

# Índex

<b>Índex</b> .....	<b>1</b>
<b>Índex de figures</b> .....	<b>3</b>
<b>Capítol 1</b>	
<b>Introducció</b> .....	<b>7</b>
1.1 Descripció General .....	7
1.2 Objectius .....	8
1.3 Abast i sortides del projecte .....	10
1.4 Planificació .....	11
1.5 Estructura del document .....	12
<b>Capítol 2</b>	
<b>Introducció a la visió per computador</b> .....	<b>17</b>
2.1 Utilitzacions de la visió per computador .....	17
2.2 Reconstrucció d'una imatge .....	19
2.2.1 Adquisició i representació d'una imatge .....	19
2.2.2 Projecció al pla imatge .....	22
2.2.3 Càmera: Model Pinhole .....	23
2.2.4 Projecció ortogràfica .....	25
2.3 Reconstrucció 3D .....	27
<b>Capítol 3</b>	
<b>Reconstrucció gran a partir de vistes 3D parcials</b> .....	<b>30</b>
3.1 Mètodes de factorització .....	30
3.1.1 Introducció .....	30
3.1.2 Factorization Method .....	31
3.1.3 Power factorization .....	37
3.2 Iterative Closest Point (ICP) .....	41
3.3 Mètode de Reconstruccions 3D grans a partir de seqüències d'imatges 2D .....	44

3.3.1 Introducció .....	44
3.3.2 Recuperació de la forma a partir de poques imatges .....	45
3.3.3 Registre i recuperació a partir de l'increment de vistes 3D .....	47
<b>Capítol 4</b>	
<b>Resultats .....</b>	<b>50</b>
4.1 Introducció .....	50
4.2 Experiments sintètics .....	51
4.2.1 Cub .....	51
4.2.2 Cub amb punts aleatoris .....	56
4.3 Experiments reals .....	61
4.3.1 Casa .....	61
4.3.2 Màscara .....	67
4.4 Conclusions .....	74
<b>Capítol 5</b>	
<b>Conclusions i treballs futurs .....</b>	<b>75</b>
5.1 Conclusions .....	75
5.2 Treballs futurs .....	76
5.3 Aplicacions .....	78
<b>Bibliografia.....</b>	<b>80</b>
<b>Annex</b>	
<b>Toolbox MatLab: Manual d'usuari.....</b>	<b>82</b>
A.1 Estructura .....	82
A.2 Funcions .....	83

## Índex de figures

1.1	Esquema general del projecte .....	9
1.2	Pantalla principal de l'entorn MatLab utilitzat en aquest projecte. ....	13
1.3	Calendari de la planificació .....	16
2.1	Exemple de digitalització espacial. La primera foto és la que menys píxels per centímetre té, i a mesura que anem augmentant aquesta relació, es pot comprovar com la imatge es veu més clara i definida.....	20
2.2	Exemple de digitalització en amplitud. La primera imatge conté 2 bits per píxel, per tant, només és blanc o negre, i la última, conté més bits per píxel i s'hi poden20	
2.3	Exemple de digitalització en color. Les tres primeres matrius contenen la intensitat de cada color i al final es sumen les matrius per formar la imatge final en color...21	
2.4	Exemple d'una imatge amb soroll gaussià. ....	22
2.5	Model de projecció de la imatge .....	23
2.6	Model d'una càmera Pinhole. ....	25
2.7	Exemple d'una imatge amb projecció ortogràfica .....	26
2.8	Esquema de la projecció ortogràfica. ....	26
4.1	Representació de la seqüència 2D d'un cub simple. Frames 1,20,40,70.....	52
4.2	Primera reconstrucció obtinguda de la seqüència del cub utilitzant el Factorization Method. Cada figura representa una vista diferent del cub. ....	53
4.3	Desena reconstrucció obtinguda de la seqüència del cub utilitzant el Factorization Method. Cada figura representa una vista diferent del cub. ....	53
4.4	Última reconstrucció obtinguda de la seqüència del cub utilitzant el Factorization Method. És la reconstrucció final. Cada figura representa una vista diferent del cub. ....	53
4.5	Primera reconstrucció parcial obtinguda de la seqüència del cub utilitzant el Power Factorization. Cada figura representa una vista diferent del cub. ....	55
4.6	Desena reconstrucció parcial obtinguda de la seqüència del cub utilitzant el Power	

Factorization. Cada figura representa una visa diferent del cub.....	55
4.7 Última reconstrucció obtinguda de la seqüència del cub utilitzant el Power Factorization. És la reconstrucció final. Cada figura representa una vista diferent del cub.....	55
4.8 Representació de la seqüència 2D d'un cub simple amb 30 punts aleatoris dins seu. Aquest figura correspon als frames 1,30,60,90,120.....	56
4.9 Primera reconstrucció obtinguda de la seqüència d'un cub amb 30 punts aleatoris dins seu utilitzant el Factorization Method.. Cada imatge representa una vista diferent del cub.....	58
4.10 Vintena reconstrucció obtinguda de la seqüència d'un cub amb 30 punts aleatoris dins seu utilitzant el Factorization Method. Cada imatge representa una vista diferent del cub.....	58
4.11 Última reconstrucció obtinguda de la seqüència d'un cub amb 30 punts aleatoris dins seu utilitzant el Factorization Method. Cada imatge representa una vista diferent del cub.....	58
4.12 Primera reconstrucció parcial obtinguda de la seqüència d'un cub amb 30 punts aleatoris dins utilitzant el Power Factorization. Cada imatge representa una vista diferent del cub.....	60
4.13 Vintena reconstrucció parcial obtinguda de la seqüència d'un cub amb 30 punts aleatoris dins seu utilitzant el Power Factorization. Cada imatge representa una vista diferent del cub. ....	60
4.14 Última reconstrucció obtinguda de la seqüència d'un cub amb 30 punts aleatoris dins seu utilitzant el Power Factorization. Cada imatge representa una vista diferent del cub.....	60
4.15 Model original de la casa. Frames 1,17,35 i 50. ....	62
4.16 Seqüència de la casa generada sintèticament a partir del model 3D. Frames 1,17,35,50. ....	63
4.17 Primera reconstrucció obtinguda de la seqüència de la casa utilitzant el Factorization Method. La primera vista representa el perfil, la segona l'alçat i la	

tercera la planta. ....	64
4.18 Novena reconstrucció obtinguda de la seqüència de la casa utilitzant el Factorization Method. La primera vista representa el perfil, la segona l'alçat i la tercera la planta. ....	64
4.19 Última reconstrucció obtinguda de la seqüència de la casa utilitzant el Factorization Method. La primera vista representa el perfil, la segona l'alçat i la tercera la planta. ....	64
4.20 Primera reconstrucció obtinguda de la seqüència de la casa utilitzant el Power Factorization. La primera vista representa el perfil, la segona l'alçat i la tercera la planta. ....	66
4.21 Setena reconstrucció obtinguda de la seqüència de la casa utilitzant el Power Factorization. La primera vista representa el perfil, la segona l'alçat i la tercera la planta. ....	66
4.22 Última reconstrucció obtinguda de la seqüència de la casa utilitzant el Power Factorization. La primera vista representa el perfil, la segona l'alçat i la tercera la planta. ....	66
4.23 Mostra d'algun dels frames de la seqüència de la màscara per poder veure el moviment de la càmera. ....	68
4.24 Mostra dels punts agafats al llarg de la seqüència. Es pot veure que a mesura que van passant els frames, es van perdent molts punts. ....	69
4.25 Reconstruccions obtingudes utilitzant només SfM. La de l'esquerra s'ha obtingut utilitzant el Factorization Method i la de la dreta utilitzant el Power Factorization. ....	70
4.26 Imatge frontal de la màscara. ....	72
4.27 Una de les primeres reconstruccions obtingudes de la seqüència de la màscara. La primera vista representa la part frontal, la segona el perfil i la tercera la part superior. ....	73
4.28 Reconstrucció intermitja obtinguda de la seqüència de la màscara. La primera vista representa la part frontal, la segona el perfil i la tercera la part superior. ....	73

4.29 Última reconstrucció obtinguda de la seqüència de la màscara. La..... 73

# Capítol 1

## Introducció

### 1.1 Descripció General

Aquest projecte està emmarcat dins el grup de Visió per Computador i Robòtica del Departament d'Arquitectura i Tecnologia de Computadors de la Universitat de Girona. Pertany a l'àmbit de la Visió per Computador i, concretament, al tòpic de la recerca que estudia la reconstrucció 3D d'objectes rígids. Durant els últims anys, el problema de com trobar la forma i el moviment d'un objecte a través d'una seqüència d'imatges ha tingut una gran importància en el camp de la visió per computador. Aquest projecte es basarà en reconstruir una imatge 3D gran a partir d'una seqüència d'imatges 2D capturades per una càmera.

Ens centrem en l'estudi de les bases matemàtiques de la visió per computador així com en diferents mètodes emprats en la reconstrucció 3D d'imatges. Per portar a terme aquest estudi s'utilitza la plataforma de desenvolupament MatLab ja que permet tractar operacions matemàtiques, imatges i matrius de gran tamany amb molta senzillesa, rapidesa i eficiència, per aquesta raó s'usa en moltes recerques sobre aquest tema.

El projecte aprofundeix en el tema descrit anteriorment estudiant i implementant un mètode que consisteix en aplicar *Structure From Motion (SfM)* [2] a pocs frames seguits obtinguts d'una seqüència d'imatges 2D per crear una reconstrucció 3D. Quan s'han creat dues reconstruccions 3D consecutives i fent servir un frame com a mínim en comú entre elles, s'aplica un mètode de registre d'estructures 3D, l'*Iterative Closest Point (ICP)*[6], per crear una reconstrucció 3D més gran a través d'unir les diferents reconstruccions obtingudes a partir de *SfM*. El mètode consisteix en anar repetint aquestes operacions fins al final dels frames per poder aconseguir una reconstrucció 3D més gran que les petites imatges que s'aconsegueixen a través de *SfM*. A la Figura 1.1 es pot veure un esquema del procés que es segueix. Per avaluar el comportament del mètode, utilitzem un conjunt de seqüències sintètiques i un conjunt de seqüències reals obtingudes a partir d'una càmera.

L'objectiu final d'aquest projecte és construir una nova *toolbox* de MatLab amb tots els mètodes per crear reconstruccions 3D grans per tal que sigui possible tractar amb facilitat aquest problema i seguir-lo desenvolupant en un futur.

## 1.2 Objectius

- Aprendre els fonaments dels models de projecció de les càmeres.
- Estudi dels diferents mètodes de reconstrucció 3D basats en *Structure From Motion (SfM)*. En aquest apartat s'analitza el mètode de factorització, s'estudien i s'implementen els mètodes *Factorization Method* i *Power Factorization*.
- Estudi de mètodes de Registre d'estructures 3D.



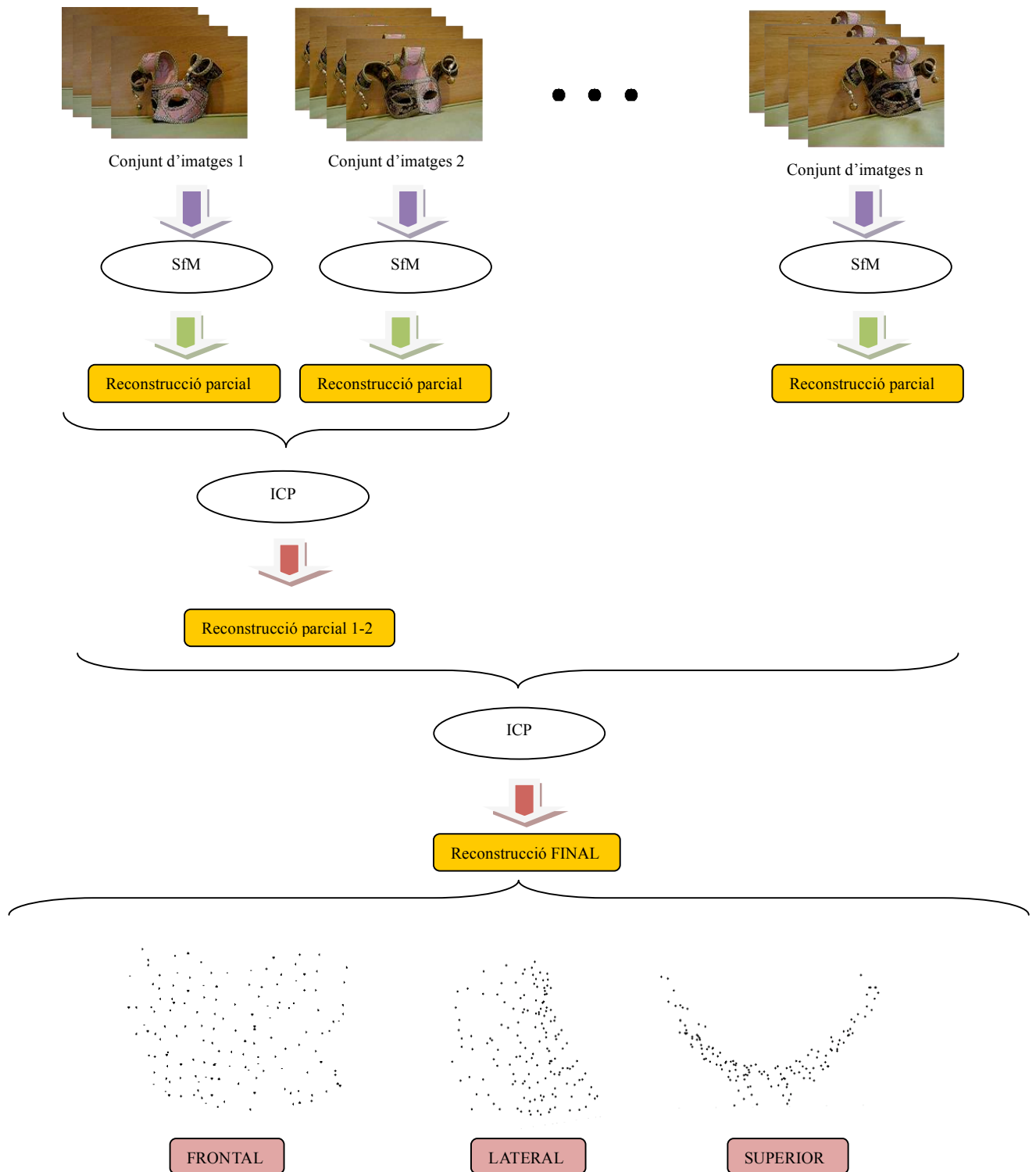


Figura 1.1 Esquema general del projecte

- Disseny d'un nou mètode que combini *Structure For Motion* i registre d'estructures 3D, per tal d'obtenir grans reconstruccions d'imatges 3D a partir de seqüències 2D. Amb els mètodes de *SfM* això no és possible degut al fet que si la seqüència d'imatges 2D obtinguda és molt llarga, és a dir, conté molts frames, és possible que al final apareguin punts que no existien al primer frame, i això fa que el mètode no sigui capaç d'obtenir una reconstrucció 3D.
- Creació de dades sintètiques i reals pels experiments i posterior anàlisi dels resultats. Es dissenyen experiments amb imatges sintètiques i es prova el mètode, posteriorment es dissenyen experiments reals amb imatges capturades amb una càmera. Com que són molt més imprecisos que els experiments sintètics, podem veure quina utilitat pot tenir el mètode a la vida real, i també es comparen els resultats obtinguts sintèticament i els reals per tal de decidir si el mètode és vàlid o no. Per tal d'obtenir els punts característics s'ha après a fer el procés de seguiment d'aquests a través del tracking.

### 1.3 Abast i sortides del projecte

A part de fer un treball de recerca i d'assolir els coneixements que ens hem proposat, aquest projecte vol aconseguir els següents punts:

- Construir una toolbox de MatLab amb la qual es pugui treballar amb facilitat el problema de la reconstrucció 3D d'objectes de gran tamany a partir d'imatges de seqüències 2D i reconstruccions parcials 3D.
- Recopilar un conjunt de seqüències d'exemples on es pugui aplicar el mètode. Hi haurà exemples generats sintèticament i d'altres de reals.
- Generar una àmplia i clara documentació.

## 1.4 Planificació

El desenvolupament d'aquest projecte es divideix en les següents fases:

- **Especificacions generals dels temes a tractar, aclariments generals sobre el projecte:** Fase en la qual es miren quins seran els temes a tractar i quin és l'objectiu principal a assolir.
- **Presa de contacte amb l'entorn MatLab:**  
Abans de començar a implementar els diferents mètodes del projecte, ha estat necessari aprofundir en les funcions bàsiques del programa. (Veure Figura 1.2)
- **Implementació de les funcions bàsiques:**  
Per començar s'han implementat algunes funcions bàsiques, com per exemple, la funció que crea un seguit d'imatges sintètiques d'algun objecte 3D en moviment.
- **Implementació del Factorization Method i Power Factorization per a objectes rígids:** S'han implementat aquests mètodes pertanyents al *SfM* prèviament analitzats en el projecte final de carrera de Ricard Campos[5]. Aquest mètodes són la base per poder fer reconstruccions completes d'un objecte.
- **Estudi del mètode Iterative Closest Points (ICP):**  
S'ha estudiat la implementació d'aquest mètode per utilitzar-lo juntament amb *Structure For Motion* i crear un nou mètode que permeti una reconstrucció 3D d'objectes grans molt més fiable i eficaç que les anteriors.
- **Implementació d'un nou mètode de reconstrucció:**  
S'ha implementat un nou mètode utilitzant *Structure For Motion* i registre

d'estructures 3D.

- **Realització de proves amb vídeos reals i interpretació i anàlisi dels resultats:**

Es crearan situacions controlades en les que es podran capturar un seguit d'imatges d'un mateix objecte des de diferents punts de vista i obtenir-ne els punts característics. Un cop realitzades les seqüències i reconstruccions a partir de les imatges reals obtingudes, les compararem amb els resultats ideals i s'obtiniran conclusions sobre si el mètode és eficaç o no.

El calendari de la Figura 1.3 representa la planificació del desenvolupament del projecte.

