput( difference->GetOutput() );
writer->SetInput( intensityInverter->GetOutput() );
writer->SetFileName( "diferencia.png" );
writer->Update();
```

9.2.4 Checkerboard

Les llibreries més destacades per calcular el checkerboard són les següents:

```
#include "itkCastImageFilter.h"
#include "itkCheckerBoardImageFilter.h"
typedef itk::CastImageFilter< FixedImageType, OutputImageType >
CastFilterType;
typedef itk::CheckerBoardImageFilter< FixedImageType > CheckerBoardFilterType;
```

Un cop tenim llegides les imatges les haurem d'afegir a la pipeline per poder calcular el checker:

```
CheckerBoardFilterType::Pointer checker = CheckerBoardFilterType::New();
checker->SetInput1( fixedImageReader->GetOutput() );
checker->SetInput2( movingImageReader1->GetOutput() );
```

Un cop ho tenim canviem el tipus de píxels de real a unsigned char mitjançant un casting i guardem la nova imatge resultant en un nou fitxer png:

```
caster->SetInput( checker->GetOutput() );
writer->SetFileName( "checker_dp_registre.png" );
writer->SetInput( caster->GetOutput() );
writer->Update();
```

9.2.5 Visualització 3D

El volum registrat serà llegit mitjançant les ITK i serà convertit a VTK per poder ser visualitzat. La classe itkImageToVTKImageFilter ens permetrà passar de ITK a VTK. El mètode usat és el següent:

```
template <typename VTK_Exporter, typename ITK_Importer>
void ConnectPipelines(VTK_Exporter* exporter, ITK_Importer importer)
{
importer->SetUpdateInformationCallback(exporter-
>GetUpdateInformationCallback());
importer->SetPipelineModifiedCallback(exporter-
>GetPipelineModifiedCallback());
importer->SetWholeExtentCallback(exporter->GetWholeExtentCallback());
importer->SetSpacingCallback(exporter->GetSpacingCallback());
importer->SetOriginCallback(exporter->GetOriginCallback());
importer->SetScalarTypeCallback(exporter->GetScalarTypeCallback());
importer->SetNumberOfComponentsCallback(exporter-
```

```

>GetNumberOfComponentsCallback() ;
importer->SetPropagateUpdateExtentCallback (exporter-
>GetPropagateUpdateExtentCallback() );
importer->SetUpdateDataCallback (exporter->GetUpdateDataCallback() );
importer->SetDataExtentCallback (exporter->GetDataExtentCallback() );
importer->SetBufferPointerCallback (exporter->GetBufferPointerCallback() );
importer->SetCallbackUserData (exporter->GetCallbackUserData() );
}

```

Creem l'objecte mitjançant el seu constructor i tot seguit actualitzem la pipeline,

```

typedef itk::ImageToVTKImageFilter<InputImageType> ConnectorTypeVTK;
ConnectorTypeVTK::Pointer connector= ConnectorTypeVTK::New();
connector->SetInput(reader->GetOutput()); //itk -> vtk
connector->Update();

```

Realitzarem en vtk el nou volum 3D passant-li els paràmetres del color, opacitat, interpolació...

```

vtkVolume *newvol = vtkVolume::New();
newvol->SetMapper(volumeMapper);
newvol->GetProperty()->SetColor(ctfun);
newvol->GetProperty()->SetScalarOpacity(tfun);
newvol->GetProperty()->SetInterpolationTypeToLinear();
newvol->GetProperty()->ShadeOn();

```

Un cop tenim l'objecte que volem visualitzar preparat l'hem d'incloure a l'escena. Creem un actor per el volum 3D (outlineActor) i altre actor per el tall histològic d'un eix que també mostrarem en escena (imageActor).

```

vtkActor *outlineActor = vtkActor::New();
outlineActor->SetMapper(outlineMapper);

vtkImageActor* imageActor = vtkImageActor::New();
imageActor->SetMapper(isoMapper);
imageActor->GetProperty()->SetDiffuseColor(1,1,0.9412);
imageActor->SetProperty(property);

vtkRenderer* ren = vtkRenderer::New();
vtkRenderer* ren2= vtkRenderer::New();
vtkRenderWindow* renwin = vtkRenderWindow::New();
vtkRenderWindowInteractor* iren = vtkRenderWindowInteractor::New();

```

Ahora de visualitzar el volum podrem definir-li propietats com per exemple el color del volum, transparència, etc.

```

vtkProperty *property = vtkProperty::New();
property->SetColor(0.9412,0.9412,0.9412);
property->SetDiffuse(0.7);
property->SetSpecular(0.4);
property->SetSpecularPower(20);
property->EdgeVisibilityOn();

```

Crearem tres nous objectes corresponents a els tres plans ens els eixos X,Y i Z. Això ho farem mitjançant Planewidgets.

El PlaneWidget és un objecte creat per treballar amb imatges 2D, però exclusivament per els plans 2D dels eixos X,Y i Z.

Seguidament mostrem com crear un pla de l'eix de coordenades X. Seleccionem cadascun dels talls del volum a visualitzar i els anem afegint a l'actor que serà visualitzat a l'escena,

```
planeWidgetX = vtkImagePlaneWidget::New();
planeWidgetY = vtkImagePlaneWidget::New();
planeWidgetZ = vtkImagePlaneWidget::New();

planeWidgetX->SetInteractor( iren);
planeWidgetX->SetKeyPressActivationValue('x');
planeWidgetX->SetPicker(picker);
planeWidgetX->RestrictPlaneToVolumeOn();
planeWidgetX->GetPlaneProperty()->SetColor(1,0,0);
planeWidgetX->SetTexturePlaneProperty(ipwProp);
planeWidgetX->TextureInterpolateOff();
planeWidgetX->SetResliceInterpolateToNearestNeighbour();
planeWidgetX->SetInput(connector->GetOutput());
planeWidgetX->SetPlaneOrientationToXAxes();
planeWidgetX->SetSliceIndex(500); //32
planeWidgetX->DisplayTextOn();
planeWidgetX->On();
planeWidgetX->InteractionOff();
planeWidgetX->InteractionOn();
```

Declarem el mapa de colors de cada pla.

```
vtkImageMapToColors* colorMap = vtkImageMapToColors::New();
colorMap->PassAlphaToOutputOff();
colorMap->SetActiveComponent(0);
colorMap->SetOutputFormatToLuminance();
colorMap->SetInput(planeWidgetZ->GetResliceOutput());
colorMap->SetLookupTable(planeWidgetZ->GetLookupTable());
```

Adherim els actors als renders per poder visualitzar els objectes. Estan possats el volum, la superfície del volum i el tall de l'eix escollit per visualitzar.

```
ren2->AddActor( imageActor);
ren->AddActor( outlineActor);
ren->AddActor( newvol);
```

Definim les propietats de l'escena i l'iniciem,

```
renwin->SetSize(600, 500);
renwin->AddRenderer( ren );
renwin->AddRenderer( ren2 );
iren->SetRenderWindow(renwin);

ren->SetBackground( 0.1, 0.1, 0.2);
ren2->SetBackground( 0.6, 0.6, 0.6);

ren->SetViewport(0,0,0.58333,1);
ren2->SetViewport(0.5,0,1,1);

renwin->Render();
iren->Start();
```

Activem els parametres de la càmera

```
ren->GetActiveCamera()->Elevation(110);
ren->GetActiveCamera()->SetViewUp(0, 0, -1);
ren->GetActiveCamera()->Azimuth(45);
ren->GetActiveCamera()->Dolly(1.15);
ren->ResetCameraClippingRange();
```

Crearem una classe vtkCommand per interactuar amb el perifèric del teclat, i crear els events corresponents per treballar amb l'interfície.