## 1.5 Estructura del document

Aquest document consta de diferents capítols. A continuació hi ha una breu descripció de cadascun d'ells:

- **Capítol 2: Introducció a la visió per computador.**

En aquest capítol s'expliquen els conceptes que apareixen al llarg del document sobre el camp de visió per computador. Es parla de la reconstrucció 3D, de com es formen les imatges, com es fa la projecció al pla imatge, la diferència entre projecció ortogonal i projecció projectiva, etc.

- **Capítol 3: Mètode de Reconstruccions 3D grans a partir de seqüències 2D.**

En aquest capítol ens centrarem en el funcionament d'aquests dos mètodes i de les diferències que tenen. Es donaran mostres dels resultats obtinguts amb els dos mètodes.

També veurem com funciona el *Iterative Closest Point*. Ens fixarem en com tracta les matrius de mesures i com troba els punt comuns entre elles mirant quins d'ells tenen una distància mínima per així relacionar-los. També parlarem dels avantatges i els inconvenients que té aquest mètode.

Finalment estudiarem com unir els mètodes de *Structure For Motion* i *Iterative Closest Points* per poder generar una millor i completa vista de la reconstrucció 3D, i parlarem dels avantatges i els inconvenients que té aquest mètode.

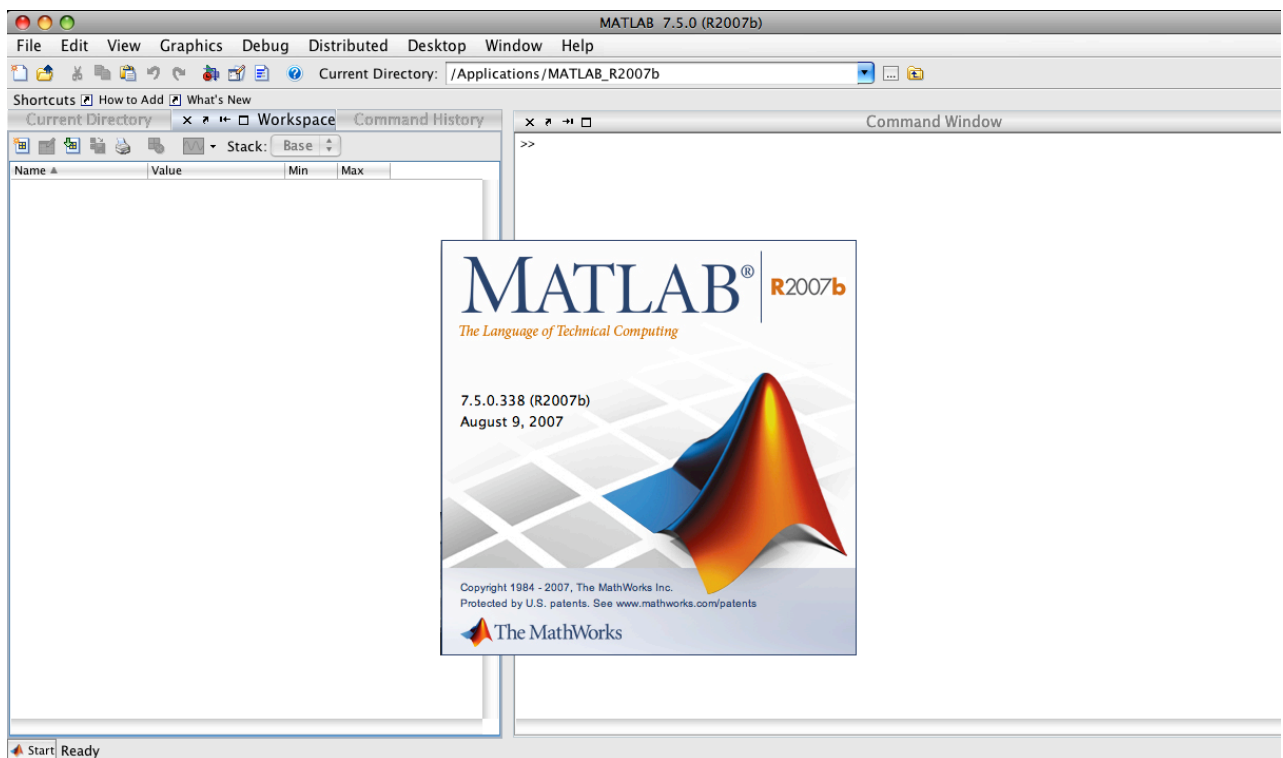


Figura 1.2 Pantalla principal de l'entorn MatLab utilitzat en aquest projecte.

- **Capítol 4: Resultats.**

En aquest capítol ens basarem en els resultats. Explicarem com hem creat tant

les seqüències sintètiques com les reals. Descriurem quin comportament té el mètode anterior quant a l'eficiència i utilitat.

- **Capítol 5: Conclusions i treballs futurs.**

Aquí valorarem l'abast d'aquest treball i la seva utilitat dins de diversos camps. A més a més, també inclourem una relació de futures ampliacions que aquest projecte podria tenir.

- **Annex.**

Inclourem tota la informació relacionada amb la toolbox de MatLab que s'ha desenvolupat durant aquest projecte. Presentarem les funcions i explicarem com fer ús de cadascuna d'elles.

OCTUBRE						
Dll	Dm	Dc	Dj	Dv	Ds	Dg
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				
NOVEMBRE						
Dll	Dm	Dc	Dj	Dv	Ds	Dg
			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		
DESEMBRE						
Dll	Dm	Dc	Dj	Dv	Ds	Dg
					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						
MARÇ						
Dll	Dm	Dc	Dj	Dv	Ds	Dg
					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

Segona quinzena: Definir el projecte

Presca de Contacte amb l'entorn MatLab

Anàlisi i implementació dels mètodes de SfM

Anàlisi i implementació dels mètodes de SfM

Búsqueda de documentació de ICP

31							
ABRIL							
Dll	Dm	Dc	Dj	Dv	Ds	Dg	
	1	2	3	4	5	6	Anàlisi ICP i implementació de SfM + ICP
7	8	9	10	11	12	13	
14	15	16	17	18	19	20	
21	22	23	24	25	26	27	
28	29	30					
MAIG							
Dll	Dm	Dc	Dj	Dv	Ds	Dg	
			1	2	3	4	Anàlisi de diferents seqüències d'imatges
5	6	7	8	9	10	11	aplicant SfM + ICP
12	13	14	15	16	17	18	Documentació
19	20	21	22	23	24	25	
26	27	28	29	30	31		
JUNY							
Dll	Dm	Dc	Dj	Dv	Ds	Dg	
						1	Anàlisi de diferents seqüències d'imatges
2	3	4	5	6	7	8	aplicant SfM + ICP
9	10	11	12	13	14	15	Documentació
16	17	18	19	20	21	22	
23	24	25	26	27	28	29	
30							
JULIOL							
Dll	Dm	Dc	Dj	Dv	Ds	Dg	
	1	2	3	4	5	6	Anàlisi de diferents seqüències d'imatges
7	8	9	10	11	12	13	aplicant SfM + ICP
14	15	16	17	18	19	20	Documentació
21	22	23	24	25	26	27	Preparació de la presentació
28	29	30	31				

Figura 1.3 Calendari de la planificació



## Capítol 2

### Introducció a la visió per computador

#### 2.1 Utilitzacions de la visió per computador

La visió per computador és un camp de recerca actual. Aquest tòpic estudia com aplicar les funcions de la visió humana al camp de la informàtica. Per aconseguir-ho utilitza una càmera que fa la funció d'ull humà recollint les imatges i d'un processador que simula el cervell, on s'analitzen les imatges per extreure'n informació. Els algoritmes solucionen el problema del funcionament de l'ull humà, una eina molt precisa, però que costa molt imitar el seu comportament, ja que primer cal saber com reacciona el cervell quan rep totes les imatges.

Els camps on més s'aplica la visió per computador són:

- Control de qualitat: en aquest camp és molt útil, ja que permet analitzar tots els productes i trobar-hi tota mena de desperfectes per petits que siguin, fins i tot si no ho podem veure a simple vista, fent que el producte resultant tingui una qualitat més elevada. A més a més, cal tenir en compte que un processador dedicat a aquesta tasca treballa a un ritme molt més elevat que si ho analitzés una persona.

- Robòtica: pels robots és molt important adquirir dades del seu entorn, especialment dades visuals per, d'aquesta manera, poder esquivar possibles objectes que hi hagi en la seva trajectòria i també per si s'han de fer una anàlisi.
- Seguretat: en aquest cas serveix per tenir vigilats els edificis, per la detecció d'explosius a través dels rajos X, entre altres.
- Biomedicina: en aquest camp permet analitzar les imatges obtingudes amb ultrasons i rajos X i d'aquesta manera obtenir un diagnòstic més acurat, i també per fer un control en temps real en les intervencions on hi és necessària una càmera.
- Identificació: en aquest cas ens permet reconèixer les faccions del rostre d'una persona i fer-ne un seguiment. També permet fer altres tipus d'identificacions, com ara reconèixer les empremtes dactilars o diversos objectes dins d'una seqüència d'imatges.
- Militar: en aquesta matèria serveix per fer reconeixements del terreny, seguiment d'objectius, anàlisi d'armes, etc., fent que sigui menys arriscat per als soldats.
- Entreteniment: degut al boom de la multimèdia, les aplicacions que utilitzen la visió per computador cada vegada són més nombroses i abracen més àmbits dins de l'entreteniment.

Tal com es pot comprovar, la visió per computador s'utilitza molt i fa que algunes tasques no resultin tant feixugues o fins i tot, que algunes feines que abans no es podien fer ara siguin possibles. En aquest projecte només ens centrem però, en el camp de la reconstrucció 3D que és aplicable al camp de l'entreteniment, de la robòtica i fins i tot de la identificació.

## 2.2 Reconstrucció d'una imatge

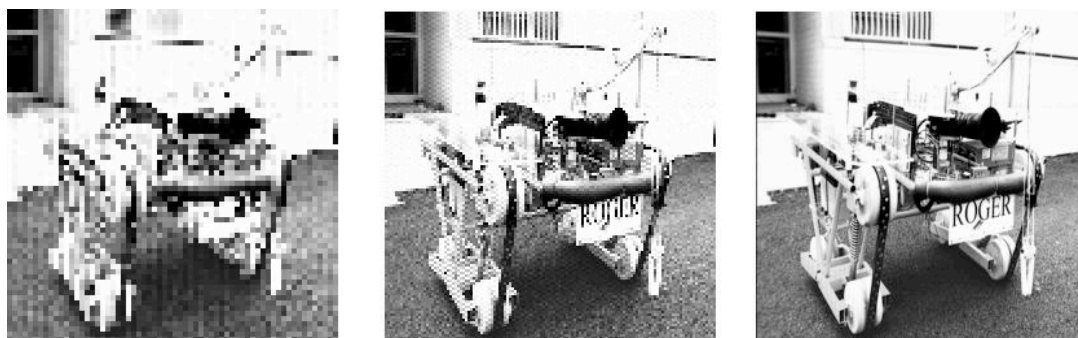
Per començar a parlar de com es fa una reconstrucció 3D a partir d'imatges 2D, primerament cal saber com es formen aquestes imatges en el pla, per tant, començarem parlant de les bases geomètriques que fan que es formi una imatge 2D d'una escena real.

### 2.2.1 Adquisició i representació d'una imatge

Primerament cal definir què és una imatge, ja que es tracta un dels termes més importants en aquest projecte. Una imatge és una funció bidimensional  $f(x,y)$ , on el valor d'aquesta funció a la coordenada  $(x,y)$  del pla  $x$ - $y$  és una mesura de la intensitat de llum d'aquest punt.

Per tractar les imatges amb ordinador, ens interessen només les imatges digitals, que són una aproximació de la funció  $f(x,y)$ . Aquesta aproximació es pot fer de dues formes:

- *Digitalització espacial*: consisteix en representar la funció de la imatge contínua original en una matriu de punts discrets anomenats píxels. En aquest cas, com més resolució més píxels per centímetre hi ha. (Veure Figura 2.1).



*Figura 2.1 Exemple de digitalització espacial. La primera foto és la que menys píxels per centímetre té, i a mesura que anem augmentant aquesta relació, es pot comprovar com la imatges es veu més clara i definida.*

- *Digitalització “en amplitud”*: cada píxel de la matriu ha de codificar la intensitat local de la imatge en aquest punt. Això s’aproxima assignant-li un valor d’entre els nivells possibles. En aquest cas, la imatge queda codificada en escala de grisos, si cada píxel només està format per un sol bit, la imatge obtinguda és una imatge binària, ja que només hi haurà blanc i negre, en canvi, com més píxels hi hagi, més escala de grisos tindrem. (Veure Figura 2.2).



*Figura 2.2 Exemple de digitalització en amplitud. La primera imatge conté 2 bits per píxel, per tant, només és blanc o negre, i la última, conté més bits per píxel i s’hi poden*

*veure més matisos i la imatge és més nítida.*

Per crear una imatge en color, s'utilitza la mateixa tècnica, però en lloc de tenir una sola matriu en tenim 3, una per a cada color bàsic, i cada píxel està format pel triple de bits. (Veure Figura 2.3).



*Figura 2.3 Exemple de digitalització en color. Les tres primeres matrius contenen la intensitat de cada color i al final es sumen les matrius per formar la imatge final en color.*

Un altre concepte a tenir en compte és el soroll. El soroll és la degradació de la imatge que es pot produir durant la seva captura, transmissió o processament. Quan es crea una imatge sintètica, està exempta de soroll i per tal de simular una imatge real se li aplica algun tipus de soroll. El tipus més emprat en visió per computador és el gaussià, que tal i com el seu nom indica, segueix una distribució gaussiana. (Veure Figura 2.4).



Imatge original

Imatge original amb  
20% de soroll gaussià*Figura 2.4 Exemple d'una imatge amb soroll gaussià.*

### 2.2.2 Projectió al pla imatge

La projecció d'un objecte és la forma que s'obté al dirigir totes les línies projectants des d'aquest objecte fins a un pla.

Tal i com es pot veure a la figura, els elements importants de la projecció d'una imatge són el punt que es vol projectar (P), el punt projectat (P'), la línia projectant (VA') i el pla sobre el qual es projecta ( $\Pi$ ), també anomenat pla imatge. (Veure Figura 2.5).

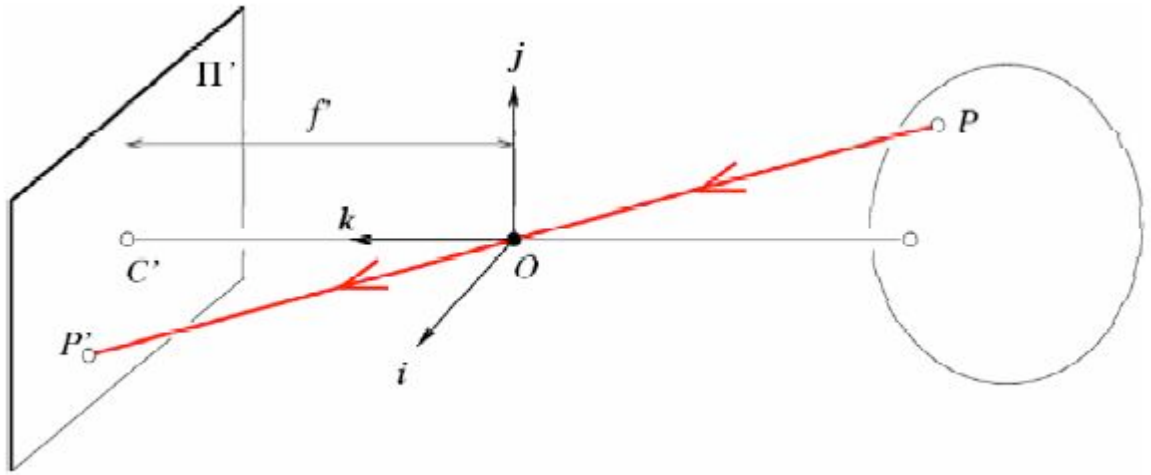


Figura 2.5 Model de projecció de la imatge

L'ull humà està acostumat a veure amb aquest tipus de projecció, anomenada perspectiva, on els objectes llunyans semblen més petits, i totes les línies perpendiculars a nosaltres convergeixen en un punt a l'infinit, anomenat focus de projecció.

La projecció perspectiva és la que es considera el model general de projecció, ja que és el model de l'ull humà, però quan es passa una imatge 3D al pla 2D es crea soroll, anomenat soroll perspectiu, que és causat per la perspectiva i la imatge resultant queda deformada.

### 2.2.3 Càmera: Model Pinhole

El model Pinhole de càmera descriu les relacions matemàtiques entre les coordenades

d'un punt 3D i la seva projecció al pla imatge per a una càmera Pinhole ideal, on l'obertura de la càmera és un punt i no s'utilitzen lents per enfocar. Cal tenir en compte que com que és un model ideal, no té en compte les distorsions geomètriques que hi pugui haver ni tampoc les imatges desenfocades. (Veure Figura 2.6).

El sistema de càmera Pinhole per crear imatges 2D és molt senzill, i és la base de totes les càmeres. En aquest cas, es fa un petit forat en una caixa per poder-hi deixar entrar la llum, i en conseqüència la imatge. La caixa ha d'estar ben tancada i fosca perquè la llum de fora no deformi la imatge que volem reproduir. La imatge es projecta sobre un material sensible a la llum i hi queda gravada. Si es fa la prova, es pot comprovar com els objectes més allunyats són més petits i els propers són més grans.

Els punts en la imatge en 3D es representen amb les coordenades  $(x,y,z)$ , i per passar d'un punt 3D a un punt 2D, hi ha una fórmula molt senzilla, els punt del pla 2D els anomenarem  $(x',y')$ :

$$x' = x/f \quad y' = y/f \tag{2.1}$$

on  $f$  és la distància que hi ha entre l'objecte i el pla on queda dibuixada la imatge 2D.



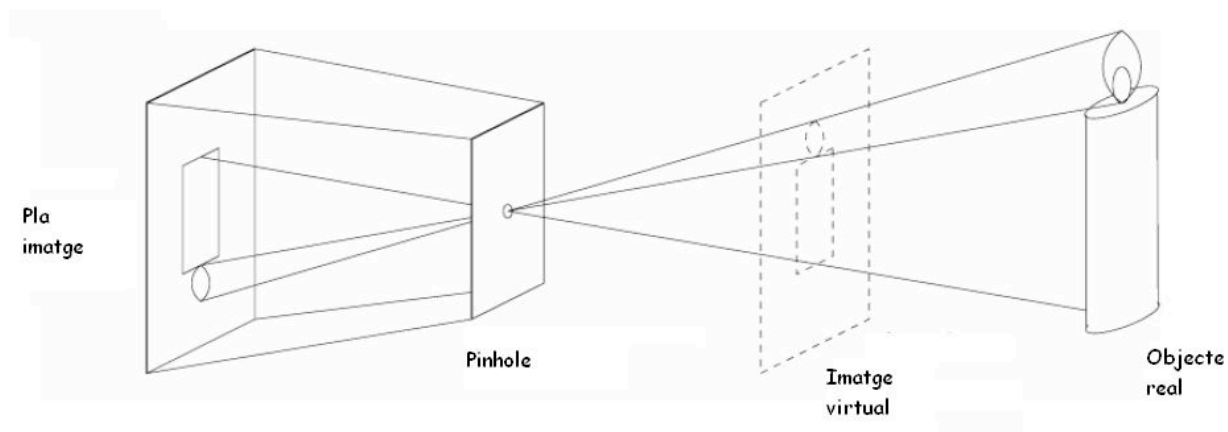


Figura 2.6 Model d'una càmera Pinhole.

## 2.2.4 Projectió ortogràfica

La principal diferència entre la projecció perspectiva i l'ortogràfica és que a l'ortogràfica no hi ha cap punt focal o focus de projecció, per tant les línies que van del pla bidimensional a l'objecte ho fan de manera paral·lela entre elles.

Dins el món de la visió per computador, la perspectiva ortogràfica és molt utilitzada ja que és un model simplificat de la projecció perspectiva, i a l'hora de fer servir vídeos reals es pressuposa que es fa servir l'ortogràfica. Però aquesta suposició no es pot fer a la lleugera, s'han de complir uns requisits mínims. Cal tenir en compte que la distància entre la càmera i l'objecte que volem projectar sigui relativament més gran que la profunditat de l'objecte, ja que d'aquesta manera el soroll perspectiu és mínim i no caldrà que el tinguem en compte. (Veure Figura 2.8).

Per tant, podem dir que el punt de mira o punt focal es troba situat a l'infinit a l'eix de les  $z$ , i la imatge 2D no queda distorsionada en les seves components  $x$  i  $y$ . (Veure Figura 2.7).

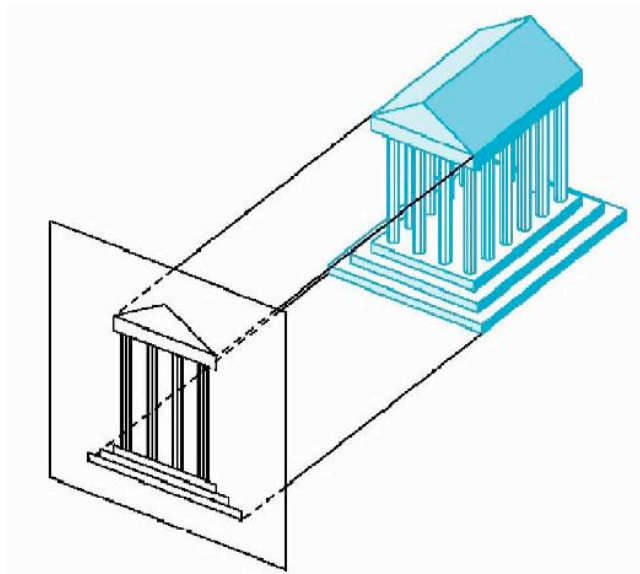


Figura 2.7 Exemple d'una imatge amb projecció ortogràfica

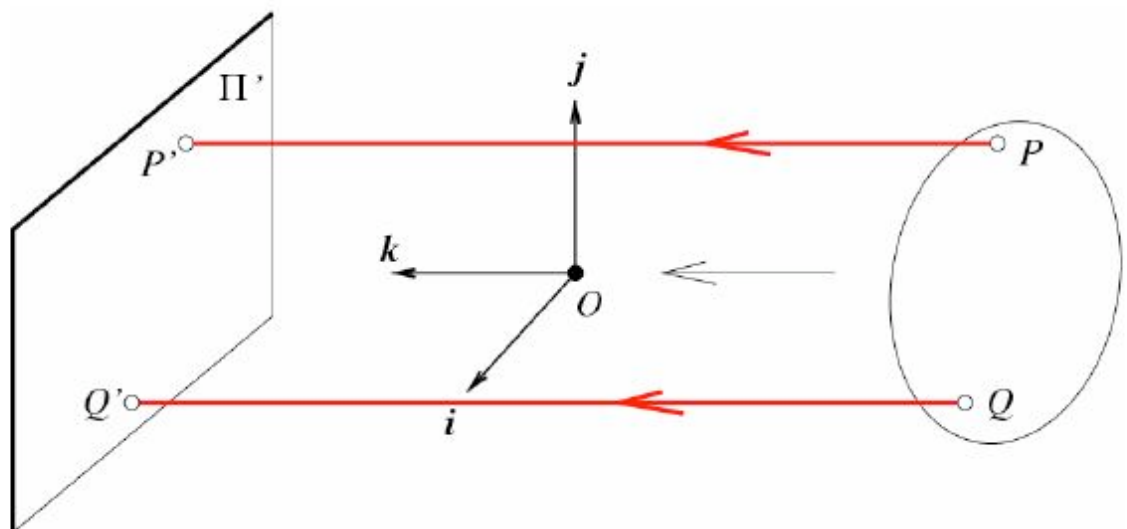


Figura 2.8 Esquema de la projecció ortogràfica.

## 2.3 Reconstrucció 3D

La reconstrucció 3D és el procés mitjançant el qual objectes reals són reproduïts a la memòria d'un computador mantenint les seves característiques físiques, és a dir, mantenint les dimensions, el volum i la forma. El que es pretén és emular el procés humà de captació i interpretació.

Dins del camp de la visió per computador hi ha múltiples de tècniques de reconstrucció 3D, però utilitzarem la basada en *Shape from Motion*. Aquest mètode parteix d'una seqüència d'imatges que contenen l'objecte a reproduir des de diferents punts de vista. De cada imatge se n'extreuen uns punts anomenats característics que pertanyen. Els mètodes de reconstrucció 3D per moviment, com és aquest cas, necessiten tenir la trajectòria que segueixen els punts característics de l'objecte a reconstruir al llarg de la seqüència. Per fer això d'una forma automàtica, s'utilitza un mètode que fa el seguiment dels punts rellevants de l'objecte. A partir d'aquest punts i de les seves trajectòries podem aconseguir-ne la reconstrucció. Cal tenir en compte que els mètodes basats en *SfM* no tenen en compte l'escala en què s'està fent la reconstrucció ni tampoc la seva rotació.

Aquests mètodes tenen un problema, i és que cal que tots els punts estiguin presents en totes les imatges, i quan s'ha de reconstruir totalment un objecte opac no és possible, ja que aniran apareixent i desapareixent punts a mesura que anem capturant totes les imatges. Per resoldre aquest problema, el que es proposa en aquest projecte és fer les reconstruccions parcials de l'objecte analitzant els punts que apareixen en pocs frames així, la diferència de punts entre el primer i l'últim frame serà mínima i per tant, la reconstrucció 3D serà més fidel a la realitat.

Un cop tenim dues reconstruccions parcials amb un frame en comú, per així assegurar que hi hauran punts en comú a les dues vistes 3D parcials, aplicarem el mètode *Iteration Closest Point*, que consisteix en buscar els punts comuns entre les dues vistes parcials

per després ajuntar-les i crear una reconstrucció més gran. De la mateixa manera s'aniran analitzant els següents frames amb grups petits per després ajuntar totes les reconstruccions parcials fins a obtenir-ne una de total.

Un cop s'hagin analitzat tots els frames de la seqüència aplicant el mètode de *Structure From Motion* i posteriorment l'*Iterative Closest Point* obtindrem la reconstrucció total de l'objecte tractat.

La part que presenta més complicació és el problema de la correspondència de punts. Els punts corresponents són aquells que contenen informació rellevant de l'objecte per poder-ne crear la vista 3D. El fet de poder saber on és cada punt en una imatge i en l'altra, anomenat problema de la correspondència de punts, ens dóna molta informació geomètrica que pot ser aprofitada per aquests mètodes de reconstrucció basats en el moviment. També cal tenir present aquest problema a l'hora d'aplicar l'*ICP*.

Els punts característics s'obtenen a partir d'un mètode de seguiment anomenat *tracking* i que els distribueix en una matriu anomenada *matriu de mesures*, que conté tots els punts que apareixen a les imatges i a quines imatges apareixen, i ve donada de la forma següent:

$$\begin{aligned}
 W = [ & \begin{array}{ccc}
 x_{1,1} & \dots & x_{1,P} \\
 y_{1,1} & \dots & y_{1,P} \\
 \cdot & & \\
 \cdot & & \\
 \cdot & & \\
 x_{2F,1} & \dots & x_{2F,P} \\
 y_{2F,1} & \dots & x_{2F,P}
 \end{array} & \qquad (2.2)
 \end{aligned}$$

on  $W$  és la matriu de mesures de dimensions  $2F \times P$ ,  $F$  és el nombre de frames i  $P$  és el nombre de punts que s'han seguit durant la seqüència. Les files senars de la matriu de

mesures representen el component  $x$  de cada punt i les parelles el component  $y$ .

Un cop obtinguda la matriu de mesures, apliquem un mètode de *Structure From Motion* on obtindrem dues matrius, una anomenada *Shape* que representa la forma de l'objecte, i una altra anomenada *Motion* que representa la rotació i la translació de l'objecte des del punt de coordenades original. Amb la matriu *Shape* ja tenim la forma de l'objecte, i si l'analitzem amb el mètode *Iterative Closest Point* juntament amb una altra matriu de forma, obtindrem una nova matriu de forma que representa la unió de les dues matrius formant una vista més gran de l'objecte.

Tenint en compte el contingut d'aquest projecte, podem dir que serà molt útil als camps de la robòtica, ja que permetrà crear la reconstrucció 3D de tot l'entorn del robot, fent que s'hi pugui moure amb més facilitat i també per si cal analitzar-lo. També és molt útil per a la identificació de persones, ja que pot reconstruir les faccions d'un rostre per així poder-les comparar i poder identificar a la persona. Últimament s'està investigant molt en l'àmbit de l'entreteniment, ja que la majoria de videojocs reconstrueixen llocs reals, com ara ciutats, per tant, també es pot utilitzar en aquest camp.

## Capítol 3

# Reconstrucció gran a partir de vistes 3D parcials

En aquest capítol es dissenyen, s'analitzen i s'implementen els diferents mètodes necessaris per poder crear una reconstrucció gran d'un objecte a partir de diferents vistes 3D obtingudes amb el mètode de *Structure from Motion*.

### 3.1 Mètodes de factorització

#### 3.1.1 Introducció

En aquest tema parlarem de dos mètodes de reconstrucció 3D a partir d'una seqüència d'imatges 2D. Ambdós mètodes pertanyen al grup de les reconstruccions 3D a través del moviment (*Shape from Motion*).

El primer mètode és el Factorization Method creat per Carlo Tomasi i Takeo Kanade [2], també creadors del mètode de seguiment que utilitzarem. Aquest mètode es basa en què la matriu de mesures té una deficiència de rang calculada de 3 sempre i quan es

consideri que la projecció és l'ortogràfica. La matriu de mesures es descomposa en dues, una que representa la forma de l'objecte i una altra que en descriu el moviment. La primera recupera la forma de l'objecte i la segons el recorregut realitzat per la càmera, és a dir, n'obté la rotació i la translació.

El segon mètode és el *Power Factorization* creat per Richard Hartley i Frederick Shaffalitzky [4], les seves bases són molt semblants a les del *Factorization Method* ja que també treballa amb la idea de la deficiència de rang de la matriu de mesures, però en aquest cas tracta de manera eficient les pèrdues de dades a la matriu de mesures. Quan es grava una seqüència d'imatges amb una càmera, el més probable és que vagin apareixent punts característics nous i en vagin desapareixent d'altres, fent que hi hagi una falta de dades, cosa que també pot ser deguda al soroll.

## 3.1.2 Factorization Method

### 3.1.2.1 Anàlisi i disseny del mètode

Tal com s'ha explicat al capítol anterior, per aquest mètode utilitzarem la *matriu de mesures* i l'anomenarem  $W$ . Per començar aquest mètode, el primer que farem serà normalitzar la matriu, és a dir, a cada valor de la matriu li restarem la mitjana dels valors de la seva fila, i la matriu resultant l'anomenarem  $W_N$  i cada valor de la matriu serà anomenat  $x_{fp}$  i  $y_{fp}$  on  $f$  indica el frame i  $p$  el punt que representen. Recordem que les files senars representen els valors  $x$  de cada punt del frame i les files parells els valors  $y$ .

Abans de continuar explicant aquest mètode primer cal comentar la relació que hi ha

entre el moviment de la càmera, la forma de l'objecte o l'escena a reconstruir i la matriu de mesures.

Es comença suposant un sistema de coordenades  $(X, Y, Z)$  fixat al centroide de punts de l'espai que correspon als punt de l'objecte de la seqüència. Un centroide és el punt on tots els plans d'un objecte es creuen i el podríem veure com el que seria un centre de gravetat en física. L'orientació del sistema de referència de la càmera respecte el sistema de coordenades corresponent al frame  $f$  ve determinat per un parell de vectors unitaris  $i$  i  $j$  que representen els vectors verticals i horitzontals del sistema de coordenades de la imatge. En una projecció ortogràfica, com hem dit al capítol anterior, les línies de projecció són paral·leles al pla imatge.

La projecció del punt  $s$  del món real al pla imatge ve donada per les equacions següents:

$$x_{fp} = i_f^T (s_p - t_f) \quad (3.1)$$

$$y_{fp} = j_f^T (s_p - t_f) \quad (3.2)$$

on el valor de  $t$  representa el vector del sistema de coordenades de referència al del pla imatge i els subíndex  $f$  i  $p$  representen l'identificador del frame i del punt a tractar respectivament.

Degut als càlculs que es realitzen per calcular les mitjanes dels valors de les coordenades  $x$  i  $y$  de cada frame, es pot escriure la matriu de mesures  $W_N$  com a :

$$W_N = RS \quad (3.3)$$

On  $R$  (*Rotació*) o  $M$  (*Motion*) representa les rotacions de la càmera i  $S$  (*Shape*) representa la forma de l'objecte.



Les files de la matriu  $R$  representen les orientacions de les coordenades verticals i horitzontals del sistema de referència, mentre que les columnes de  $S$  són les coordenades dels  $P$  punts característics de l'objecte o escena respecte el seu centroid. Les dimensions de  $M$  són  $2F \times 3$  i  $S$  és de  $3 \times P$ . A partir d'aquestes mides es pot establir el teorema del rang:

Teorema del rang: la matriu de mesures és , com a màxim, de rang 3.

Aquest teorema expressa el fet que en una matriu de mesures hi ha molta informació redundant. A causa d'això es pot dir que la translació de la seqüència 2D són les mitjanes de cada fila de la matriu de mesures  $W$  original.

Cal tenir en compte que la matriu normalitzada ha de ser forçosament de rang 3 sense soroll, per tant, quan hi ha soroll aquest rang no arriba a 3 i cal forçar-lo, i per fer-ho s'utilitza la descomposició per valors singulars.

### ***Descomposició per valors singulars***

Si la matriu normalitzada té unes dimensions de  $2F$  més gran o igual que  $P$ , es pot descomposar en una matriu  $2F \times P$  que anomenarem  $O_1$ , una matriu diagonal  $P \times P$  que anomenarem  $\Sigma$ , i una matriu  $P \times P$  que anomenarem  $O_2$ :

$$W_N = O_1 \Sigma O_2 \quad (3.4)$$

De manera que  $O_1^T O_1 = O_2^T O_2 = O_2 O_2^T = I$  on  $I$  és la matriu identitat.  $\Sigma$  és una matriu diagonal on els valors de les seves diagonals són els valors singulars ordenats en ordre decreixent.

### ***Teorema del rang per matrius de mesures amb soroll***

Un cop tenim descomposada la matriu  $W$  normalitzada, podem partir cada part de la descomposició de la següent forma:

$$O_1 = [O_1' \mid O_1''] \quad (3.5)$$

$$\Sigma = \begin{bmatrix} \underline{\Sigma} & \underline{0} \\ 0 & \Sigma'' \end{bmatrix} \quad (3.6)$$

$$O_2 = \begin{bmatrix} \underline{O}_2' \\ O_2'' \end{bmatrix} \quad (3.7)$$

$$O_1 \Sigma O_2 = O_1' \Sigma O_2' + O_1'' \Sigma O_2'' \quad (3.8)$$

La matriu de mesures ideal pel teorema del rang té com a màxim 3 valors singulars diferents de 0. Com que els valors de  $\Sigma$  estan ordenats en ordre decreixent,  $\Sigma'$  ha de contenir els valors singulars d'aquesta matriu ideal que no pertanyen al soroll. Com que el terme  $O_1'' \Sigma'' O_2''$  conté només la informació del soroll, la millor aproximació a la matriu ideal serà el producte  $W_I = O_1' \Sigma' O_2'$

Per tant, podem reescriure el teorema del rang pel cas en què la matriu de mesures tingui soroll:

Teorema del rang per matriu amb soroll: tota la informació de forma i moviment de la matriu  $W$  normalitzada es troba en els seus tres valors singulars més grans amb els corresponents vectors propis esquerres i drets.