```
class vtkEventApretatCallback : public vtkCommand
{
public:
    static vtkEventApretatCallback *New()
    { return new vtkEventApretatCallback; }
    virtual void Execute(vtkObject *caller, unsigned long eventId, void*)
    {
        int aux;
        std::cout<<"dins! " <<std::endl;
        vtkRenderWindowInteractor *iren =
reinterpret_cast<vtkRenderWindowInteractor*>(caller);
        switch (eventId) {
            case vtkCommand::KeyPressEvent:
            {
                char keypressed = *(iren -> GetKeySym());
                std::cout<<"Tecla! " << keypressed <<std::endl;
                if( keypressed== amunt)
                {
                    switch(modevis){
                        case 0: aux = indexX++; break;
                        case 1: aux = indexY++; break;
                        case 2: aux = indexZ++; break;
                    }
                    widgetActual->SetSliceIndex(aux);
                }
                else if ( keypressed== avall)
                {
                    switch(modevis){
                        case 0: aux = indexX--; break;
                        case 1: aux = indexY--; break;
                        case 2: aux = indexZ--; break;
                    }
                    widgetActual->SetSliceIndex(aux); //32
                }

                if( keypressed== 'x')
                {
                    cout << "Mode X" << endl;
                    modevis = 0;
                    widgetActual = planeWidgetX;

                    colorMap->SetInput(planeWidgetX-
>GetResliceOutput());
                    colorMap->UpdateWholeExtent();
                    imageActor->SetDisplayExtent(imageActor->GetInput()-
>GetWholeExtent());

                    ren2->ResetCamera();
                    ren2->Render();
                }
                if( keypressed== 'y')
                {
                    cout << "Mode Y" << endl;
                    modevis = 1;
                    widgetActual = planeWidgetY;
```

```

                                colorMap->SetInput(planeWidgetY-
>GetResliceOutput());
                                colorMap->UpdateWholeExtent();
                                imageActor->SetDisplayExtent(imageActor->GetInput()-
>GetWholeExtent());
                                ren2->ResetCamera();
                                ren2->Render();
                                }
                                if( keypressed== 'z')
                                {
                                    cout << "Mode Z" << endl;

                                    modevis = 2;
                                    widgetActual = planeWidgetZ;

                                colorMap->SetInput(planeWidgetZ-
>GetResliceOutput());
                                colorMap->UpdateWholeExtent();
                                imageActor->SetDisplayExtent(imageActor->GetInput()-
>GetWholeExtent());
                                ren2->ResetCamera();
                                ren2->Render();
                                }
                                renwin->Render();
                                }//final case
                                }//final switch
                                }//final Virtual
                                };//final calss

```

Finalment, caldrà lliberar tot l'espai dels objectes creats, utilitzant la funció Delete(). Això es fa al final del nostre algoritme.

```

colorMap->Delete();
imageActor->Delete();
picker->Delete();
outlineActor->Delete();
outlineMapper->Delete();
outline->Delete();
iren->Delete();
renwin->Delete();
ren->Delete();
ren2->Delete();
newvol->Delete();

```

B. Annex. Implementació eina visual

Jaume Delgado Fernandez
Enginyeria Tècnica en Informàtica de Sistemes.

Universitat de Girona, 2006

10. Annex. Implementació eina visual

10.1 Implementació

En aquest apartat mostrarem la metodologia i el codi que apliquem en el disseny de la interfície visual.

Està dividit en dues parts, com són l'arxiu .h i el .cpp. A continuació mostrem el codi del arxiu visual.h i visual.cpp en les taules Taula 10.1. Codi per el disseny de l'eina visual. i Taula 10.2.

```
#ifndef Visual_H
#define Visual_H

#include <qvariant.h>
#include <qdialog.h>
#include "formulari.h"
//-----
//   Itk-Header Files
//-----

#include "itkImageFileReader.h"
#include "itkImageFileWriter.h"
#include "itkExtractImageFilter.h"
#include "itkImage.h"
#include "itkResampleImageFilter.h"
#include "itkAffineTransform.h"
#include "itkImageSeriesReader.h"
#include "itkImageSeriesWriter.h"
#include "itkNumericSeriesFileNames.h"
#include "itkVTKImageIO.h"
#include "itkPNGImageIO.h"
#include "itkRescaleIntensityImageFilter.h"
#include "itkImageToVTKImageFilter.h"

using namespace std;
class vtkEventApretatCallback;

//-----
//   Vtk-Header Files
//-----
#include "vtkRenderer.h"
#include "vtkRenderWindow.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkActor.h"
#include "vtkOutlineFilter.h"
#include "vtkCamera.h"
#include "vtkLookupTable.h"
#include "vtkProperty.h"
#include "vtkPolyDataNormals.h"
#include "vtkContourFilter.h"
#include "vtkImageMapToColors.h"
#include "vtkImageActor.h"
#include "vtkSmoothPolyDataFilter.h"
#include "vtkImageMarchingCubes.h"
#include "vtkPolyDataMapper.h"
#include "vtkInteractorStyleTrackballCamera.h"
#include "vtkCommand.h"
#include "vtkBoxWidget.h"
#include "vtkTransform.h"
#include "vtkPiecewiseFunction.h"
#include "vtkColorTransferFunction.h"
```

```

#include "vtkVolumeRayCastCompositeFunction.h"
#include "vtkVolumeRayCastMapper.h"
#include "vtkVolumeProperty.h"
#include "vtkInteractorEventRecorder.h"
#include "vtkImagePlaneWidget.h"
#include "vtkCellPicker.h"
#include "vtkRegressionTestImage.h"
#include "vtkImagePlaneWidget.h"
#include "vtkCylinderSource.h"
#include "vtkPolyDataMapper.h"
#include "vtkLightKit.h"

class Visual: public Form2
{
    Q_OBJECT

public:
    Visual( QWidget* parent = 0, const char* name = 0, bool modal = FALSE,
WFlags fl = 0 );
    ~Visual();

    //=====
    // Your member data
    //=====
    QString DirName, FileName;

    vtkCylinderSource* source;
    vtkPolyDataMapper* mapper;
    vtkActor* actor;
    vtkLightKit* lightKit;

    // talls data
    typedef itk::Image< unsigned char, 3 > InputImageType;
    typedef itk::Image< unsigned char, 2 > OutputImageType;
    typedef itk::Image< unsigned char, 2 > OutputImageTypeChar;
    typedef itk::ImageFileReader< InputImageType > ReaderType;
    typedef          itk::ResampleImageFilter<OutputImageType,OutputImageType>
FiltreCanvi;
    typedef          itk::RescaleIntensityImageFilter<          OutputImageType,
OutputImageTypeChar > RescalerType;
    typedef itk::AffineTransform<double,2> TransformType;
    typedef itk::ImageToVTKImageFilter<InputImageType> ConnectorTypeVTK;
    vtkImagePlaneWidget* planeWidgetX;
    vtkImagePlaneWidget* planeWidgetY;
    vtkImagePlaneWidget* planeWidgetZ;
    vtkCellPicker* picker;
    vtkImageMapToColors* colorMap;
    vtkImageActor* imageActor;
    vtkRenderer* ren2;
    vtkRenderWindowInteractor* iren;
    vtkRenderer* ren;
    vtkProperty* ipwProp;
    vtkVolume *newvol;
    vtkActor *outlineActor;
    vtkVolumeRayCastCompositeFunction *compositeFunction;
    vtkVolumeRayCastMapper *volumeMapper;
    vtkPolyDataMapper *outlineMapper;
    vtkOutlineFilter *outline;
    vtkEventApretatCallback *callEventApretatback;

    ConnectorTypeVTK::Pointer connector;
    ReaderType::Pointer reader;
    float      cWinZoom;

    //=====
    // Your member function
    //=====

```

```

public:
    void CleanMemory();
public:

//=====
// Your GUI functions
//=====
public slots:
    void LoadDataset();
    void Render();
    void Exit();
    void ZoomIn();
    void SetObjectColor();
    void SetBackgroundColor();
    void EixZ();
    void EixY();
    void EixX();
    void BaixarSlice();
    void PujarSlice();
    void PujarMoltsSlices();
    void BaixarMoltsSlices();
};

#endif // VISUAL_H

```

Taula 10.1. Codi per el disseny de l'eina visual.

La següent Taula ens mostra el codi del funcionament de l'eina per visualitzar les regions de la pell.