### 3.1.2.2 Transformació mètrica

Tal com s'ha dit, les matrius  $M_I$  i  $S_I$  són transformacions lineals de les veritables matrius

$M$  i  $S$ , això vol dir que hi ha una matriu de mida  $3 \times 3$  tal que:

$$M = M_I Q \tag{3.9}$$

$$S = Q^{-1} S_I \tag{3.10}$$

Per trobar aquesta matriu  $Q$  primerament haurem de trobar la matriu simètrica  $L$  a partir de les dades donades a les propietats següents:

$$|i_f| = |j_f| = 1 \tag{3.11}$$

$$i_f^T j_f = 0 \tag{3.12}$$

que són les propietats dels vectors ortogonals, i a partir d'aquí, poden treure la següent equació:

$$L = A A^T \tag{3.13}$$

Per tant, per fer més entenedor aquest sistema, definim les següents equacions:

$$i_f^T = [i_{f1}, i_{f2}, i_{f3}] \tag{3.14}$$

$$j_f^T = [j_{f1}, j_{f2}, j_{f3}] \tag{3.15}$$

$$L = \begin{bmatrix} l_1 & l_2 & l_3 \\ & l_4 & l_5 \\ & & l_6 \end{bmatrix} \tag{3.16}$$

i a partir d'aquí s'aconsegueix reescriure el sistema de la següent manera:

$$Gl = c \quad (3.17)$$

On  $G$  és una matriu de  $3F \times 6$ ,  $l$  de  $6 \times 1$  i  $c$  de  $3F \times 1$ . On  $l$  està formada per els valors de  $l_1$  fins a  $l_6$ , i  $c$  està formada per 2/3 d'uns i la resta zeros, i  $g$  és una funció. Es resoldrà el sistema a partir del mètode de mínims quadrats. Un cop trobada la matriu  $L$  cal descomposar-la i obtenir el valor d' $A$ , cosa que farem a partir del mètode de *Cholesky*, i un cop obtinguda aquesta matriu, ja podrem calcular  $M$  i  $S$  de la següent manera:

$$M = M_l A \quad (3.18)$$

$$S = A - I S_l \quad (3.19)$$

I un cop obtenim aquestes matrius, ja hem obtingut el moviment i la forma de l'objecte que estàvem analitzant i ja es pot fer la reconstrucció 3D.

### 3.1.2.3 Implementació

Per al pseudocodi d'aquest mètode que es descriu a continuació s'ha conservat la notació i el tractament de matrius de l'entorn MatLab, ja que és l'entorn utilitzat en aquest projecte.

```
Algorisme factorization_method(W: matriu de mesures) retorna
W:matriu de mesures, ^S: matriu de forma, ^M: matriu de moviment
    f=CalcularFiles(W)
    per i de 1 fins a f fer
        W(i,:)=W(i,:) - Mitjana(W(i,:))
    fper
    //descomposem en valors singulars i només considerem els 3
    //valors propis i vectors propis.
    [U P V]=svd(W)
    //càlcul de les matrius intermitges
    ^M = U * ArrelQuadrada(P)
```

```

    ^S = ArrelQuadrada(P) * V'
falgorisme
algorisme Metric_Constraints(^M, ^S)   retorna Shape:matriu de
forma, Motion:matriu de moviment
    f=CalcularFiles(W) //
    //calcular les matrius del sistema d'equacions G1 = c
    [G1 G G3]= calcular_sistema_equacions_Glc()

    //descomposem en valors singulars i nomes considerem els 3
    //valors propis i vectors propis.
    [ls lv ld]= svd([G-c])

    l=ld(:,7)/ld(7,7)
    L=[l(1) l(2) l(3); l(2) l(4) l(5); l(3) l(5) l(6)]
    C1 = Cholesky(L)
    //Matriu de rotacions euclídea
    Motion = Ms *C1
    //Matriu de forma euclídea
    Shape = inv(C1) * Ss
    retorna [Motion, Shape]
falgorisme

```

### 3.1.3 Power factorization

El Power Factorization és una variant del mètode anterior, que consisteix en crear un algorisme iteratiu per solucionar la pèrdua de dades que es pot originar al capturar la imatge 2D. El mètode consisteix en anar estimant els valors de Motion i Shape a cada iteració fins que s'obté el resultat final.

La majoria de mètodes de reconstrucció 3D tracten la deficiència de dades i el soroll mitjançant aproximacions, per tant, la matriu resultat no s'aproxima molt a la realitat, en canvi, el *Power Factorization* utilitza la informació de què disposa per intentar trobar una solució apropiada.

### 3.1.3.1 Anàlisi i disseny del mètode

El *Power Method* és un mètode que troba el valor principal dominant d'una matriu i que convergeix ràpidament si el valor principal trobat a una iteració és suficientment gran comparat amb el trobat a la següent. Aquest mètode també es pot aplicar per trobar el subespai dominant de dimensió  $r$  creat pels primers  $r$  vectors principals que ha de servir per obtenir la descripció de les dades que el formen per després crear la reconstrucció.

En el mètode explicat anteriorment, utilitzem la *Descomposició per Valors Singulars (SVD)* de la matriu  $W$  per poder trobar un aproximació de baix rang, però també es pot obtenir trobant tant els valors com els vectors principals associats de la matriu  $WW^T$ .

Si apliquem el concepte dels subespais a aquesta nova matriu podem obtenir la següent fórmula:

$$U_k = (WW^T)U_{k-1}N_k \quad (3.20)$$

Comencem amb una matriu  $U_0$  de valors aleatoris i  $N_k$  és una matriu triangular superior que fa que les columnes de  $U_k$  siguin ortonormals. Un vector ortonormal és un vector amb norma 1, és a dir, la seva longitud és 1, però aquest concepte aplicat a les matemàtiques representa la perpendicularitat. Aquesta matriu es pot trobar pel mètode de la descomposició QR de  $(WW^T)U_{k-1}$ , conegut com a *orthonormal power iteration*.

Aquesta descomposició és un mètode de factorització en què donada una matriu  $A$  de columnes independents, troba la seva factorització en dues matrius de la forma  $A = QR$ , on  $Q$  és una matriu de  $m \times n$  amb les columnes formant una base ortogonal a  $A$  i  $R$  és una matriu triangular superior no singular, és a dir, que el seu determinant és diferent de zero.

Tenint en compte aquest mètode, podem definir la següent equació:  $W=SM^T$ , i a partir

d'aquí i començant per una matriu de rang  $r$ , es poden definir les següents equacions:

$$S_k = W^T M_{k-1} N_k \quad (3.21)$$

$$M_k = W M_k (S_k^T S_k)^{-1} \quad (3.22)$$

Si es van iterant aquestes equacions, la matriu  $W$  convergirà ràpidament si té un rang proper a  $r$ . Es comença creant una matriu  $M$  aleatòria, i a partir d'aquesta, es va solucionant alternativament  $S$  i  $M$  fins que convergeix  $W$ , però la part més important d'aquest mètode és que es pot aplicar encara que la matriu  $W$  tingui entrades buides. Per a cada valor de  $W$  hi ha una equació i aquesta s'omet quan no hi ha valor a  $W$ . Per tant, quan tenim una falta de dades a la matriu  $W$  apliquem un conjunt d'equacions lineals en lloc d'aplicar el mètode de mínims quadrats, que és una tècnica d'optimització matemàtica.

### ***Procediment de reconstrucció 3D***

El procediment de reconstrucció 3D que s'utilitza en aquest mètode s'inicia amb una matriu  $W$  de dimensions  $2F \times P$ , es comença a crear una matriu  $M_o$  de mida  $2F \times r$  i a partir d'aquesta es busca la matriu  $S_k$  de  $n \times r$  fent servir la fórmula

$$S_k = W^T M_k - I N_k \quad (3.23)$$

Per minimitzar la diferència que hi hagi entre  $\|W - M S^T\|$ , tenint en compte que només són vàlides les posicions on la matriu  $W$  té dades, en aquesta part s'utilitza el mètode de mínims quadrats. Tot seguit, a partir de la matriu  $S_k$ , trobarem la matriu  $M_k$  de  $2F \times r$  que encara minimitza més la diferència entra la matriu de mesures original i la matriu de mesures calculada, en aquest cas també utilitzem el mètode de mínims quadrats, sempre tenint en compte que només compten les posicions on a la matriu de mesures original hi hagi dades. Aquest procés s'ha d'anar repetint amb les noves  $S$  i  $M$  que es van trobant

fins que la matriu de mesures convergeixi.

Un cop trobades la  $S_I$  i la  $M_I$  cal aplicar-hi la transformació mètrica explicada anteriorment per poder recuperar les veritables  $M$  i  $S$ , és a dir, per poder obtenir el moviment i la forma que té realment l'objecte.

### 3.1.3.2 Implementació

En aquest apartat presentarem el pseudocodi del mètode de Power Factorization mantenint també la notació del MatLab.

```
Algorisme Power_Factorization(W:Matriu de mesures) retorna: W:
Matriu de mesures, ^S: Matriu de forma, ^M: Matriu de moviment
    [f,p] = CalculFilesColumnes(W) // calculem el nombre de
files i columnes de W
    //Eliminem la translació de W. Cal tenir en compte que no
//sempre tindrem totes les dades
W = treure_translacio(W)
//Calcular matriu aleatòria A0 de m x r (r=3 en aquest cas)
M=CrearMatriuAleatoria(f,3)
// iteració per trobar M i S. Acaba quan el resultat
//convergeix o quan ha fet més de 500 iteracions
mentre Resultat != ResultatAnterior i comptador <500
    //Shape
    per i de 1 fins a p fer
        w = []
        m = []
        per j de i fins a f/2 fer
            w = emplenar_amb_valors_W(w);
            m = emplenar_amb_valors_W(m);
        fper
        //Controlo que no hi hagi hagut informació per
crear les matrius w i m
        si no EsBuida(w) i no EsBuida(m) llavors
            S(i,1:3) = SolucioSistemaEquacions(w,m)T
        fsi
    fper
    //Calcular descomposició QR. Agafar les 3 primeres
columnes de Q
    [QR]= DescomposicióQR(Shape)
```



```

S = Q(:, 1:3)
//Motion
per i de 1 fins a f/2 fer
w=[]
s=[]
per j de 1 fins a p fer
    w = emplenar_amb_valors_W(w);
    s = emplenar_amb_valors_W(s);
fper
//Controlo que no hi hagi hagut informació per crear
les matrius w i m
si no EsBuida(w) i no EsBuida(s) llavors
    M(2*i-1:2*i,1:3) = SolucioSistemaEquacions(w,m)T
fsi
fper
//Guardo el resultat anterior i calculo el resultat actual
ResultatAnterior = Resultat
Resultat = MST
comptador = comptador + 1
fmentre
//Invertim S perquè quedi de la mateixa forma que en el mètode
//Factorization Method
    S = ST
retorna [W, S, M]
falgorisme

```

## 3.2 Iterative Closest Point (ICP)

La majoria de la informació referent al funcionament d'aquest mètode ha estat extreta de l'article titulat "*A Method for Registration 3D Shapes*" de Paul J. Besl i Neil D. McKay. ([6]). També s'han consultat les referències [7][8][9].

L'algorisme *Iterative Closest Point (ICP)* basat només en la geometria i alguna vegada en els colors de les dades s'ha convertit en un mètode dominant a l'hora d'alinear models tridimensionals, però aquí ens centrem només en la part geomètrica. Aquest algorisme s'usa sobretot per sortides 3D obtingudes d'un escàner on només s'escaneja l'objecte des d'una direcció cada vegada. Aquest mètode comença a partir de dos núvols de punts obtinguts d'aplicar un mètode de *Structure from Motion* a dues

seqüències d'imatges consecutives i d'una matriu que representa una transformació aproximada que hi pot haver entre els dos núvols. A partir d'aquí, i de manera iterativa, es va refinant la transformació a mesura que es van generant parelles de punts corresponents, és a dir, una parella de punts que pertanyen cadascun a un núvol de punts i que tenen les mateixes característiques, i d'aquesta manera també es va reduint l'error que sorgeix al calcular la transformació.

A un núvol de punts li direm '*data*' o  $P$ , que és la matriu amb els punts corresponents que representen la segona reconstrucció 3D obtinguda i que té una rotació i una translació respecte del primer anomenat '*model*' o  $X$ . El nombre de punts que formen  $P$  l'anomenarem  $N_p$  i seguint la mateixa notació, el nombre de punts de  $X$  serà representat per  $N_x$ .

Primerament es calcula la distància mètrica  $d$  que hi ha entre un punt de la matriu  $P$  i la matriu  $X$  mitjançant la següent fórmula:

$$d(p,A) = \min d(p, a_i) \quad i \in \{1..N_a\} \quad (3.24)$$

El punt més proper a  $X$  s'anomena  $y$  tal que  $d(p,y) = d(p,X)$  on  $y \in X$ . Aquest procés de buscar quina és la distància mínima entre un punt del conjunt  $P$  al conjunt  $X$  s'ha de repetir per a tots els punts de la matriu '*data*' i el representarem com a  $C$ , i a tot aquest conjunt de punts propers entre les dues reconstruccions l'anomenarem  $Y$ .

$$Y = C(P,X) \quad (3.25)$$

Un com tenim el conjunt de punts  $Y$  només ens faltaria fer el registre, que l'anomenarem

$$Q: (q,d) = q(P,Y) \quad (3.26)$$

Ara ja tenim les matrius ‘*data*’ i ‘*model*’, només ens falta acabar d’inicialitzar alguns elements per a la iteració on  $P_0 = P$ ,  $q_0 = [1, 0, 0, 0, 0, 0, 0]^t$  i  $k = 0$ . Com que els vectors de registre estan definits segons  $P_0$  el registre final ja serà la representació completa final.

La iteració comença calculant els punts propers tal com s’ha explicat anteriorment, fent

$$Y_k = C(P_k, X) \tag{3.27}$$

Tot seguit es calcula el registre  $(q_k, d_k) = Q(P_0, Y_k)$ , s’aplica el registre calculat a  $P$ :

$$P_{k+1} = q_k(P_0) \tag{3.28}$$

Es van iterant aquests passos fins que la distància entre les matrius és igual o menor que un *threshold* prèviament entrat al mètode.

La clau per a la convergència d’aquest mètode és que a l’anar recalculant les distàncies i aquestes fer-se més petites, fa que la distància global dels dos conjunts de punts també es vagi reduint, i l’error que es produeix al calcular aquesta distància mínima entre  $P_k$  i  $X$  es representa per la següent fórmula:

$$e_k = (1/N_P) \sum ||y_{ik} - p_{ik}||^2 \text{ on } i = 1..N_P \tag{3.29}$$

A continuació s’aplica la distància mínima trobada a  $P_0$  per poder calcular  $P_{k+1}$  tenint en compte que si l’anterior correspondència de punts amb  $Y_k$  es manté, l’error del càlcul de les distàncies serà el mateix, és a dir, continua essent  $d_k$ . Com que mentre s’executa la part que calcula el punt més proper també es crea un nou núvol de punts anomenat  $Y_{k+1}$  està clar que :

$$\|y_{i,k+1} - p_{i,k+1}\| \leq \|y_{ik} - p_{ik+1}\| \text{ per cada } i \text{ entre } 1 \text{ i } N_p \quad (3.30)$$

Normalment aquest mètode convergeix ràpid durant les primeres iteracions, però a mesura que s'aproxima al valor mínim trobat cada vegada convergeix més a poc a poc.

## 3.3 Mètode de Reconstruccions 3D grans a partir de seqüències d'imatges 2D.

### 3.3.1 Introducció

Tot i que la recuperació de la forma i del moviment d'una seqüència d'imatges és un camp molt estudiat, encara es presenten molts problemes, com per exemple el soroll generat a l'obtenir la seqüència 2D i les estimacions no lineals que es calculem a través d'un computador.

La informació referent a aquest tema està basada en el document [11].

Com que el resultat de la recuperació de la forma d'una superfície parcial d'un objecte és un núvol de punts 3D, es necessiten molts núvols de punts per poder captar la forma completa de l'objecte. Per poder ajuntar tots aquests núvols de punts, cal que primer els analitzem i els posem sobre les mateixes coordenades.

Crear la forma completa d'un objecte a partir de diferents punts de vista és conegut com '*Shape Registration Problem*'. L'algorisme *Iterative Closest Point (ICP)* [6] és el mètode principal que s'utilitza per alinear una seqüència de núvols de punts ja que, tal com hem dit anteriorment, aquest mètode calcula les corresponents parelles de punts en els núvols i minimitza l'error.

Recentment s'ha presentat un framework per alinear vistes contínues a 3D. S'analitzen els núvols de punts 3D de les imatges obtingudes a partir de càmeres situades de tal manera que captin tot l'objecte des de tots els angles, és a dir, es tan posades de forma estèreo, en aquest cas, la reconstrucció es centra en un increment de la recuperació i del registre de la forma, és a dir, que s'analitzen els núvols de punts respecte d'un núvol global, un cop analitzat el nou núvol de punts, s'ajunta al principal i tot seguit s'analitza la imatge següent.

En aquest projecte es presenta un únic framework que reconstrueix de manera automàtica el núvol de punts 3D a través de les imatges capturades. Primerament, detectem les característiques 2D del frame a analitzar, tot seguit s'analitzen les característiques dels punts obtinguts i també els dels següents frames. Un cop analitzats quatre frames consecutius, es busquen els punts que tenen les mateixes característiques i es fa una estimació de les coordenades 3D, seguidament es fa la reconstrucció d'aquestes frames, i s'obté una reconstrucció parcial que finalment s'unirà a la reconstrucció global, que no deixa de ser un núvol de punts 3D.

### **3.3.2 Recuperació de la forma a partir de poques imatges**

Està comprovat que la reconstrucció de coordenades 3D a partir de quatre punts visibles en un frame és possible si tots els punts que no tenen error de correspondència es troben

en els següents tres frames.[12]. S'ha trobat que la recuperació de la forma utilitzant només tres frames no és apropiada pels frames obtinguts a través d'un vídeo, ja que el desplaçament de tots els punts que tenen correspondència entre ells en els frames consecutius és molt semblant i molt freqüentment produeix errors.

A continuació s'explica com s'utilitza un mètode de factorització per crear la reconstrucció de quatre frames.

### 3.3.2.1 Shape Factorization

El mètode de factorització desenvolupat per Tomasi i Kanade [13] explicat anteriorment, recupera tant la forma com la rotació i translació de l'objecte a través d'una seqüència d'imatges sense haver de calcular la profunditat. Aquest mètode és robust i ràpid, però requereix una correspondència de punts de quatre o més frames, és a dir, que un mateix punt ha d'aparèixer com a mínim a quatre frames per poder-lo aplicar. D'acord amb això per poder utilitzar el màxim de punts possibles no hem de fer servir més de quatre frames, ja que al ser una seqüència d'imatges pot haver-hi punts que apareguin als primers frames però que no ho facin en els següents, per tant si agafem més de quatre frames, el nombre de punts que apareguin a tots anirà disminuint.

Assumim que hi ha quatre o més punts ( $M$ ) que tenen les mateixes característiques en els quatre frames que analitzem, és a dir, que són el mateix punt però en cadascuna de les imatges, les coordenades dels quals són:

$$(u_{(i)}^j, v_{(i)}^j) \tag{3.31}$$

on les  $i$  indiquen el frame, és a dir  $i = 1 \dots 4$ , i les  $j$  indiquen el punt,  $j = 1 \dots M$ .

Primerament es construeix la matriu de mesures  $W$ , que serà de mida  $8 \times M$ , ja que

només tindrem quatre frames, i un cop obtinguda aquesta ja podem començar a aplicar el *Factorization Method* explicat al primer apartat d'aquest capítol del projecte, ja que un cop obtinguda la matriu de mesures original, el procediment és exactament el mateix.

Un cop aplicat el mètode de factorització, obtindrem la matriu de Rotació i la matriu de Forma.

### 3.3.3 Registre i recuperació a partir de l'increment de vistes 3D

A través de quatre frames i utilitzant el *Power Factorization* descrit anteriorment, s'obté una forma parcial a través dels punts en comú entre les quatre imatges, mesurada en un sistema de coordenades local.

Anomenem  $P_k$  el  $k$ -èssim núvol de punts recuperat, és a dir, que s'han analitzat els quatre frames corresponents i s'hi ha aplicat el mètode de factorització pertinent. El resultat obtingut és la matriu de forma ja que de moment no tenim en compte el moviment. El primer element  $P_1$  s'obté a partir dels primers quatre frames,  $I_1, I_2, I_3$ , i  $I_4$ , i l'element  $k$ -èssim està format pels frames  $I_{3k-2}, I_{3k-1}, I_{3k}, I_{3k+1}$ .

La base del procés de registrar els punts és calcular totes les transformacions des de  $P_k$  fins a  $P_{k-1}$  on  $k \geq 2$ . L'última imatge utilitzada per reconstruir  $P_{k-1}$  correspon a la primera vista utilitzada per reconstruir  $P_k$ , això vol dir que l'últim frame utilitzat d'una seqüència de quatre, és el primer frame de la següent seqüència, perquè d'aquesta manera s'asseguri que hi ha punts en comú per després poder fer la reconstrucció total.

### 3.3.3.1 Registre de núvol de punts

Quan un conjunt de punts  $P$  és analitzat a través dels quatre frames, el que s'ha de fer a continuació és registrar  $P$  a un núvol de punts global  $Q$  per crear la reconstrucció completa de l'objecte. Aquest procés el que fa és alinear els dos conjunts de punts al màxim possible. La majoria dels estudis assumeixen que tots els conjunts de punts són coneguts abans de començar a fer el registre, però en el nostre cas és diferent per dues raons:

1. No necessitem inicialitzar cap mètode per començar a fer el registre dels punts.
2. L'estructura 3D reconstruïda utilitzant frames anteriors sempre està disponible, per tant, no necessitem saber quins són els núvols de punts abans de fer res, sinó que els anem analitzant i comparant amb els frames passats que ja coneixem. Això permet analitzar grans aplicacions reals on s'hi han de posar objectes virtuals.

Per a cada parell de punts que es corresponen  $p_i$  a  $P$  i  $q_i$  a  $Q$  hem de trobar la transformació rígida  $T$  tal que  $q_i = T(p_i)$ . Aquesta transformació  $T$  té sis graus de llibertat corresponents als tres angles de rotació i als tres desplaçaments o translacions, és a dir, a partir d'aquí apliquem el mètode *Iterative Closest Points (ICP)* explicat a l'apartat anterior, on busca els punts en comú entre els dos núvols de punts mirant quina és la seva distància mínima, i en acabar, ens retorna una matriu amb la rotació i una altra amb la translació que s'ha aplicat al núvol de punts obtingut dels últims quatre frames respecte del núvol de punts global.

### 3.3.3.2 Disseny de l'algorisme

El mecanisme del mètode de reconstrucció i de registre incremental funciona bàsicament en quatre passos:



**Pas 1 (inicialització):** s'inicialitza el núvol de punts global  $Q$  com un conjunt buit, és a dir, es crea una matriu buida. S'analitza el primer frame,  $I_1$  i es detecten els punts característics. El conjunt  $i$  passa a anomenar-se 1.

**Pas 2:** El conjunt  $i$  passa a anomenar-se  $i + 1$ . S'analitza el nou frame  $I_i$  i s'obtenen els punts característics d'aquest. Si els punts característics són menors que  $M$  (recordem que ha de ser de com a mínim quatre), es torna a analitzar  $I_i$  per extreure'n nous punts. Si  $(i \bmod 3) = 1$ , llavors saltarem al pas 3, ja que vol dir que hem analitzat quatre frames, si no és així, tornarem a repetir aquest pas.

**Pas 3:** recuperem la  $k$ -èssima estructura parcial,  $P_k$ , utilitzant els punts analitzats de les últimes quatre imatges (obtingudes del pas anterior)  $I_{3k-2}$ ,  $I_{3k-1}$ ,  $I_{3k}$  i  $I_{3k+1}$  on  $k = (i - 1)/3$ .

**Pas 4:** Aplicar el mètode *Iterative Closest Points* per detectar els punts en comú entre la reconstrucció obtinguda en el pas anterior i la reconstrucció global, un cop obtingudes la rotació i la translació de la nova matriu respecte de la global, caldrà registrar la nova matriu amb els seus punts amb la matriu global, per així tenir una nova matriu global amb totes les reconstruccions parcials analitzades fins al moment. Un cop arribats a aquest punt, cal repetir el pas 2 fins que ja no hi hagi més frames per analitzar.

Aquest mètode és aplicable per a qualsevol seqüència d'imatges obtinguda, ja sigui amb una càmera professional com amb una càmera d'ús domèstic. El moviment d'aquesta pot ser totalment lliure i no cal que faci cap recorregut en concret, només cal tenir en compte que com més translació i rotació hi hagi d'un frame a un altre més difícil serà la reconstrucció, ja que el nombre de *missing data* entre una matriu de punts característics d'un frame al següent pot ser molt elevat i això faria que el programa no fos del tot eficient.

## Capítol 4

### Resultats

#### 4.1 Introducció

En aquest capítol veurem el comportament dels mètodes explicats al capítol anterior.

Començarem fent les proves amb seqüències sintètiques, ja que serà més fàcil veure com es van desenvolupant els mètodes i per comprovar el seu funcionament. El soroll que s'aplicarà de manera aleatòria a la seqüència seguirà una distribució gaussiana i es provarà amb diferents valors de desviació a la mateixa seqüència per veure com afecta al desenvolupament dels mètodes. Les seqüències sintètiques les farem a partir d'un model 3D que projectarem al pla imatge de la forma que s'ha explicat en capítols anteriors. Els models sintètics 3D que utilitzarem seran un cub i un cub amb punts generats aleatòriament dins seu.

Un cop analitzades les seqüències sintètiques passarem a les reals. D'entre elles hi haurà una seqüència d'una casa de la qual disposem d'un model d'un objecte real 3D del qual nosaltres, sintèticament, n'extraïem una seqüència. Reconstruïrem la seqüència d'una casa a partir de la informació 3D obtinguda i finalment analitzarem la seqüència obtinguda d'una màscara veneciana capturada a casa, d'aquesta manera es podrà acabar d'analitzar l'eficiència del mètode per proves reals.

Les proves es realitzaran tant aplicant el *Factorization Method* com els *Power Factorization* per poder-ne comparar la seva eficiència. Cal tenir en compte que a les seqüències reals no se'ls hi aplicarà cap tipus de soroll ni de pèrdua de dades ja que al ser reals, ja tenen aquestes variables. Un cop realitzades les proves n'extraurem algunes conclusions.

## 4.2 Experiments sintètics

Aquest apartat conté les proves sintètiques amb les que provarem el mètode de reconstrucció 3D a partir de vistes 3D parcials aplicat amb cadascun dels mètodes de reconstrucció *SfM*. Cal tenir en compte que aquest tipus de proves són totalment controlables i podem disposar dels seus models 3D originals i també del moviment que ens serveixen per comparar amb els obtinguts a l'hora de reconstruir l'objecte.

Degut a la dificultat que suposa crear proves sintètiques on hi van apareixent i desapareixent punts, és a dir, de simular la visibilitat que tindriem en un objecte real i que per aplicar el *SfM* es necessiten un mínim de punts, sempre aplicarem el mètode a tot el núvol de punts 3D. Tot i això, hem anat fent grups de pocs frames, normalment de quatre, on s'hi ha aplicat primer el *SfM* i posteriorment el *ICP* fins a obtenir la reconstrucció final.

### 4.2.1 Cub

La primera seqüència sintètica que provarem serà la d'un cub 3D rígid. La funció que genera el cub permet especificar els angles de rotació i la translació que es vol que faci, així es té un control total de la trajectòria que seguirà l'estructura. A la Figura 4.1 es pot veure una mostra de la seqüència.

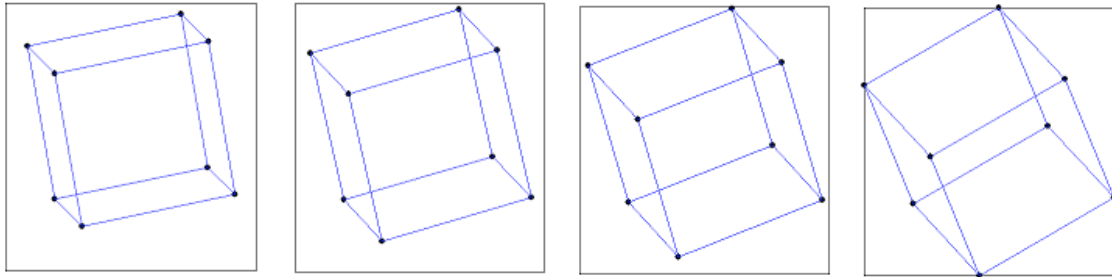


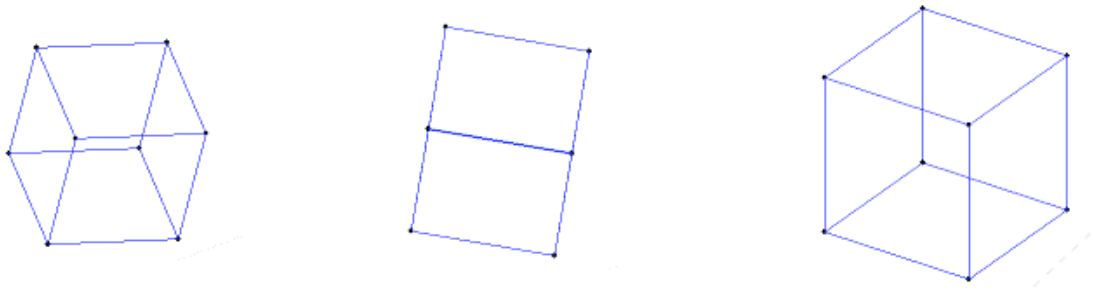
Figura 4.1 Representació de la seqüència 2D d'un cub simple. Frames 1,20,40,70.

### Factorization Method

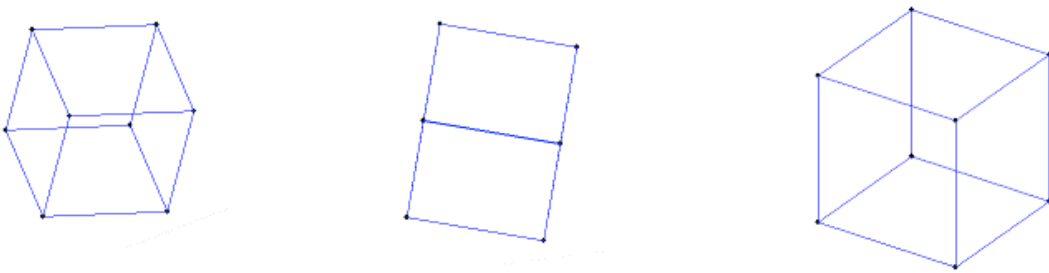
Hem analitzat diferents seqüències amb angles de rotació i translacions diferents tal com es pot veure a la Taula 4.1. Aquestes seqüències són les més simples que es poden crear, per tant, són les que ens poden indicar amb més fiabilitat si el mètode de reconstrucció es comporta de la manera esperada, ja que és un entorn totalment controlat on sabem quins han de ser els resultats. A les Figura 4.2 i Figura 4.3 es poden observar diferents reconstruccions parcials. La primera representa la primera reconstrucció obtinguda després de fer el primer *ICP*, i a la segona representa una reconstrucció intermitja. En aquest cas, des de la primera fins a la última reconstrucció, el resultat ha de ser exactament el mateix, ja que tots els punts són visibles des de la primera a la última imatge, i al haver-hi pocs punts, quasi no es va acumulant error. A la Figura 4.4 es pot veure l'última reconstrucció obtinguda, és a dir, la final, i tal com es pot comprovar, s'ha obtingut una reconstrucció perfecta.

FRAMES	ANGLES	TRANSLACIÓ
70	(20,15,10)	(10,10)
50	(20,10,15)	(20,10)
80	(40,30,20)	(50,30)
100	(50,30,20)	(20,30)

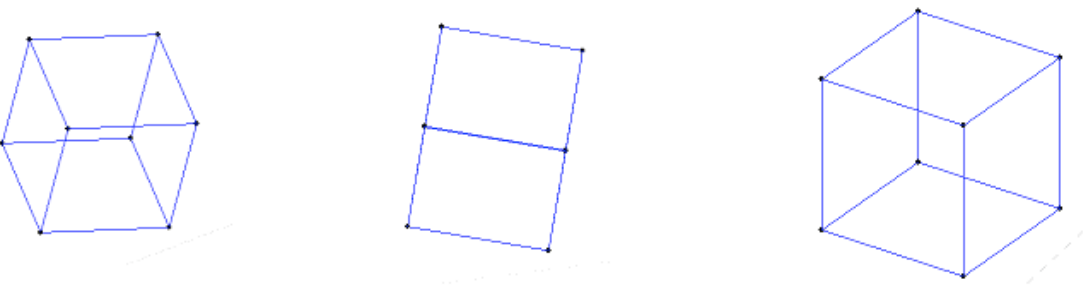
Taula 4.1 Valors de les seqüències utilitzades per a l'anàlisi del cub.



*Figura 4.2 Primera reconstrucció obtinguda de la seqüència del cub utilitzant el Factorization Method. Cada figura representa una vista diferent del cub.*



*Figura 4.3 Desena reconstrucció obtinguda de la seqüència del cub utilitzant el Factorization Method. Cada figura representa una vista diferent del cub.*

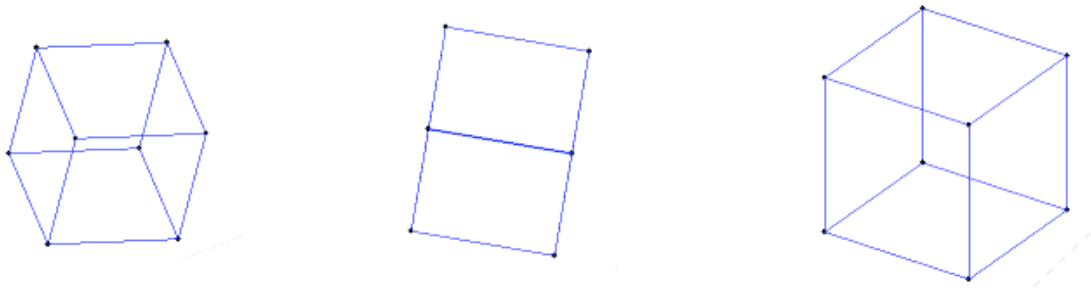


*Figura 4.4 Última reconstrucció obtinguda de la seqüència del cub utilitzant el Factorization Method. És la reconstrucció final. Cada figura representa una vista diferent del cub.*

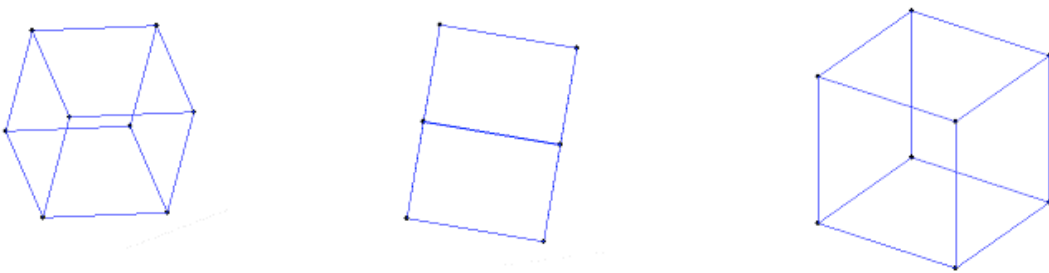
### **Power Factorization**

En aquest cas hem analitzat les mateixes seqüències anteriors però aquesta vegada utilitzant el *Power Factorization*. A les Figura 4.5 i Figura 4.6 es poden observar reconstruccions intermitges, la primera representa la primera reconstrucció parcial obtinguda, i la segona la desena reconstrucció. A la Figura 4.7 podem veure la reconstrucció final. També obtenim el mateix resultat que en el cas del *Factorization Method*.