```

/*=====
Programa:   Eina Visual (C++/Vtk/itk/Qt)
Autor:     Jaume Delgado
Llenguatge: C++/Vtk/itk/Qt
=====*/

#include <qapplication.h>
#include "QVTKWidget.h"
#include "visual.h"
#include <qcolordialog.h>
#include <qpushbutton.h>
#include <stdlib.h>
#include <qlabel.h>
#include <qlineedit.h>
#include <qfiledialog.h>
#include <qpixmap.h>
#include <qdir.h>
#include <qcheckbox.h>
#include <qcolordialog.h>

//de tall.cxx
#include "itkImageFileReader.h"
#include "itkImageFileWriter.h"
#include "itkExtractImageFilter.h"
#include "itkImage.h"
#include "itkResampleImageFilter.h"
#include "itkAffineTransform.h"
#include "itkImageSeriesReader.h"
#include "itkImageSeriesWriter.h"
#include "itkNumericSeriesFileNames.h"
#include "itkVTKImageIO.h"
#include "itkPNGImageIO.h"
#include "itkRescaleIntensityImageFilter.h"
#include "itkImageToVTKImageFilter.h"

```

```

#include "vtkRenderer.h"
#include "vtkRenderWindow.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkActor.h"
#include "vtkOutlineFilter.h"
#include "vtkCamera.h"
#include "vtkLookupTable.h"
#include "vtkProperty.h"
#include "vtkPolyDataNormals.h"
#include "vtkContourFilter.h"
#include "vtkImageMapToColors.h"
#include "vtkImageActor.h"
#include "vtkSmoothPolyDataFilter.h"
#include "vtkImageMarchingCubes.h"
#include "vtkPolyDataMapper.h"
#include "vtkInteractorStyleTrackballCamera.h"
#include "vtkCommand.h"
#include "vtkBoxWidget.h"
#include "vtkTransform.h"
#include "vtkPiecewiseFunction.h"
#include "vtkColorTransferFunction.h"
#include "vtkVolumeRayCastCompositeFunction.h"
#include "vtkVolumeRayCastMapper.h"
#include "vtkVolumeProperty.h"
#include "vtkInteractorEventRecorder.h"
#include "vtkImagePlaneWidget.h"
#include "vtkCellPicker.h"
#include "vtkRegressionTestImage.h"
#include "vtkImagePlaneWidget.h"
#include "vtkLightKit.h"

class vtkCylinderSource;
class vtkPolyDataMapper;
class vtkExodusReader;
class vtkDataSetMapper;
class vtkActor;
class vtkRenderer;

#define PATH "."
#define FILTER "*.vtk"

class vtkEventApretatCallback : public vtkCommand
{
    // atributs
public:
    int indexX , indexY, indexZ;
    int modevis;
    vtkImagePlaneWidget* widgetActual;
    Visual * pare;
    int aux;

public:
    static vtkEventApretatCallback *New()
    { return new vtkEventApretatCallback; }
    virtual void Execute(vtkObject *caller, unsigned long eventId, void*)
    {
        //int aux;
        char amunt='U';//tecla del cursor cap adalt
        char avall='D';//tecla del cursor cap abaix
        std::cout<<"dins! " <<std::endl;
        vtkRenderWindowInteractor *iren =
reinterpret_cast<vtkRenderWindowInteractor*>(caller);
        switch (eventId) {
            case vtkCommand::KeyPressEvent:

```

```

{
char keypressed = *(iren -> GetKeySym());
std::cout<<"Tecla! "<< keypressed <<std::endl;
if( keypressed== amunt)
{
switch(modevis){
case 0: aux = indexX++; break;
case 1: aux = indexY++; break;
case 2: aux = indexZ++; break;
}
widgetActual->SetSliceIndex(aux); //32
pare->ren2->Render();
pare->ren->Render();
pare->ren2->ResetCamera();
pare->ren2->GetRenderWindow()->Render();
pare->ren->GetRenderWindow()->Render();

}
else if ( keypressed== avall)
{
switch(modevis){
case 0: aux = indexX--; break;
case 1: aux = indexY--; break;
case 2: aux = indexZ--; break;
}
widgetActual->SetSliceIndex(aux); //32
pare->ren2->Render();
pare->ren->Render();
pare->ren2->ResetCamera();
pare->ren2->GetRenderWindow()->Render();
pare->ren->GetRenderWindow()->Render();

}

if( keypressed== 'x')
{
cout << "Mode X" << endl;
modevis = 0;
widgetActual = pare->planeWidgetX;

pare->colorMap->SetInput(widgetActual-
>GetResliceOutput());
pare->colorMap->UpdateWholeExtent();
pare->imageActor->SetDisplayExtent(pare->imageActor-
>GetInput()->GetWholeExtent());
pare->ren2->ResetCamera();
pare->ren2->Render();
}
if( keypressed== 'y')
{
cout << "Mode Y" << endl;
modevis = 1;
widgetActual = pare->planeWidgetY;

pare->colorMap->SetInput(widgetActual-
>GetResliceOutput());
pare->colorMap->UpdateWholeExtent();
pare->imageActor->SetDisplayExtent(pare->imageActor-
>GetInput()->GetWholeExtent());
pare->ren2->ResetCamera();
pare->ren2->Render();
}
if( keypressed== 'z')
{
cout << "Mode Z" << endl;
}
}

```

```

modevis = 2;
widgetActual = pare->planeWidgetZ;

pare->colorMap->SetInput(widgetActual-
>GetResliceOutput());
pare->colorMap->UpdateWholeExtent();
pare->imageActor->SetDisplayExtent(pare->imageActor-
>GetInput()->GetWholeExtent());
pare->ren2->ResetCamera();
pare->ren2->Render();
}
// comentat!! renwin->Render();
} //final case
} //final switch
} //final Virtual
}; //final calss

//*****
*****
Visual::Visual( QWidget* parent, const char* name, bool modal, WFlags fl )
: Form2( parent, name, modal, fl ) {
//*****
*****

connect( bt_load, SIGNAL( clicked() ) , this , SLOT( LoadDataset() ) );
connect( ren_bt, SIGNAL( clicked() ) , this , SLOT( Render() ) );
connect( Sortir, SIGNAL( clicked() ) , this , SLOT( Exit() ) );
connect( ColorBackground, SIGNAL( clicked() ) , this , SLOT(
SetBackgroundColor() ) );
connect( Color, SIGNAL( clicked() ) , this , SLOT( SetObjectColor() ) );
connect( ckEixZ, SIGNAL( clicked() ) , this , SLOT( EixZ() ) );
connect( ckEixX, SIGNAL( clicked() ) , this , SLOT( EixX() ) );
connect( ckEixY, SIGNAL( clicked() ) , this , SLOT( EixY() ) );
connect( Pujar, SIGNAL( clicked() ) , this , SLOT( PujarSlice() ) );
connect( Baixar, SIGNAL( clicked() ) , this , SLOT( BaixarSlice() ) );
connect( AvPagina, SIGNAL( clicked() ) , this , SLOT( PujarMoltsSlices() )
);
connect( RePagina, SIGNAL( clicked() ) , this , SLOT( BaixarMoltsSlices() )
);

setCaption("Eina Visual");
NullObjects();
ren = vtkRenderer::New();
ren2 = vtkRenderer::New();
iren = vtkRenderWindowInteractor::New();
callEventApretatback = vtkEventApretatCallback::New();
callEventApretatback->pare = this;

//Afegeixo els Renders a les finestres on es visulaitzarà els volums.
vtkWidgetE->GetRenderWindow()->SetInteractor(iren);
vtkWidgetE->GetRenderWindow()->AddRenderer(ren);
vtkWidgetD->GetRenderWindow()->AddRenderer(ren2);

ren_bt->setEnabled(true);

}

//*****
void Visual::CleanMemory() {
//*****
}

//*****
Visual::~Visual() {
//*****
}

```

```

CleanMemory();
}

//*****
void Visual::LoadDataset() {
//*****
// This function extract the volume sizes (slices, width, height)
QFileDialog dlg;

QString filePath = dlg.getOpenFileName(PATH,
    FILTER,
    this,
    "Open File Dialog"
    "Select any file from the dataset");

if(filePath.isEmpty() || filePath.isNull()) {
    status->setText("Error: You did not select a file..!");
    return;
} else {

    DirName = filePath.left(filePath.findRev('/')+1);
    FileName = filePath;
    status->setText(FileName);
    ren_bt->setEnabled(true);
}
}

//*****
void Visual::Render() {
//*****

    status->setText( tr( "Renderitzant en Proces...!" ) );
    callEventApretatback->modevis = 2; // current_plane a z

    connector= ConnectorTypeVTK::New();
    reader = ReaderType::New();

    reader->SetFileName( FileName );
    reader->Update();

    InputImageType::RegionType inputRegion = reader->GetOutput()-
>GetLargestPossibleRegion();
    InputImageType::SizeType size = inputRegion.GetSize();
    InputImageType::IndexType start = inputRegion.GetIndex();
    InputImageType::RegionType desiredRegion;

    // passem el volum ITK a VTK per poder visualitzar
    connector->SetInput( reader->GetOutput() ); //itk -> vtk
    connector->Update();