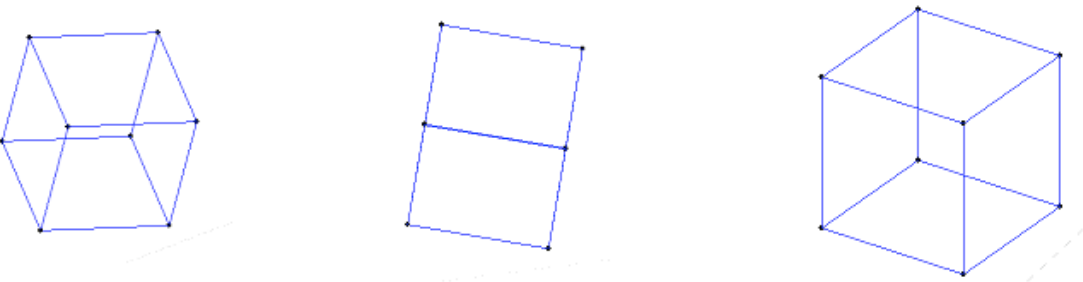
Ja s'ha explicat en capítols anteriors que el *Power Factorization* pot treballar amb falta de dades a la matriu de mesures. Per simular aquest fet hem escollit de manera aleatòria alguns valors de la matriu de mesures i se'ls ha canviat per un NaN (Not a Number). Quan s'analitza la seqüència amb aquest valors canviats es dispara l'error, ja que cada vegada es va acumulant més error. Cal tenir en compte que el *Power Factorization* fa una aproximació del la posició del valor que s'ha eliminat creant un error, si aquest error es va afegint a cada iteració, fa que es vagi acumulant juntament amb l' obtingut amb el *Iterative Closest Point* que calcula la rotació i la translació entre les dues reconstruccions. L'acumulació de l'error és el que obtenim a l'hora de sumar tots els errors obtinguts a les diferent funcions utilitzades



*Figura 4.5 Primera reconstrucció obtinguda de la seqüència del cub utilitzant el Power Factorization. Cada figura representa una vista diferent del cub.*



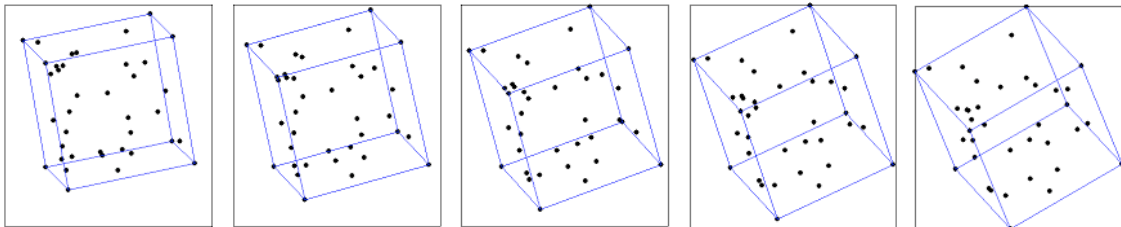
*Figura 4.6 Desena reconstrucció obtinguda de la seqüència del cub utilitzant el Power Factorization. Cada figura representa una vista diferent del cub.*



*Figura 4.7 Última reconstrucció obtinguda de la seqüència del cub utilitzant el Power Factorization. És la reconstrucció final. Cada figura representa una vista diferent del cub.*

## 4.2.2 Cub amb punts aleatoris

Un cop observats els resultats obtinguts amb la seqüència del cub, hem creat una seqüència una mica més complicada. La seqüència consisteix en un cub amb punts generats aleatòriament al seu interior. Un exemple de la seqüència és el representat a la Figura 4.8.



*Figura 4.8 Representació de la seqüència 2D d'un cub simple amb 30 punts aleatoris dins seu. Aquesta figura correspon als frames 1,30,60,90,120.*

### Factorization Method

El fet d'afegir més punts a la seqüència ens proporciona un exemple del comportament dels mètodes proposats amb un conjunt més elevat de punts rellevants. Donat que hi ha molts punts pot ser que ens trobem amb ambigüitats i per tant, que es produeixin més errors. Com que en aquest cas es veuen tots els punts des del primer a l'últim frame, ens permet veure si el mètode és eficaç o no. Per poder-ho comprovar hem generat unes quantes seqüències on s'ha variat el nombre de frames de cadascuna, el nombre de punts aleatoris i també les seves rotacions i translacions. A la Taula 4.2 es poden veure els diferents valors triats.



FRAMES	ANGLES	TRANSLACIÓ	PUNTS
70	(20,15,10)	(10,10,0)	30
50	(20,10,15)	(20,10,0)	40
80	(40,30,20)	(50,30,0)	50
100	(50,30,20)	(20,30,0)	70

Taula 4.2 Valors de les seqüències utilitzades per a l'anàlisi del cub amb punts aleatoris.

A les figures Figura 4.9 i Figura 4.10 hi podem observar algunes reconstruccions parcials obtingudes. La primera figura correspon a la primera reconstrucció obtinguda on hi apareixen diferents vistes per poder comprovar que es genera un cub perfecte. La segona figura correspon a la vintena reconstrucció obtinguda i també s'hi poden observar diferents vistes de la reconstrucció. Finalment, a la Figura 4.11, es pot observar la reconstrucció final obtinguda. Tal com es pot veure hi apareix un punt que no correspon als del cub, això és degut a tots els errors acumulats al llarg de la reconstrucció, ja que en aquest cas es disposa de més punts que en el cub normal i també de més frames, fet que fa que l'error vagi augmentant. Aquest error prové de l'aplicació del mètode de l'ICP per obtenir la rotació i la translació del primer núvol de punts amb el segon, ja que si es miren els resultats obtinguts per Ricard Campos [5], es pot veure com la reconstrucció utilitzant qualsevol dels dos mètodes estudiats de SfM no genera cap error i la matriu de mesures que conté els punts 2D és correcta. Però tot i així és un error molt petit, ja que només ha creat un sol punt més i no ha afectat a la forma real del cub i només ha aparegut a les darreres tres reconstruccions, per tant, es pot considerar que és una bona reconstrucció.

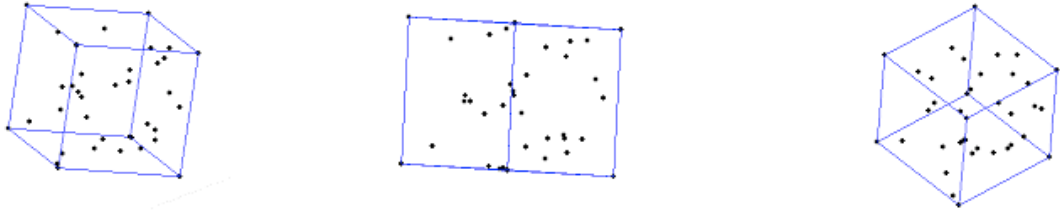


Figura 4.9 Primera reconstrucció obtinguda de la seqüència d'un cub amb 30 punts aleatoris dins seu utilitzant el Factorization Method. Cada imatge representa una vista diferent del cub.

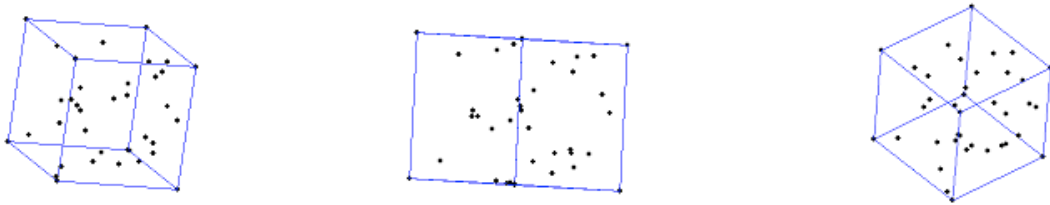


Figura 4.10 Vintena reconstrucció obtinguda de la seqüència d'un cub amb 30 punts aleatoris dins seu utilitzant el Factorization Method. Cada imatge representa una vista diferent del cub.

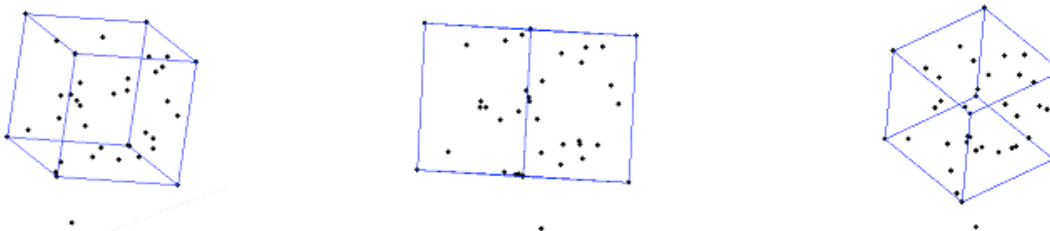


Figura 4.11 Última reconstrucció obtinguda de la seqüència d'un cub amb 30 punts aleatoris dins seu utilitzant el Factorization Method. Cada imatge representa una vista diferent del cub

### **Power Factorization**

Ara tornarem a realitzar les proves, però aquesta vegada utilitzant el *Power Factorization*.

A les Figura 4.12 i Figura 4.13 es poden observar diferents reconstruccions obtingudes. Tal com s'ha mostrat en els exemples anteriors, la primera correspon a la primera reconstrucció obtinguda amb diferents vistes per poder observar que el resultat és el desitjat, i la segona correspon a una reconstrucció intermitja, concretament la vintena, on també es poden observar diferents vistes.

A la Figura 4.14 es pot observa la reconstrucció final obtinguda, on el resultat coincideix amb l'obtingut amb el mètode anterior, el *Factorization Method*, però en aquest cas, aquest error també pot ser degut a la inicialització del *Power Factorization*, ja que inicialitza la matriu de rotacions de manera aleatòria i va fent aproximacions, fet que provoca un augment de l'error. També cal tenir en compte que és un mètode iteratiu, i que per tant, va acumulant els errors obtinguts a cada iteració efectuada.

En aquest cas també hem fet proves amb falta de dades, i els resultat ha estat els mateixos que en el cas del cub, com més falta de dades, més error acumulat apareix, ja que cal recordar que aquest mètode fa una estimació de la possible situació on es trobaria el punt que ha desaparegut, fet que el soroll augmenti i en conseqüència l'error.

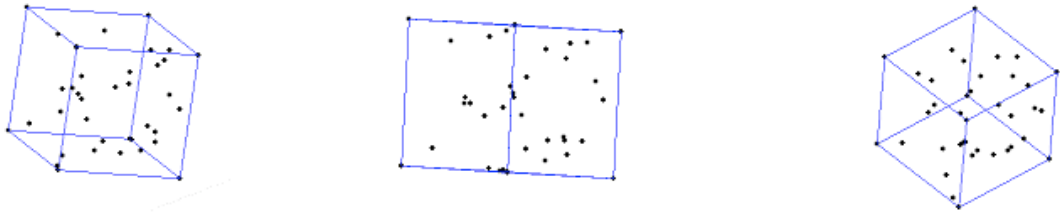


Figura 4.12 Primera reconstrucció obtinguda de la seqüència d'un cub amb 30 punts aleatoris dins seu utilitzant el Power Factorization. Cada imatge representa una vista diferent del cub.

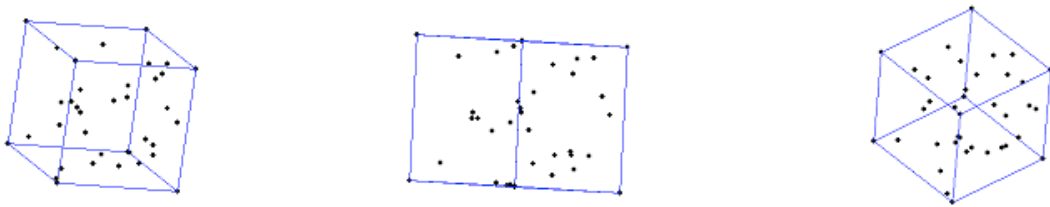


Figura 4.13 Vintena reconstrucció obtinguda de la seqüència d'un cub amb 30 punts aleatoris dins seu utilitzant el Power Factorization. Cada imatge representa una vista diferent del cub.

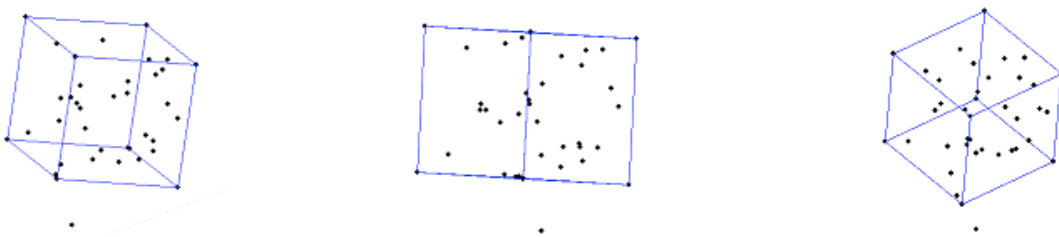


Figura 4.14 Última reconstrucció obtinguda de la seqüència d'un cub amb 30 punts aleatoris dins seu utilitzant el Power Factorization. Cada imatge representa una vista diferent del cub

## 4.3 Experiments reals

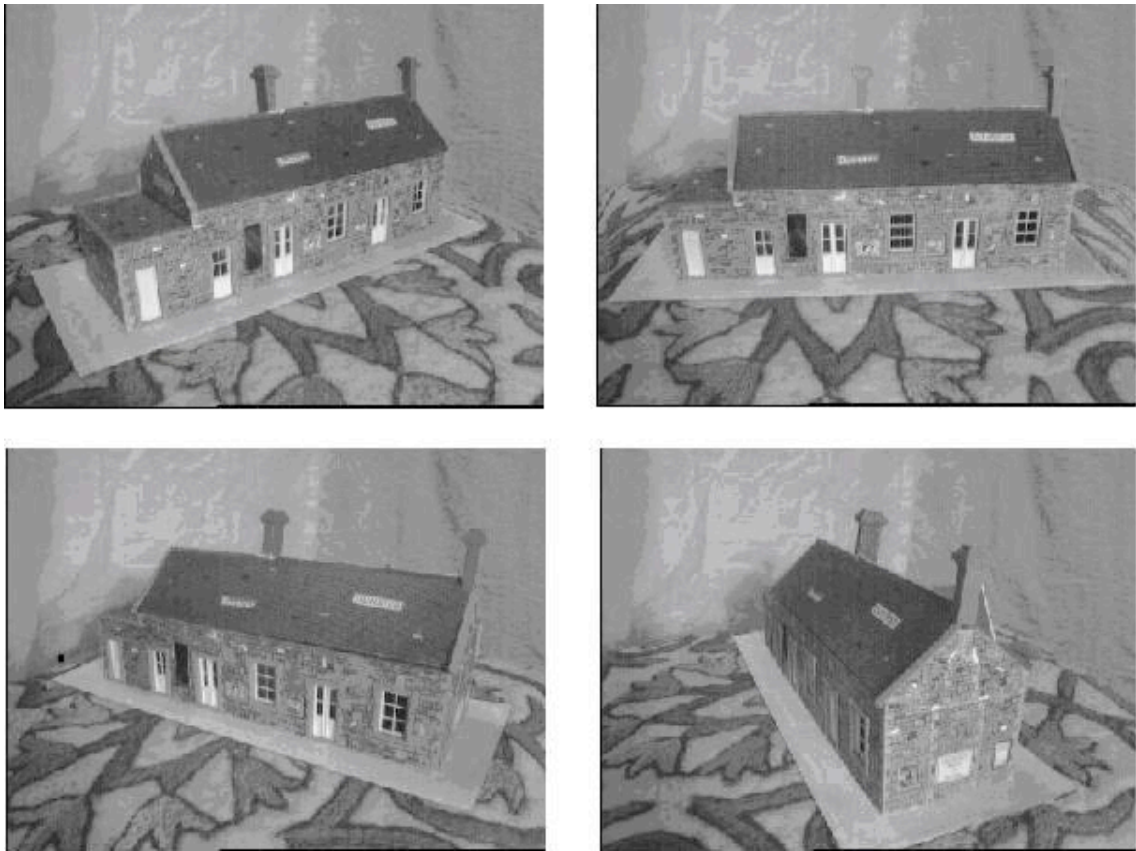
En aquest apartat provarem l'efectivitat dels mètodes a l'enfrontar-se amb problemes reals. Les estructures són més complicades que les presentades anteriorment i, per tant, reflectiran millor si són útils o no.

A diferència de les proves de la secció anterior, amb les següents seqüències no es disposa del model 3D original i per tant, només podrem comparar les resultats en funció de la seva semblança amb la imatge 2D.

### 4.3.1 Casa

En aquest experiment real disposem del model 3D d'una casa a escala tal com es veu a la Figura 4.15.

Amb aquesta prova podrem veure el comportament del mètode per objectes reals complexos. A partir del model 3D es genera la seqüència sintètica. Hem tingut en compte que els angles i translacions de la seqüència siguin els més ajustats als que es farien amb una càmera real. La seqüència generada té un total de 50 frames, els angles de rotació són (70,50,60) i la translació és de (10,0). Es pot veure una mostra de la seqüència a la Figura 4.16.

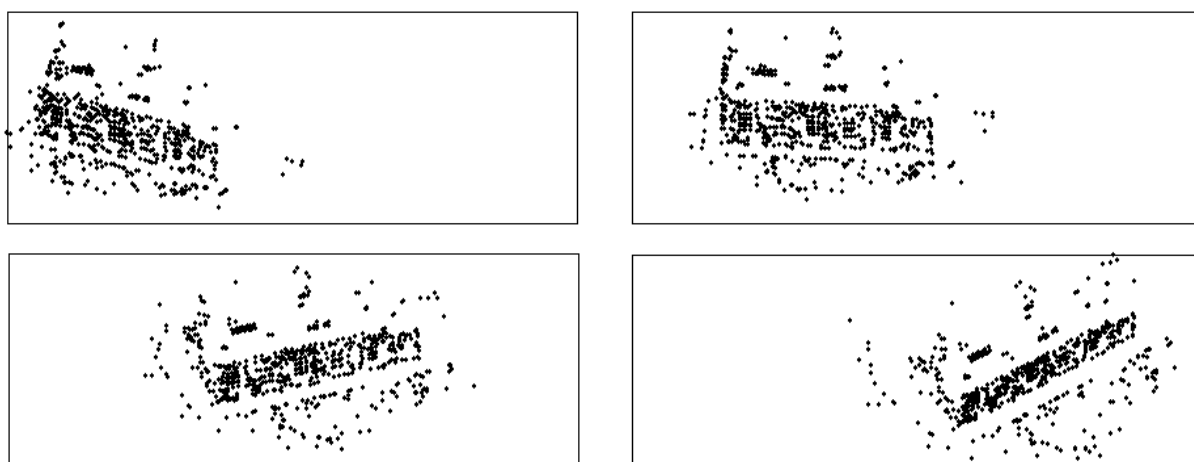


*Figura 4.15 Model original de la casa. Frames 1,17,35 i 50.*

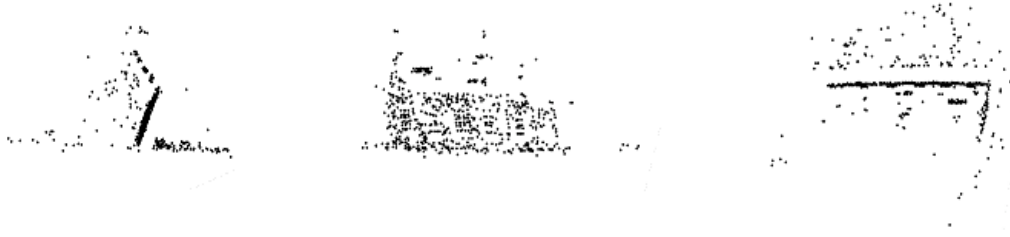
### **Factorization Method**

A les Figura 4.17 i Figura 4.18 podem veure la reconstrucció 3D de la casa en diferents etapes. Tal com es pot observar, en aquest cas tots els punts apareixen des del primer a l'últim frame, per tant, les reconstruccions obtingudes han de ser iguals. Però a mesura que anem fent més iteracions, es pot observar que hi apareix més error a l'hora de fer les reconstruccions, s'observen punts que en un principi no hi eren, com per exemple, on es veu molt clar és a la Figura 4.19 a la primera vista, a la part de la dreta hi apareixen un parell de punts que a les altres reconstruccions no hi són, això és degut a l'error acumulat, però sobretot al generat pel mètode *ICP*. Però aquests punts no són els

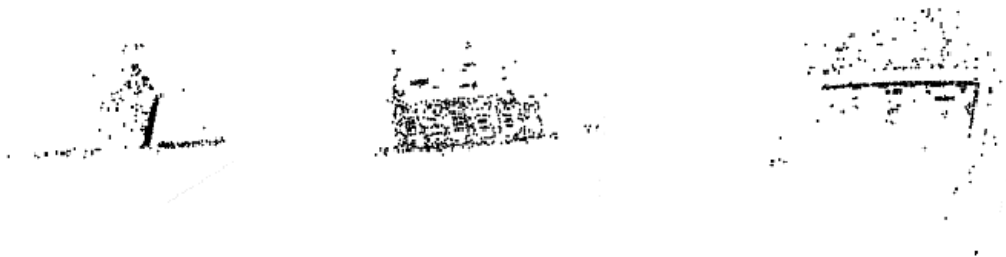
únicos que hi apareixen de més, sinó que a les altres vistes també se n'hi poden veure. A la vista que representa l'alçat, al costat dreta també n'hi apareixen.



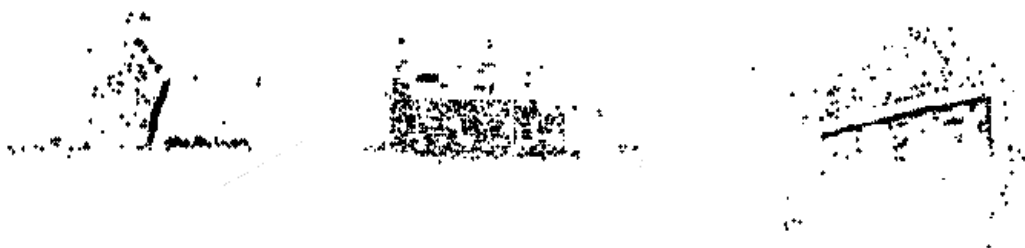
*Figura 4. 16 Seqüència de la casa generada sintèticament a partir del model 3D.  
Frames 1,17,35,50.*



*Figura 4.17 Primera reconstrucció obtinguda de la seqüència de la casa utilitzant el Factorization Method. La primera vista representa el perfil, la segona l'alçat i la tercera la planta.*



*Figura 4.18 Novena reconstrucció obtinguda de la seqüència de la casa utilitzant el Factorization Method. La primera vista representa el perfil, la segona l'alçat i la tercera la planta.*



*Figura 4.19 Última reconstrucció obtinguda de la seqüència de la casa utilitzant el Factorization Method. La primera vista representa el perfil, la segona l'alçat i la tercera la planta.*



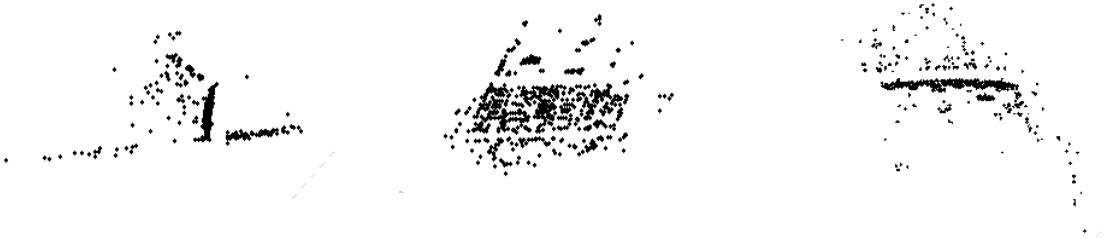
Tal com es pot comprovar a la Figura 4.19 els resultats obtinguts són bons. Si s'observen les figures amb atenció es pot veure com a les últimes figures que representen la casa, hi ha algun punt que no apareix a les altres reconstruccions intermitges, això és degut al soroll però aquest afecta molt poc al resultat final, per tant, es pot dir que la reconstrucció és bona.

### **Power factorization**

A continuació analitzarem la mateixa seqüència de la casa però aquesta vegada utilitzant el *Power Factorization*. En aquest cas a la seqüència sempre hi apareixen tots els punts a tots els frames, però tot i així, s'analitza amb aquest mètode per comprovar si afecta o no al resultat final.

A les Figura 4.20 i Figura 4.21 es pot observar com els resultats obtinguts no són massa bons, sobretot si ens fixem en la figura que representa l'alçat, ja que es pot observar clarament com queda inclinat cap a la dreta quan hauria de ser una paret dreta, a més a més, a la figura que representa la planta, l'angle que es forma hauria de ser de 90 graus, ja que representa les dues parets, en canvi el resultat obtingut és un angle més obert.

Tots aquest errors que ja es poden apreciar a les primeres reconstruccions s'accentuen quan arriben a la última, tal i com es pot veure a la Figura 4.22. Aquesta diferència d'errors entre els dos mètodes de *SfM* és deguda que en el *Power Factorization* es genera una matriu de rotacions aleatòria i es va aproximant, fet que fa que cada vegada que s'apliqui el mètode els resultats siguin lleugerament diferents. El fet que sigui un mètode iteratiu també provoca un augment de l'error, i també cal tenir en compte que és un mètode molt sensible al soroll, i al ser una seqüència real, conté soroll.



*Figura 4.20 Primera reconstrucció obtinguda de la seqüència de la casa utilitzant el Power Factorization. La primera vista representa el perfil, la segona l'alçat i la tercera la planta.*



*Figura 4.21 Setena reconstrucció obtinguda de la seqüència de la casa utilitzant el Power Factorization. La primera vista representa el perfil, la segona l'alçat i la tercera la planta.*



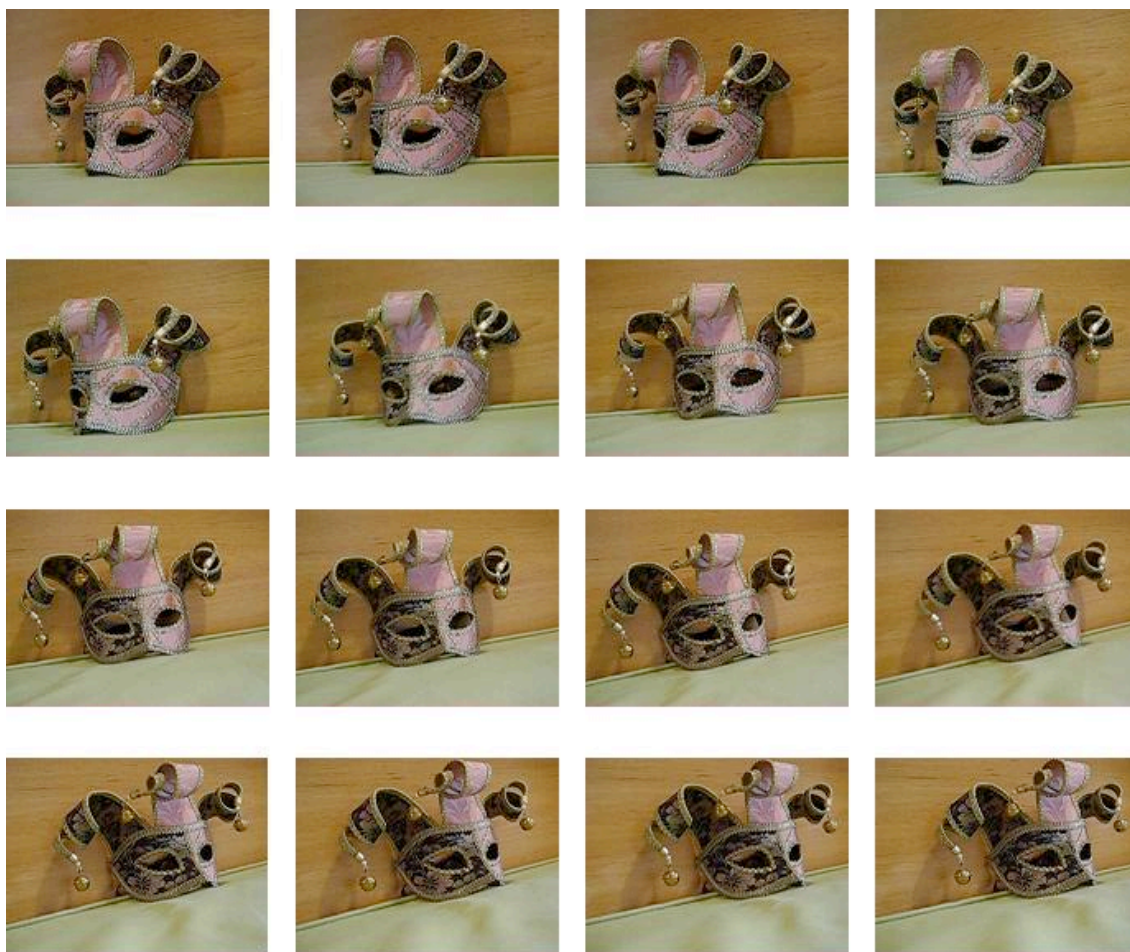
*Figura 4.22 Última reconstrucció obtinguda de la seqüència de la casa utilitzant el Power Factorization. La primera vista representa el perfil, la segona l'alçat i la tercera la planta.*

### 4.3.2 Màscara

Per acabar, demostrarem la utilitat del mètode per a una seqüència que s'ha gravat amb una càmera digital domèstica de marca KONICA model Revio KD-410Z de 4.0 Megapíxels. En aquest cas tampoc disposem del model 3D per poder comparar amb l'original, però disposem del model real, per tant, podem comparar el resultats obtinguts amb els de la Figura 4.23 i determinar si la reconstrucció és bona.

La seqüència consta d'una màscara i la càmera que comença la gravació en el costat esquerra d'aquesta fins al costat dret.

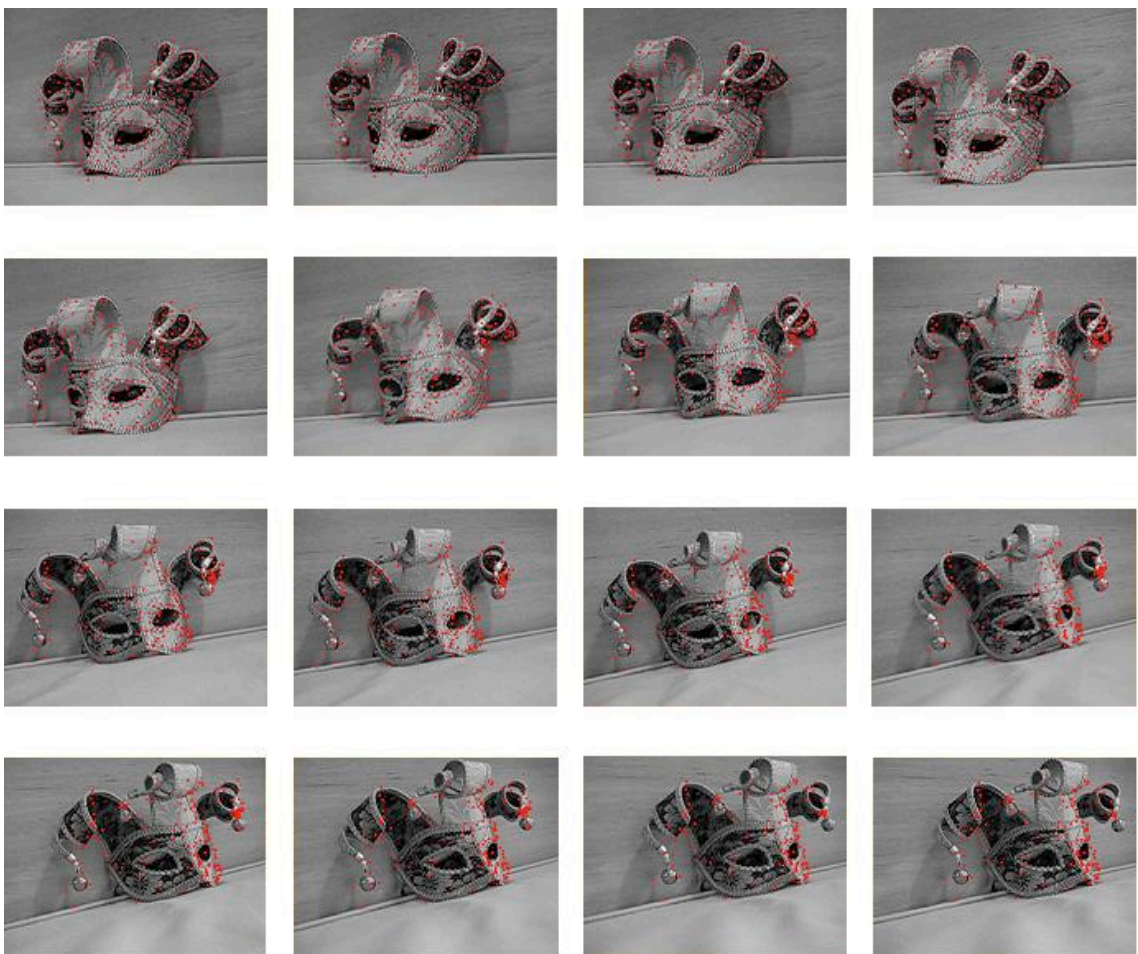
La seqüència real consta de 149 frames, una mostra dels quals és la que trobem a la Figura 4.23. A partir d'ella s'ha pogut fer un seguiment dels punts rellevants amb el sistema KLT durant tota la seqüència. Tal com podem observar en la imatge, s'ha començat gravant la cara esquerra de la màscara fent que la dreta quasi no es vegi, per tant, en el primer frame no podem obtenir punts característics d'aquest costat, i per tant, quan arribem al final de la seqüència ens trobarem que no s'ha mantingut gaires punts. A la figura Figura 4.24 es pot veure l'evolució del punts al llarg de la seqüència. Si ens hi fixem podrem veure que a les últimes imatges encara es conserven forces punts, aquest fet és degut a que les llibreries KLT s'han confós a l'hora d'agafar-los i es pensa que està agafant un punt existent quan en realitat aquest no hi és, però com que les característiques s'assemblen molt, tant les del punt original com les obtingudes, aquestes l'agafen com a punt bo.



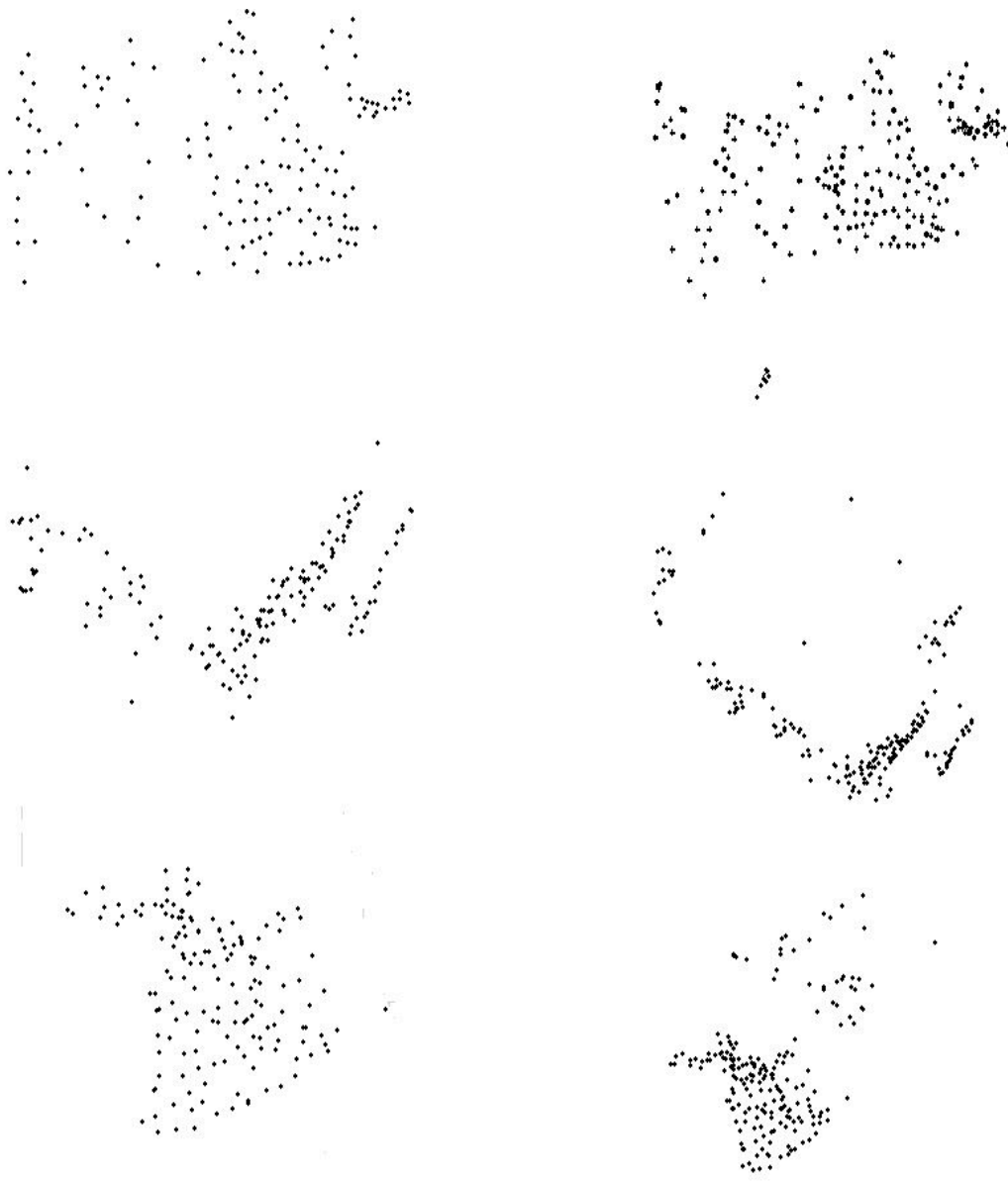
*Figura 4.23 Mostra d'alguns dels frames de la seqüència de la màscara per poder veure el moviment de la càmera.*

Si amb aquests punts utilitzem els mètodes de *Factorization Method* i *Power Factorization* es pot observar que els resultats obtinguts no s'ajusten gaire a la forma de la màscara. A la Figura 4.25 es poden observar els resultats. Les imatges de l'esquerra corresponen al *Factorization Method* i les de la dreta al *Power Factorization*. Tal com es pot comprovar, la reconstrucció obtinguda amb el primer mètode és la que més s'ajusta a la realitat, tot i que no és una bona reconstrucció, això és degut a que només agafa els punts que s'ha pogut seguir fins a l'últim frame, en canvi, en el segon mètode

es fa una aproximació de la situació dels punts que van desapareixent fent que apareguin molts punts que dificultin la reconstrucció. La imatge on es veu més clar que la reconstrucció no és bona és la del mig, que correspon a la vista que tindríem de la màscara si la miréssim des de dalt ja que a la reconstrucció s'obté una forma punxeguda i en canvi a la realitat és una corba, i si ens fixem més bé amb la imatge de la dreta veurem que hi apareixen molts punts de profunditat que no hi hauries de ser.