    ////////////////
    vtkPiecewiseFunction *tfun = vtkPiecewiseFunction::New();
    tfun->AddPoint(10.0,0.0);
    tfun->AddPoint(255.0,1.0);

    vtkColorTransferFunction *ctfun = vtkColorTransferFunction::New();
    ctfun->AddRGBPoint(0.0,0.5,0.0,0.0);
    ctfun->AddRGBPoint(255.0,0.81,0.27,0.1);

    compositeFunction = vtkVolumeRayCastCompositeFunction::New();

    //Carreguem el volum d'entrada procedent de ITK al volum VTK
    volumeMapper = vtkVolumeRayCastMapper::New();
    volumeMapper->SetInput( connector->GetOutput() );
    volumeMapper->SetVolumeRayCastFunction( compositeFunction );

    //Creem el volum 3D amb vtk passant-li els parametres del

```



```

color,Opacitat,Inteerpolació...
    newvol = vtkVolume::New();
    newvol->SetMapper(volumeMapper);
    newvol->GetProperty()->SetColor(ctfun);
    newvol->GetProperty()->SetScalarOpacity(tfun);
newvol->GetProperty()->SetInterpolationTypeToLinear();
newvol->GetProperty()->ShadeOn();

    vtkOutlineFilter *outline = vtkOutlineFilter::New();
outline->SetInput(connector->GetOutput());
    vtkPolyDataMapper *outlineMapper = vtkPolyDataMapper::New();
outlineMapper->SetInput(outline->GetOutput());
    outlineActor = vtkActor::New();
outlineActor->SetMapper(outlineMapper);

    picker = vtkCellPicker::New();
picker->SetTolerance(0.005);
    // Plane stuff
    planeWidgetX = vtkImagePlaneWidget::New();
    planeWidgetY = vtkImagePlaneWidget::New();
    planeWidgetZ = vtkImagePlaneWidget::New();

    //////////////////////////////////////
    //***** Pla d'eixos Z *****
    //////////////////////////////////////

    callEventApretatback->indexZ = size[2]/2;
planeWidgetZ->RestrictPlaneToVolumeOn();
planeWidgetZ->SetResliceInterpolateToNearestNeighbour();

planeWidgetZ->SetInteractor(iren);
planeWidgetZ->DisplayTextOn();
planeWidgetZ->SetKeyPressActivationValue('z');
planeWidgetZ->SetPicker(picker);
planeWidgetZ->GetPlaneProperty()->SetColor(0,0,1);
planeWidgetZ->SetInput(connector->GetOutput());
planeWidgetZ->SetPlaneOrientationToZAxes();
planeWidgetZ->SetSliceIndex(callEventApretatback->indexZ); //25
planeWidgetZ->SetLookupTable(planeWidgetZ->GetLookupTable());
planeWidgetZ->DisplayTextOn();
planeWidgetZ->On();
planeWidgetZ->InteractionOff();
planeWidgetZ->InteractionOn();

    //////////////////////////////////////
    //***** Pla d'eixos Y *****
    //////////////////////////////////////
    callEventApretatback->indexY = size[1]/2;
planeWidgetY->SetInteractor(iren);
planeWidgetY->SetKeyPressActivationValue('y');
planeWidgetY->SetPicker(picker);
planeWidgetY->GetPlaneProperty()->SetColor(0,0,1);
planeWidgetY->SetInput(connector->GetOutput());
planeWidgetY->SetPlaneOrientationToYAxes();
planeWidgetZ->SetSliceIndex(callEventApretatback->indexY);
planeWidgetY->SetLookupTable(planeWidgetX->GetLookupTable());
planeWidgetY->DisplayTextOn();
planeWidgetY->Off();

    //////////////////////////////////////
    //***** Pla d'eixos X *****
    //////////////////////////////////////
    callEventApretatback->indexX = size[0]/2;
planeWidgetX->SetInteractor(iren);
planeWidgetX->SetKeyPressActivationValue('x');
planeWidgetX->SetPicker(picker);
planeWidgetX->GetPlaneProperty()->SetColor(0,0,1);
planeWidgetX->SetInput(connector->GetOutput());

```

```

planeWidgetX->SetPlaneOrientationToXAxes();
planeWidgetX->SetLookupTable( planeWidgetX->GetLookupTable());
planeWidgetX->DisplayTextOn();
    planeWidgetX->Off();

    //Inicialitzem el ColorMap
    colorMap = vtkImageMapToColors::New();
colorMap->PassAlphaToOutputOff();
colorMap->SetActiveComponent(0);
colorMap->SetOutputFormatToLuminance();
colorMap->SetInput(planeWidgetZ->GetResliceOutput());
colorMap->SetLookupTable(planeWidgetZ->GetLookupTable());

    //Creem el ImageActor que serà l'actor que ens mostrarà el tall
    escollit.
    imageActor = vtkImageActor::New();
    imageActor->PickableOff();
    imageActor->SetInput(colorMap->GetOutput());

    vtkInteractorStyleTrackballCamera          *style          =
vtkInteractorStyleTrackballCamera::New();
    iren->SetInteractorStyle(style);

    callEventApretatback->widgetActual = planeWidgetZ;
    iren->AddObserver(vtkCommand::KeyPressEvent, callEventApretatback);

    //-----
// Adjust Light
//-----
    iren->LightFollowCameraOff();
    ren->LightFollowCameraOn();
    lightKit = vtkLightKit::New();
    lightKit->AddLightsToRenderer(ren);

// Add Actor to renderer
    ren->AddActor(newvol);
    ren->AddActor(outlineActor);

    iren->Initialize();

    ren->ResetCamera();
    ren->GetRenderWindow()->Render();

    iren->Start();

    //Netegem la memòria
status->setText( tr( "Netejant Memoria...!" ) );
    CleanMemory();

    status->setText( tr("Comença la Visualització, espereu uns segons..."));
}

//*****
void Visual::EixZ() {
//*****

    // comprovar si esta activat
    if(ckEixZ->isChecked())
    {
        planeWidgetZ->On();
        callEventApretatback->modevis = 2;
        callEventApretatback->widgetActual = planeWidgetZ;

        colorMap->SetInput(callEventApretatback->widgetActual-
>GetResliceOutput());

```

```

        colorMap->UpdateWholeExtent();
        imageActor->SetDisplayExtent(imageActor->GetInput()->GetWholeExtent());
        ren2->ResetCamera();
        ren2->Render();
    }
    else{
        //desactivar el pla Z
        planeWidgetZ->Off();
    }
    ren2->AddActor(imageActor);
    ren->ResetCamera();
    ren2->GetRenderWindow()->Render();
}

//*****
void Visual::EixX() {
//*****

    // comprovar si esta activat
    if(ckEixX->isChecked())
    {
        planeWidgetX->On();
        callEventApretatback->modevis = 0;
        callEventApretatback->widgetActual = planeWidgetX;

        colorMap->SetInput(callEventApretatback->widgetActual-
>GetResliceOutput());
        colorMap->UpdateWholeExtent();
        imageActor->SetDisplayExtent(imageActor->GetInput()->GetWholeExtent());
        ren2->ResetCamera();
        ren2->Render();
    }
    else{
        //desactivar el pla X
        planeWidgetX->Off();
    }

    ren2->AddActor(imageActor);
    ren->ResetCamera();
    ren2->GetRenderWindow()->Render();
}

//*****
void Visual::EixY() {
//*****

    // comprovar si esta activat
    if(ckEixY->isChecked())
    {
        planeWidgetY->On();
        callEventApretatback->modevis = 1;
        callEventApretatback->widgetActual = planeWidgetY;

        colorMap->SetInput(callEventApretatback->widgetActual-
>GetResliceOutput());
        colorMap->UpdateWholeExtent();
        imageActor->SetDisplayExtent(imageActor->GetInput()->GetWholeExtent());
        ren2->ResetCamera();
        ren2->Render();
    }
    else{
        //desactivar el pla Y
        planeWidgetY->Off();
    }
    ren2->AddActor(imageActor);
    ren->ResetCamera();
    ren2->GetRenderWindow()->Render();
}
}

```

```

//*****
void Visual::BaixarSlice() {
//*****
    switch(callEventApretatback->modevis){
        case 0: callEventApretatback->aux = callEventApretatback-
>indexX++; break;
        case 1: callEventApretatback->aux = callEventApretatback-
>indexY++; break;
        case 2: callEventApretatback->aux = callEventApretatback-
>indexZ++; break;
    }
    callEventApretatback->widgetActual-
>SetSliceIndex(callEventApretatback->aux); //32
    ren2->Render();
    ren->Render();
    ren2->ResetCamera();
    ren2->GetRenderWindow()->Render();
    ren->GetRenderWindow()->Render();
}

//*****
void Visual::BaixarMoltsSlices() {
//*****
    switch(callEventApretatback->modevis){
        case 0: callEventApretatback->aux = callEventApretatback-
>indexX+10; break;
        case 1: callEventApretatback->aux = callEventApretatback-
>indexY+10; break;
        case 2: callEventApretatback->aux = callEventApretatback-
>indexZ+10; break;
    }
    callEventApretatback->widgetActual-
>SetSliceIndex(callEventApretatback->aux); //32
    ren2->Render();
    ren->Render();
    ren2->ResetCamera();
    ren2->GetRenderWindow()->Render();
    ren->GetRenderWindow()->Render();
}

//*****
void Visual::PujarMoltsSlices() {
//*****
    switch(callEventApretatback->modevis){
        case 0: callEventApretatback->aux = callEventApretatback->indexX-
10; break;
        case 1: callEventApretatback->aux = callEventApretatback->indexY-
10; break;
        case 2: callEventApretatback->aux = callEventApretatback->indexZ-
10; break;
    }
    callEventApretatback->widgetActual-
>SetSliceIndex(callEventApretatback->aux); //32
    ren2->Render();
    ren->Render();
    ren2->ResetCamera();
    ren2->GetRenderWindow()->Render();
    ren->GetRenderWindow()->Render();
}

//*****
void Visual::PujarSlice() {
//*****

```

```

        switch(callEventApretatback->modevis) {
            case 0: callEventApretatback->aux = callEventApretatback->indexX-
-; break;
            case 1: callEventApretatback->aux = callEventApretatback->indexY-
-; break;
            case 2: callEventApretatback->aux = callEventApretatback->indexZ-
-; break;
        }
        callEventApretatback->widgetActual-
>SetSliceIndex(callEventApretatback->aux); //32
        ren2->Render();
        ren->Render();
        ren2->ResetCamera();
        ren2->GetRenderWindow()->Render();
        ren->GetRenderWindow()->Render();
    }

//*****
void Visual::Exit() {
//*****
    status->setText( tr( "Sortint...!" ) );
    CleanMemory();
    close();
}

//*****
void Visual::SetObjectColor() {
//*****
    // change the object color
    QColor c = QColorDialog::getColor();
    outlineActor->GetProperty()->SetColor(c.red()/255.0,    c.green()/255.0,
c.blue()/255.0);
    ren2->GetRenderWindow()->Render();
}

//*****
void Visual::SetBackgroundColor() {
//*****
    // change the background color
    QColor c = QColorDialog::getColor();
    ren2->SetBackground(c.red()/255.0, c.green()/255.0, c.blue()/255.0);
    ren2->GetRenderWindow()->Render();
    ren->SetBackground(c.red()/255.0, c.green()/255.0, c.blue()/255.0);
    ren->GetRenderWindow()->Render();
}

//*****
void Visual::ZoomIn()
//*****
{
    cWinZoom += 1;
    update();
    this->update();
}

/* Aquesta funció permet connectar ITK amb VTK. Per poder visualitzar el
volum3D d'entrada.Es necessita un objecte VTK i com que el volum d'entrada
l'hem llegit utilitzant les llibreries ITK doncs mitjançant aquesta funció
podrem passar el volum ITK a volum VTK. */

template <typename VTK_Exporter, typename ITK_Importer>
void ConnectPipelines(VTK_Exporter* exporter, ITK_Importer importer)
{
    importer->SetUpdateInformationCallback(exporter-
>GetUpdateInformationCallback());
    importer->SetPipelineModifiedCallback(exporter-

```

```
>GetPipelineModifiedCallback();
importer->SetWholeExtentCallback (exporter->GetWholeExtentCallback());
importer->SetSpacingCallback (exporter->GetSpacingCallback());
importer->SetOriginCallback (exporter->GetOriginCallback());
importer->SetScalarTypeCallback (exporter->GetScalarTypeCallback());
importer->SetNumberOfComponentsCallback (exporter-
>GetNumberOfComponentsCallback());
importer->SetPropagateUpdateExtentCallback (exporter-
>GetPropagateUpdateExtentCallback());
importer->SetUpdateDataCallback (exporter->GetUpdateDataCallback());
importer->SetDataExtentCallback (exporter->GetDataExtentCallback());
importer->SetBufferPointerCallback (exporter->GetBufferPointerCallback());
importer->SetCallbackUserData (exporter->GetCallbackUserData());
}
```

Taula 10.2. Codi del arxiu visual.cpp

C. Annex. Eines de desenvolupament

Jaume Delgado Fernandez
Enginyeria Tècnica en Informàtica de Sistemes.

Universitat de Girona, 2006

11. Annex. Eines de Desenvolupament

En aquest PFC proposem construir una eina informàtica per la visualització i tractament d'imatges mèdiques, per obtenir un model en tres dimensions a partir d'aquestes imatges confocals que permetria un primer pas a l'hora de identificar possibles lesions de pell. Per assolir els objectius d'aquest projecte és necessari, primer de tot, identificar-se amb les eines a utilitzar que, com hem explicat abans, són les llibreries Qt per a la interfície, les ITK per el tractament d'imatges mèdiques, i les VTK per a la visualització 3D .

11.1 Entorn de Desenvolupament

Aquest PFC es desenvoluparà sota la plataforma Windows. Utilitzant el llenguatge C++ per a la implementació. Utilitzarem les llibreries Qt per implementar la interfície gràfica. A través del QDesigner, un software que permet crear la interfície d'una manera ràpida, construirem la nostra aplicació. Aquestes llibreries utilitzen un mètode de signals i slots que permet la comunicació entre els diferents objectes de la nostra interfície. Així podem enviar senyals als objectes per a que actuïn com desitgem.

Per al tractament de les imatges confocals hem escollit les llibreries ITK. Són unes llibreries per el processament, segmentació i registre d'imatges mèdiques. No tenen interfície gràfica ni serveixen per a visualitzar. Són de codi obert, i utilitzen una metodologia d'implementació basada en un pipeline de procés. Per cada operació que vulguem dur a terme crearem el pipeline corresponent, hi passarem les dades de la imatge i n'obtidrem el resultat un cop executat.

Finalment per a la visualització del volum en 3D de la pell utilitzarem les llibreries VTK que són també de codi obert. S'utilitza per processar gràfics en 3D, procés d'imatges i visualització. Suporta gran varietat d'algorismes de visualització incloent escalars, vectors, textures i mètodes volumètrics.

El fet d'utilitzar aquestes llibreries ens permet garantir la portabilitat del sistema a altres plataformes, adaptant-se a les eines dels especialistes.

Tot seguit, hi han totes aquelles eines de desenvolupament de software que hem utilitzat per a l'anàlisi, disseny i implementació d'aquest projecte. Per a cadascuna d'elles expliquem què i com és, el coneixement que en teníem i l'ús específic que n'hem fet en aquest projecte.

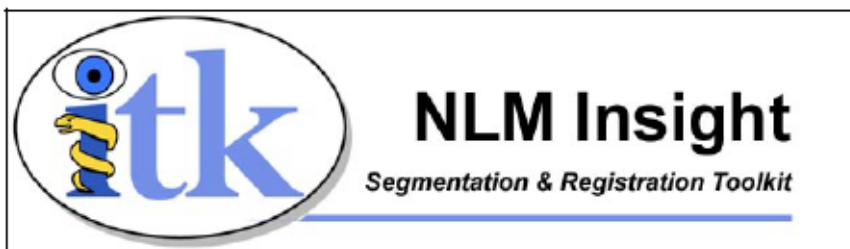
11.2 Llenguatge C++



C++ ha estat el llenguatge escollit per tal de desenvolupar l'aplicació utilitzant l'entorn Visual C++ 2005 Express gratuïta i descarregable (<http://msdn.microsoft.com/vstudio/express/visualc/download/>). A més a més també cal instal·lar la plataforma de software SDK descarregable també al mateix lloc.

La mateixa pàgina web permet seguir els passos d'instal·lació per un correcte funcionament. [Visual C++]

11.3 ITK



Les llibreries ITK (National Library of Medicine Insight Segmentation and Registration Toolkit) són unes llibreries open-source (codi obert) per tal de poder processar, segmentar i registrar imatges mèdiques.

Aquestes llibreries són utilitzades per moltíssims programadors d'arreu del món. Estan implementades en C++ seguint els principis bàsics d'orientació a objectes i a més són multiplataforma (Unix, Windows i MacOSX) i també s'hi pot treballar amb altres llenguatges com Tcl, Java o Python. Per ser open-source els programadors d'arreu del món poden usar, mantenir i ampliar aquest software.

Cal destacar que l'estil de programació de les ITKs està basada en la programació genèrica. L'essència d'aquesta programació són els contenidors per emmagatzemar dades, els iteradors per accedir-hi i els algorismes genèrics que usen els contenidors i iteradors per tal de crear algorismes fonamentals de forma eficient. Aquesta programació genèrica amb C++ està implementada amb mecanisme de programació templates i l'ús de les llibreries STK (Standard Template Library).

Els objectes de les ITKs estan estructurats seguint una estructura de data-flow. Les dades són estructurades en forma de data-objects (classes per representar dades) i aquestes dades són processades per process-objects (classes per operar sobre els data-objects). Els process-objects estan classificats en sources, filters i mappers, tot i que sovint es generalitza i s'utilitza el terme filter per a tots tres tipus. Els data-objects i els process-objects es relacionen a mitjançant pipelines. El processat de dades pipeline encadena aquests dos tipus d'objectes usant els mètodes SetInput() i GetOutput().

Abans d'utilitzar els mètodes de les classes cal definir els tipus de dades amb els quals es treballaran (imatges). Aquestes imatges vénen definides per la dimensió i el tipus de píxel (unsigned char, float,..). L'elecció del tipus de dimensió i píxel només funcionen amb alguns mètodes determinats i per tant cal anar amb compte. Tanmateix mitjançant càsting es poden canviar el tipus de dades tot i que sempre no és possible fer aquest tipus de conversions. La versió de les ITK usada per fer el projecte ha estat la 2.4.0 i no ha donat cap de problema alhora de llegir qualsevol tipus d'imatges i de tractar amb elles.

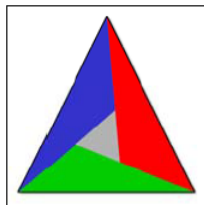
11.4 VTK



Les VTK (Visualization Toolkit) són igual que les ITKs unes llibreries open source, enfocades al processat d'imatges tant en 2D com en 3D i visualització. La implementació de les llibreries VTK ha estat també seguint els principis de l'orientació a objectes i també són multiplataforma (Unix, Windows, MacOSX). El model gràfic d'VTK té un nivell d'abstracció molt major que el d'altres llibreries de renderització d'imatges com OpenGL. Això provoca una major facilitat a l'hora d'implementar aplicacions gràfiques o de visualització.

En el cas d'aquest projecte ens interessarà visualitzar imatges i volums 3D. Per tal de poder processar la imatge primer de tot la tractarem com a un objecte ITK i tot seguit la transformarem a VTK utilitzant la classe `vtkImageExporter` (ITK) i `vtkImageImport` (VTK). Hem fet servir la versió 5.0.0 i no ens ha donat cap tipus de problema.

11.5 CMake



El CMake ens permet que s'utilitzin aquestes llibreries per qualsevol plataforma o compilador que s'utilitzi. És un sistema de codi obert que ens permetrà controlar i gestionar projectes. El CMake a més a més ens generarà els arxius de treball necessaris (el Makefile en linux/C++ o els projectes de Visual Studio). Inicialment, tant les ITKs com les VTKs s'han de compilar amb el CMake. Abans de tot, s'han de tenir instal·lades. Un cop les haguem compilat ja podrem començar a utilitzar-les. Per tal de crear el projecte necessitem el fitxer "CMakeLists.txt" que ens servirà per definir on tenim les llibreries instal·lades. Un cop ho tenim tot, només ens caldrà generar el projecte.

Hem utilitzar la versió 2.4.1 i juntament amb les versions ITK i VTK no ens ha donat problemes alhora de generar els projectes. [CMake]

11.6 Qt



Qt és una llibreria per la creació d'interfícies gràfiques. Es troba disponible en un gran nombre de plataformes: Linux, MacOS X, Solaris, HP-UX, MS-Windows XX, UNIX amb X11. És orientat a objectes, el que facilita el desenvolupament de software. És una llibreria que es basa en els conceptes de widgets (objectes), signals-slots i events (pe. click del mouse). Els signals i els slots és el mecanisme amb el que uns widgets es comuniquen amb els altres. Els widgets poden contenir qualsevol nombre de fills. En definitiva, Qt són un conjunt de llibreries multi-plataforma per el desenvolupament de l'esquelet de aplicacions GUI, escrites en codi C++. A més està completament orientada a objectes i la principal eina d'ajuda el programador, QDesigner, és de fàcil ús i intuïtiva.

Diferents entorns com KDE amb Konqueror o l'entorn de desenvolupament KDevelop, entre d'altres, utilitzen aquestes llibreries. També importants empreses han apostat per elles en els seus projectes: NASA, IBM, XEROX, Borland, etc.

A l'hora d'utilitzar aquestes llibreries hem hagut d'aprendre el funcionament dels tipus principals de widgets que utilitza (QDialog, QMainWindow, QWidget). Cada un d'ells disposa d'unes propietats específiques segons la utilitat que se'n vulgui extreure. I també del mecanisme de comunicació entre widgets. Això ha suposat un temps en l'aprenentatge.

11.7 Configuració CMake, ITK i VTK

Per tal de poder crear un nou projecte primer de tot s'ha de generar amb el CMake amb l'ajut del fitxer "CMakeLists.txt". Aquest fitxer ha de tenir la següent estructura:

```
PROJECT (nom_projecte)
FIND_PACKAGE (ITK)
IF (ITK_FOUND)
INCLUDE (${ITK_USE_FILE})
ELSE (ITK_FOUND)
MESSAGE (FATAL_ERROR "ITK not found. Please set ITK_DIR")
ENDIF (ITK_FOUND)
FIND_PACKAGE (VTK REQUIRED)
IF (NOT VTK_USE_RENDERING)
MESSAGE (FATAL_ERROR "Example ${PROJECT_NAME} requires
VTK_USE_RENDERING.")
ENDIF (NOT VTK_USE_RENDERING)
INCLUDE (${VTK_USE_FILE})
ADD_EXECUTABLE (nom_projecte nom_projecte.cxx)
TARGET_LINK_LIBRARIES (nom_projecte ITKBasicFilters ITKCommon ITKIO
vtkWidgets vtkRendering vtkGraphics vtkHybrid vtkImaging vtkIO
vtkFiltering vtkCommon)
```

La primera línia fa referència al nom del projecte que ens apareixerà al Visual Studio (sota UNIX no és aplicable). La següent línia fa referència a la configuració de les ITKs i les següent a la configuració de les VTKs. Tot seguit la comanda ADD_EXECUTABLE defineix el primer argument del nom de l'executable que serà

produït en generar l'executable. Tot seguit la comanda `TARGET_LINK_LIBRARIES` fa referència al linkatge de les llibreries que utilitzarà el projecte. Un cop tenim configurat aquest fitxer ja podem generar el projecte. Si no tenim les rutes ben configurades de les ITKs i de les VTKs les haurem d'escriure manualment al CMake i un cop les tenim entrades ja podem generar el projecte i esperar que no sorgeixi cap error de configuració. En el meu cas al utilitzar un entorn Windows em generarà el projecte que ja es pot obrir directament amb el Visual C++ sense cap tipus de problema.

D. Annex. Contingut del CD Rom

Jaume Delgado Fernandez
Enginyeria Tècnica en Informàtica de Sistemes.

Universitat de Girona, 2006

12. Annex. Contingut del CD Rom

En aquest annex explicarem el contingut del CR Rom del projecte. Separarem la informació en tantes parts com objectius ha tingut el nostre treball, tal com mostra la Figura 12.1.

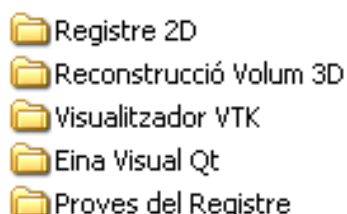


Figura 12.1 Diagrama de organització dels fitxers

Una de les carpetes que contindrà el CD serà la referent a les proves realitzades en el registre 2D, per treure les conclusions de quin era el millor mètode per registrar imatges confocals. Aquesta carpeta estarà dividida en 6 subcarpetes, una per cada mètode aplicat i les dues últimes on estan els resultats visuals de les imatges MCF i MRI del cervell, com veiem a la Figura 12.2.

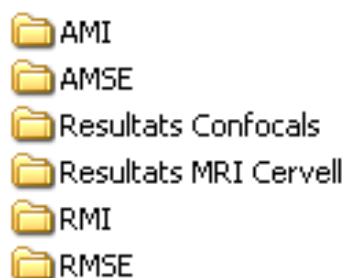


Figura 12.2

Una d'aquestes parts serà el codi del primer objectiu del meu projecte com és el registre d'imatges 2D. Posarem a la carpeta "Registre 2D" l'arxiu metodeRegRMSE.cpp juntament amb el seu fitxer per fer tota la compilació i linkatge, com és CMakeLists.txt. (Veure Figura 12.3)



Figura 12.3 Contingut de la carpeta Registre 2D.

La segona part contindrà el codi de la reconstrucció de l'isovolum tridimensional, juntament amb els arxius de text que contenen cada tall amb tota la informació.

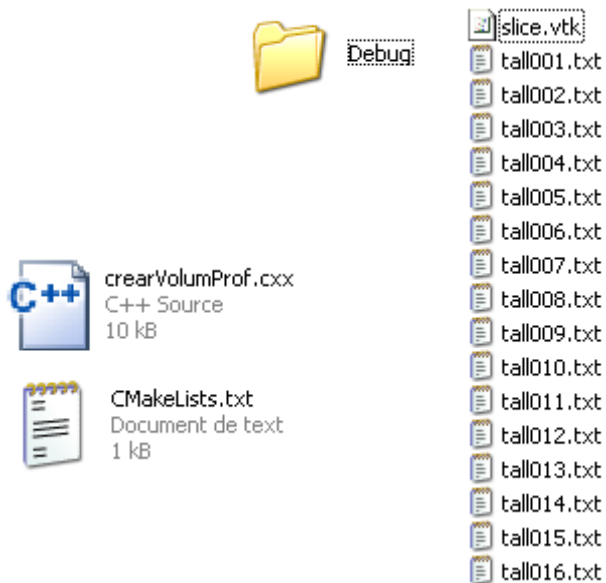


Figura 12.4. Contingut part de reconstrucció de l'isovolum.

En el CD Rom hi ha una carpeta anomenada "Visualitzador VTK" on guardem els arxius de la visualització dels volums 3D i dels talls 2D de cada eix de coordenades.



Figura 12.5. Carpeta del codi del visualitzador en vtk.

En últim terme, a la carpeta anomenada "Eina Visual Qt" posarem els arxius corresponents a la programació de la interfície visual. Aquest arxius són els provinents de l'aplicació treballada amb Qt Design. Separarem el contingut en tres carpetes. Una per la part binària "bin", una altra on estarà el codi corresponent a l'eina "src" i la darrera on hi haurà els arxius per treballar amb l'interfície, com poden ser els arxius .vtk. (Figura 12.6)



Figura 12.6. Contingut de l'Eina Visual Qt.

ÍNDIX DE FIGURES

FIGURA 1.1. ESQUEMA DEL SISTEMA CONFOCAL PER L' APLICACIÓ DE TRACTAMENT D' IMATGES.	6
FIGURA 2.1. CAPES DE LA PELL.	13
FIGURA 2.2. TIPUS DE CÈL·LULES QUE COMPOSEN LA EPIDERMIS.	14
FIGURA 2.3. IMATGES DEL CARCINOMA DE CÈL·LULES ESCAMOSOS.	15
FIGURA 2.4. IMATGES DEL CARCINOMA DE CÈL·LULES BASALS.	16
FIGURA 2.5. IMATGES DEL SARCOMA DE KAPOSI.	16
FIGURA 2.6. IMATGES DEL LIMFOMA CUTANI.	17
FIGURA 2.7. SEGUIMENT D' UN PACIENT AMB LESIONS DE MELANOMES.	18
FIGURA 2.8. IMATGE PROCEDENT D' UN DIAGNOSI DE MCF.	22
FIGURA 2.9. IMATGES PROCEDENTS D' UN DIAGNÒSTIC AMB DERMATOSCÒPIA.	22
FIGURA 2.10. EXEMPLE DE LES FASES DEL CÀNCER.	24
FIGURA 2.11. TRACTAMENT DE LA QUIMIOTERÀPIA.	25
FIGURA 2.12. TRACTAMENT DE LA RADIOTERÀPIA.	25
FIGURA 3.1. PAS DE LES IMATGES DESALINEADES OBTINGUDES AMB EL MICROSCOPI CONFOCAL ON MITJANÇANT UN REGISTRE 2D/2D DE CADA TALL OBTINDREM L' ALINEAT DELS TALLS.	28
FIGURA 3.2. EL REGISTRE D' IMATGES ÉS EL PROCÉS DE TROBAR UN MAPA DE PUNTS D' UNA IMATGE A UNA ALTRA.	28
FIGURA 3.3 AQUESTS SÓN ELS COMPONENTS BÀSICS D' UN REGISTRE. DUES IMATGES D' ENTRADA, EL TRANSFORMADOR, LA MÈTRICA, L' INTERPOLADOR I L' OPTIMITZADOR.	29
FIGURA 3.4. EXEMPLE DE CÀLCUL DE LA TAULA DE L' HISTOGRAMA PER CALCULAR L' ENTROPIA CONJUNTA DE A I B.	31
FIGURA 3.5. INTERPOLACIÓ D' UN PUNT M DE LA IMATGE MOGUDA TRANSFORMADA QUE CAU SOBRE LA IMATGE FIXA.	32
FIGURA 3.6. RECORREGUT DE L' OPTIMITZADOR BUSCANT LA TRAJECTÒRIA MÉS EFICIENT.	33
FIGURA 3.7. GRÀFIC QUE MOSTRA UN ALTRE EXEMPLE DEL FUNCIONAMENT DE L' OPTIMITZADOR DE GRADIENT DESCENDENT.	33
FIGURA 3.8. RESULTATS DE REGISTRAR IMATGES CONFOCALS.	35
FIGURA 3.9. RESULTATS DE REGISTRAR IMATGES MRI DEL CERVELL.	35
FIGURA 4.1. (C) ÉS LA IMATGE DIFERÈNCIA (A)–(B).	37
FIGURA 4.2. (C) ÉS LA IMATGE DESPRÉS D' APLICAR-HI AL CHECKBOARD COMPOSADA PER QUADRANTS D' (A) I DE (B).	38
FIGURA 4.3. TALLS CONFOCALS I MRI ORIGINALS.	40
FIGURA 4.4. IMATGES RESULTANTS DEL REGISTRE AMI.	41
FIGURA 4.5. IMATGES RESULTANTS DEL REGISTRE AMSE.	41
FIGURA 4.6. IMATGES RESULTANTS DEL REGISTRE RMI.	42
FIGURA 4.7. IMATGES RESULTANTS DEL REGISTRE RMSE.	42
FIGURA 4.8. IMATGES RESULTANTS DEL REGISTRE AMI.	43
FIGURA 4.9. IMATGES RESULTANTS DEL REGISTRE AMSE.	43
FIGURA 4.10. IMATGES RESULTANTS DEL REGISTRE RMI.	44
FIGURA 4.11. IMATGES RESULTANTS DEL REGISTRE RMSE.	44
FIGURA 4.12. RESUM DEL REGISTRE AMI AMB IMATGES CONFOCALS.	46
FIGURA 4.13. RESUM DEL REGISTRE AMI AMB IMATGES MRI DEL CERVELL.	46
FIGURA 4.14. RESUM DEL REGISTRE AMSE AMB IMATGES CONFOCALS.	47
FIGURA 4.15. RESUM DEL REGISTRE AMSE AMB IMATGES MRI DEL CERVELL.	48
FIGURA 4.16. RESUM DEL REGISTRE RMI AMB IMATGES CONFOCALS.	49
FIGURA 4.17. RESUM DEL REGISTRE RMI AMB IMATGES MRI DEL CERVELL.	50
FIGURA 4.18. RESUM DEL REGISTRE RMSE AMB IMATGES CONFOCALS.	51
FIGURA 4.19. . RESUM DEL REGISTRE RMSE AMB IMATGES MRI DEL CERVELL.	52
FIGURA 4.20. IMATGES DIFERÈNCIA ABANS I DESPRÉS DE REGISTRAR AMB REGISTRE RMI.	54
FIGURA 4.21. IMATGES CHECKBOARD ABANS I DESPRÉS DE REGISTRAR AMB REGISTRE RMI.	54
FIGURA 4.22. IMATGES CHECKBOARD ABANS I DESPRÉS DE REGISTRAR AMB REGISTRE RMI.	55
FIGURA 4.23. IMATGES CHECKBOARD REGISTRADES AMB RMI I RMSE.	55
FIGURA 4.24. GRÀFIC DE LES DADES OBTINGUDES EN EL REGISTRE UTILITZANT MÈTRICA MSE.	56
FIGURA 4.25. GRÀFIC DE LES DADES OBTINGUDES EN EL REGISTRE UTILITZANT MÈTRICA AMI.	57
FIGURA 4.26. IMATGES ORIGINALS AMB DIFERENTS PROFUNDITATS.	58
FIGURA 5.1. ESQUEMA DEL PROCÉS DE VISUALITZACIÓ.	60
FIGURA 5.2. IMATGE D' UN MICROSCOPI CONFOCAL.	61

FIGURA 5.3. IMATGE DEL FUNCIONAMENT DEL MICROSCOPI CONFOCAL	61
FIGURA 5.4. MESURA DEL MICROSCOPI CONFOCAL	62
FIGURA 5.5. EXEMPLE DE RECONSTRUCCIONS TRIDIMENSIONALS	62
FIGURA 5.6. IMATGE ORIGINAL DE LA PELL I CONTINGUT DEL FITXER DE TEXT.	63
FIGURA 5.7. PAS DE LES IMATGES DESALINEADES OBTINGUDES AMB EL MICROSCOPI CONFOCAL ON MITJANÇANT UN REGISTRE 2D/2D DE CADA TALL OBTINDREM L'ALINEAMENT DELS TALLS.	64
FIGURA 5.8. FUNCIONAMENT DEL REGISTRE AMB VARIES IMATGES DE REFERÈNCIA	65
FIGURA 5.9. FUNCIONAMENT DEL REGISTRE AMB UNA IMATGE DE REFERÈNCIA	66
FIGURA 5.10. FUNCIONAMENT DEL REGISTRE MIXTE	67
FIGURA 5.11. SEQÜÈNCIA DE RECONSTRUCCIÓ DE L'ISOVOLUM INTRODUIINT TALL PER TALL REGISTRAT. .	68
FIGURA 5.12. EXEMPLES DE ISOVOLUM APLICANT METODOLOGIA MIXTA	68
FIGURA 6.1. DISSENY DE LA INTERFÍCIE VISUAL	71
FIGURA 6.2. DIAGRAMA DE CLASSES PER OBRIR EL VOLUM.	71
FIGURA 6.3. DIAGRAMA FUNCIONAL PER OBRIR EL VOLUM.....	72
FIGURA 6.4. DIAGRAMA DE CLASSES DE LA VISUALITZACIÓ I INTERACCIÓ.	72
FIGURA 6.5. DIAGRAMA FUNCIONAL DE LA VISUALITZACIÓ.....	73
FIGURA 6.6. DIAGRAMA DE CLASSES EN LA SELECCIÓ DE PLANS.....	73
FIGURA 6.7. DIAGRAMA FUNCIONAL PER SELECCIONAR UN PLA.	73
FIGURA 6.8. EXEMPLE DE COM ES VEU EL PLA SELECCIONAT EN EL VOLUM 3D.	74
FIGURA 6.9. BOTONS PREPARATS PER TREBALLAR AMB LA PROFUNDITAT DELS TALLS.....	74
FIGURA 6.10. DIAGRAMA DE CLASSES PER ESCOLLIR PROFUNDITAT D'UN TALL.	75
FIGURA 6.11. COMANDAMENTS PER CANVIAR LES PROPIETATS DE LA PANTALLA.....	75
FIGURA 6.12. VOLUM CARREGAT I RENDERITZAT.	76
FIGURA 6.13. VISUALITZACIÓ DEL TALL X AMB UNA PROFUNDITAT ESCOLLIDA PER L'USUARI	76
FIGURA 6.14. EXEMPLE DE VISUALITZACIÓ DEL TALL X AMB PROFUNDITAT A LA MEITAT DEL VOLUM.	77
FIGURA 9.1. DIAGRAMA DE CLASSES UTILITZADES A L'ALGORITME DE REGISTRE.	84
FIGURA 9.2. ESQUEMA DEL PROCES SEGUIT DURANT LA TASCA DE REGISTRE.	85
FIGURA 9.3. DIAGRAMA DE CLASSES UTILITZADES EN EL REGISTRE AMI.	86
FIGURA 9.4. DIAGRAMA DE CLASSES UTILITZADES EN EL REGISTRE AMI.	87
FIGURA 9.5. DIAGRAMA DE CLASSES UTILITZANT EL REGISTRE AMB TRANSFORMADOR RÍGID I INFORMACIÓ MÚTUA.	88
FIGURA 9.6. DIAGRAMA DE CLASSES UTILITZANT EL REGISTRE RMSE.	89
FIGURA 9.7. DIAGRAMA DE CLASSES PER CALCULAR LA DIFERÈNCIA ENTRE IMATGES.	90
FIGURA 9.8. DIAGRAMA DE CLASSES PER CALCULAR EL CHECKBOARD ENTRE DUES IMATGES.....	90
FIGURA 9.9. DIAGRAMA DE CLASSES PER L'ADQUISICIÓ DE LES DADES	91
FIGURA 9.10. DIAGRAMA DE CLASSES EN LA CONSTRUCCIÓ DEL VOLUM.....	92
FIGURA 9.11. DIAGRAMA DE CLASSES VTK PER LA VISUALITZACIÓ.	93
FIGURA 9.12. COMPONENTS BÀSICS D'UN REGISTRE. DUES IMATGES D'ENTRADA, EL TRANSFORMADOR, LA MÈTRICA, L'INTERPOLADOR I L'OPTIMITZADOR	94
FIGURA 12.1. DIAGRAMA DE ORGANITZACIÓ DELS FITXERS	124
FIGURA 12.2.....	124
FIGURA 12.3. CONTINGUT DE LA CARPETA REGISTRE 2D.	124
FIGURA 12.4. CONTINGUT PART DE RECONSTRUCCIÓ DE L'ISOVOLUM.....	125
FIGURA 12.5. CARPETA DEL CODI DEL VISUALITZADOR EN VTK.....	125
FIGURA 12.6. CONTINGUT DE L'EINA VISUAL QT.	125

ÍNDIX TAULES

TAULA 4.1. TAULA RESUM DELS RESULTATS MÈTRICS ABANS I DESPRÉS D'ALINEAR AMB REGISTRE AMI .	46
TAULA 4.2. TAULA RESUM DELS RESULTATS MÈTRICS ABANS I DESPRÉS D'ALINEAR AMB REGISTRE AMI .	47
TAULA 4.3. TAULA RESUM DELS RESULTATS MÈTRICS ABANS I DESPRÉS D'ALINEAR AMB REGISTRE AMSE	48
TAULA 4.4. TAULA RESUM DELS RESULTATS MÈTRICS ABANS I DESPRÉS D'ALINEAR AMB REGISTRE AMSE	49
TAULA 4.5. TAULA RESUM DELS RESULTATS MÈTRICS ABANS I DESPRÉS D'ALINEAR AMB REGISTRE RMI..	50
TAULA 4.6. TAULA RESUM DELS RESULTATS MÈTRICS ABANS I DESPRÉS D'ALINEAR AMB REGISTRE RMI..	50
TAULA 4.7. TAULA RESUM DELS RESULTATS MÈTRICS ABANS I DESPRÉS D'ALINEAR AMB REGISTRE RMSE	51
TAULA 4.8. TAULA RESUM DELS RESULTATS MÈTRICS ABANS I DESPRÉS D'ALINEAR AMB REGISTRE RMSE	52
TAULA 4.9. TAULA RESUM DELS RESULTATS MÈTRICS ABANS D'ALINEAR	53
TAULA 4.10. TAULA RESUM DELS RESULTATS MÈTRICS DESPRÉS D'ALINEAR	53
TAULA 5.1. TAULA DESCRIPTIVA DEL CONTINGUT DEL FITXER DE TEXT.	63
TAULA 10.1. CODI PER EL DISSENY DE L'EINA VISUAL.	106
TAULA 10.2. CODI DEL ARXIU VISUAL.CPP.....	116