*Figura 4.24 Mostra dels punts agafats al llarg de la seqüència. Es pot veure que a mesura que van passant els frames, es van perdent molts punts.*



*Figura 4.25 Reconstruccions obtingudes utilitzant només SfM. La de l'esquerra s'ha obtingut utilitzant el Factorization Method i la de la dreta utilitzant el Power Factorization.*

Per poder resoldre aquest problema és quan s'utilitza el nostre mètode proposat a la secció 3.3 . Primer apliquem les llibreries KLT a grups de pocs frames per obtenir els punts característics. Després apliquem *SfM* a tots els conjunts obtinguts tenint en compte que hi ha d'haver un mínim d'un frame en comú entre ells per assegurar que hi haurà punts en comú entre les estructures 3D obtingudes i tot seguit apliquem *ICP* per obtenir la rotació i la translació entre el núvol de punts 3D global i el parcial per després unir-los i formar un nou núvol de punts global, i anar repetit aquest procés fins arribar al final de la seqüència i obtenir la reconstrucció 3D final.

Com que la seqüència d'imatges s'analitza agafant pocs frames cada vegada, utilitzarem el *Factorization Method* ja que la pèrdua de punts que hi pot haver no és elevada i a més a més, tenim la seguretat que aquests punts existeixen i no són punts calculats de manera aproximada. Aquest fet permet que la reconstrucció sigui més fiable, un exemple d'això el trobem en la seqüència de la casa analitzada anteriorment, on quan s'ha utilitzat el *Power Factorization* el resultat no ha estat bo.

En aquesta seqüència hem anat agrupant pocs frames que hem analitzat amb les llibreries KLT per agafar els punts característics comuns entre ells, sempre tenint en compte que entre els diferents grups sempre hi ha d'haver com a mínim un frame en comú amb el grup anterior per assegurar que hi haurà punts en comú entre les reconstruccions parcials, cosa que fins ara no ens preocupava, ja que tots els punts apareixien a tots els frames.

Un cop obtinguts els punts característics de cada frame, hem triat els que pertanyien a la màscara eliminant els del fons. També hem eliminat els que formen la part de dalt de la màscara, ja que al ser tirabuixons, això crea confusió entre la capa de sota i la de sobre degut a que queden superposades.

A les Figura 4.27 i Figura 4.28 es pot veure l'evolució de la reconstrucció i a la Figura 4.29 la reconstrucció final obtinguda

La primera vista representa la part frontal de la màscara, que si la comparem amb la imatge de Figura 4.26 es pot veure com les reconstruccions van evolucionant cap a una forma molta semblant, ja que es poden veure clarament els forats dels ull i la forma del nas. Sobretot a la primera, ja que és la que té menys error acumulat al ser una de les primeres reconstruccions obtingudes.

Si ens anem fixant en les diferents reconstruccions, també veurem com cada vegada hi van apareixent més punts, per exemple, a la vista frontal, la part dreta quasi no té cap punt que la representi, en canvi, quan arriben a la última reconstrucció, veiem que conté molts més punts representatius. A la vista superior també es veu molt clarament com van apareixent més punts de la meitat cap a la dreta. Això ja ens indica que aquest mètode de reconstrucció funciona en els casos reals on hi van apareixent i desapareixent punts a mesura que la seqüència avança.



*Figura 4.26 Imatge frontal de la màscara*





*Figura 4.27 Una de les primeres reconstruccions obtingudes de la seqüència de la màscara. La primera vista representa la part frontal, la segona el perfil i la tercera la part superior.*



*Figura 4.28 Reconstrucció intermitja obtinguda de la seqüència de la màscara. La primera vista representa la part frontal, la segona el perfil i la tercera la part superior.*



*Figura 4.29 Última reconstrucció obtinguda de la seqüència de la màscara. La primera vista representa la part frontal, la segona el perfil i la tercera la part superior.*

## 4.4 Conclusions

Si s'observa l'evolució de les reconstruccions de la màscara es pot veure que a mesura que es va arribant als últims es va augmentat el nombre de punts. Aquest fet demostra que aquest mètode permet fer reconstruccions grans, ja que és capaç d'anar unint les diferents reconstruccions parcials per crear-ne una de nova.

Tal com es pot comprovar, en tots els casos que s'ha n utilitzat els dos mètodes de *SfM* el que millor resultats ha donat ha estat el *Factorization Method* ja que no ha de fer una estimació de les dades que es van perdre, per tant, sempre treballa sobre valors reals.

En el cas d'aquest projecte era interessant provar el mètode del *Power Factorization* ja que si els resultats obtinguts amb aquest haguessin estat bons, s'haurien pogut ampliar el nombre de frames per crear les reconstruccions parcials fent que es disminuís el temps d'execució.

## Capítol 5

### Conclusions i treballs futurs

#### 5.1 Conclusions

Durant el desenvolupament d'aquest projecte s'han anat assolint els objectius que es van definir al principi d'aquesta documentació i al full de projecte.

S'ha intentat introduir-nos dins dels món de la reconstrucció 3D. S'ha escollit fer-ho a partir d'uns mètodes que en formen part d'aquesta temàtica i s'ha intentat comprendre el seu comportament, però per poder-ho fer ha calgut repassar conceptes de geometria i de matemàtiques en general, com per exemple el maneig de matrius, ja que és la base de tot aquest projecte. També cal dir que s'ha aprofundit en coneixements sobre aquest tema, com per exemple, el fet de formar imatges a través d'una càmera.

Aquests coneixements han permès entendre millor la teoria que ens plantejava el funcionament dels mètodes implementats fent que fos possible implementar-los en MatLab.

També s'ha aprofunditzat molt en l'ús de l'entorn MatLab, que tot i no ser del tot desconegut, s'ha comprovat que és una eina molt senzilla d'utilitzar però amb una gran potència de càlcul, sobretot a l'hora de tractar amb matrius. També cal tenir en compte que té un gran nombre de funcions ja implementades que faciliten molt el

desenvolupament de tasques complexes.

La toolbox que s'ha creat en aquest projecte està formada per un seguit de mètodes que ens permetés generar i mostrar seqüències 2D, escollir els punts que més ens interessin o eliminar dades, obtenir la rotació i la translació entre dos núvols de punts 3D, mostrar les matrius de forma i visualitzar-les des de tots els punts de vista possibles per analitzar els resultats, entre moltes altres.

S'han fet proves utilitzant tant el *Power Factorization* com el *Factorization Method* en seqüències sintètiques i reals. Les seqüències sintètiques són totalment controlables en canvi les reals estan formades per objectes més complexes. Dins de les seqüències reals, s'ha generat una seqüència a partir d'una càmera digital comuna, se li ha aplicat el seguiment de punts i tot seguit s'ha procedit a fer la reconstrucció.

## 5.2 Treballs futurs

Durant el transcurs d'aquest projecte s'han intentat assolir els objectius proposats en un principi. Tots aquests objectius han estat assolits, però això no vol dir que el projecte hagi quedat tancat. Hi ha un munt de possibilitats per poder continuar amb aquest tema. Algunes de les possibles continuacions són:

**Ampliacions immediates del projecte:** Comprovar el comportament del mètode quan es varia el nombre de frames que s'utilitza per obtenir les reconstruccions parcials i també el seu comportament variant el percentatge de *missing data* en els casos de les

sintètiques.

**Optimitzar el *Power Factorization*:** optimitzar el *Power Factorization* per reduir els errors obtinguts quan s'utilitza juntament amb l'*ICP* per poder agafar conjunts de frames més grans.

**Ampliació dels mètodes per a objectes deformables:** Els mètodes estudiats en aquest projecte només permeten reconstruir objectes rígids, és a dir, que no es deformen. Si volguéssim reconstruir una cara que va canviant d'expressió a mesura que es va gravant la seqüència no podríem. Per tant, una possible continuació seria modificar els mètodes per poder fer reconstruccions d'objectes deformables.

**Ampliació dels mètodes per a projeccions perspectives:** Tal com s'ha dit des del principi d'aquest projecte, només s'ha tractat amb projeccions ortogràfiques, ja que les perspectives ofereixen un grau de distorsió molt més elevat, per tant, els mètodes implementats no tractes la perspectiva. Un possible treball futur seria modificar els mètodes implementat perquè funcionin amb projecció perspectiva. Tot i que no s'ha tractat en aquest projecte, actualment hi ha diferents mètodes per resoldre aquest problema.

**Utilitzar aquest mètode juntament amb el de segmentació d'objectes a partir del moviment:** Els mètodes estudiats podrien fer servir la sortida d'un mètode de segmentació de moviments. El mètode de segmentació separa les trajectòries dels punts de cada objecte a partir del moviment que segueixen a la seqüència. Aquest mètode separa cada objecte de la matriu de mesures original i en crea una de més petita per a

cadascun. A partir d'aquestes matrius més petites que representen els objectes s'hi podria aplicar el mètode de reconstrucció 3D.

**Implementació del mètode de reconstrucció 3D amb un altre llenguatge:** L'entorn MatLab on s'ha desenvolupant aquest projecte està dirigit als projectes d'investigació ja que proporciona moltes eines i funcions que faciliten molt la implementació ràpida de codi. Aquest mètode estaria bé poder-ne disposar amb altres llenguatges lliures, com podria ser el C++, amb un temps d'execució suficientment baix com per poder-los utilitzar a temps real.

## 5.3 Aplicacions

Aquest projecte té moltes aplicacions. A continuació se'n descriuen unes quantes:

**Reconeixement d'objectes a partir de la seva forma:** Els computadors solen fer el reconeixement d'objectes a partir d'imatges 2D, però sempre comporta més ambigüitat que no pas una estructura 3D, per tant, seria més eficient poder fer reconeixements a partir de seqüències d'imatges per obtenir el 3D corresponent.

**Reconstrucció d'escenaris per realitat virtual:** Actualment s'utilitza molt la reconstrucció virtual per a edificis. Els mètodes estudiats es poden utilitzar per crear aquests escenaris virtuals de manera que s'ajustin més a la realitat.

**Guiatge de robots:** Guiar el robot a través d'una càmera que aniria analitzant les imatges a temps real i en faria la corresponent reconstrucció 3D i així evitar que el robot xoqui amb els diversos obstacles que es vagi trobant.

**Aplicacions mèdiques:** Actualment en la medicina s'utilitzen moltes càmeres per poder detectar diferents malalties. Aquests mètodes podrien servir per reconstruir diferents teixits i formes per poder fer el posterior diagnòstic que seria més acurat, ja que es disposaria de més informació.

**Control de qualitat:** Per al control de qualitat pot ésser molt útil ja que permet analitzar tots els productes amb molta més precisió i rapidesa que no pas l'ull humà, i per tant, permet saber si el producte compleix els requisits o no.

## Bibliografia

- [1] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, March 2004.
- [2] Carlo Tomasi and Takeo Kanade. Shape and Motion from Image Streams: a Factorization Method Parts 2, 8, 10 Full Report on the Orthographic Case. Technical report, Pittsburg PA, USA, 1992.
- [3] T. Morita i Takeo Kanade. A sequential factorization method for recovering shape and motion from image streams. In *Proceedings of 1994 ARPA Image Understanding Workshop*, volume 2, pages 1177 – 1188, November 1994.
- [4] Richard Hartley and Frederick Schaffalitzky. Power Factorization: 3D Recosntrucion with Missing or Uncertain Data. In *Australia-Japan Advanced Workshop on Computer Vision*, 2003.
- [5] Ricard Campos. Mètodes de reconstrucció 3D d'objectes a partir del moviemnt en seqüències d'imatges. 2007.
- [6] Paul J. Besl and Neil D. McKay. A Method for Registration 3D Shapes. *IEEE Conference on Pattern Analysis And Machine Intelligence*. 1992.
- [7] Timo Zinber, Jochen Schmidt and Heinrich Nierman. A Refines ICP Algorithm for Robust 3D Correspondence Estimation. *IEEE Conference on Image Processing*. 2003.
- [8] D. Chetverokov, D.Svirko, D.Stepanov and Pavel Krsek. The Trimmed Iterative Closest Point Algoarithm. *IEEE Conference on Pattern Recognition*. 2002.
- [9] Timothée Jost andt Heinz Hügli. .A Multi-Resolution Scheme ICP Algorithm for Fast Shape Registrarion. *3D Data Processing Visualization and Transmission*. 2002.
- [10] Ma Carme Julià Ferré. Missing Data Matrix Factorization Adressing the Structure From Motion Problem. Bellaterra, Decembre 2007.



- [11] Jong-Seung Park. An Incremental Approach to Shape Recovery and Registration from a Sequence of Images.
- [12] T. Huang and C. Lee. Motion and structure from orthographic projections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1989.
- [13] C. Tomasi and T. Kanade,. Shape and motion from image streams under orthography: A factorization approach," *International Journal of Computer Vision*. 1992.
- [14] A. Stoddart and A. Hilton, \Registration of multiple point sets," in *The 13th International Conference on Pattern Recognition*. 1996.
- [15] Stan Brichfield. Derivation of Kanade-Lucas-Tomasi Tracking Equation. January 1997.

## **Annex**

# **Toolbox MatLab: Manual d'usuari**

## **A.1 Estructura**

Aquest annex servirà com a manual d'usuari de la ToolBox generada en aquest projecte de final de carrera. Explicarem el funcionament dels mètodes i funcions que hem implementat especificant les seves entrades i les sortides.

- **Funcions per la reconstrucció:**

- factorization\_method
- power\_factorization
- metric\_constraints
- icp

- MÈTODE NOU!
- **Funcions per crear i manipular seqüències sintètiques:**
  - soroll
  - cub\_rigid
- **Funcions per visualitzar resultats:**
  - mostrar
  - mostrar2
- **Altres funcions**
  - klt\_converter
- **Funcions externes**
  - rotation\_matrix2euler
  - rotation\_euler2matrix

## **A.2 Funcions**

- **Factorization\_method**

**Descripció:**

Mètode de reconstrucció 3D d'una seqüència d'imatges 2D.

**Input:**

W: matriu de mesures

**Output:**

Motion: matriu de rotacions

Shape: Matriu de punt de l'objecte, estructura 3D.

- **Power\_factorization**

**Descripció:**

Mètode de reconstrucció 3D d'una seqüència d'imatges 2D. Útil per quan hi ha pèrdua de dades.

**Input:**

W: Matriu de mesures

**Output:**

Motion: matriu de rotacions

Shape: matriu de punts de l'objecte, estructura 3D.

W: matriu de mesures normalitzada.

- **Metric\_constraints**

**Descripció:**

Transformació mètrica. Transforma d'afi a euclídea.

**Input:**

S: matriu de forma de la reconstrucció

M: matriu de moviment de la reconstrucció

**Output:**

S: matriu de forma de la reconstrucció en forma euclídea

M: matriu de moviment de la reconstrucció en forma euclídea.

- **ICP (*Iterative Closest Point*)**

**Descripció:**

Mètode de reconstrucció 3D a partir de dues matrius de punts 3D.

**Input:**

Model: matriu de punts 3D global

Data: matriu de punts 3D amb dades

**Output:**

Q: matriu de punts 3D global (Model + data)

- **Reconstruccio\_total**

**Descripció:**

Reconstrucció 3D gran d'un objecte a partir d'una seqüència d'imatges 2D.

**Input:**

W: matriu de mesures.

**Output:**

Q: matriu de punts 3D amb la reconstrucció final de l'objecte.

- **Soroll**

**Descripció:**

Afegeix soroll a la matriu de mesures

**Input:**

W: matriu de mesures d'entrada

Soroll: tant per cent de soroll que s'aplica a la matriu W.

**Output:**

Ws: matriu d'entrada amb el soroll afegit.

- **Cub\_rigid**

**Descripció:**

Crea la matriu de mesures a partir d'un cub

**Input:**

F: nombre de frames que té la seqüència

Euler: matriu 1x3 que representa els angles de rotació del cub

T: vector 2x1 que indica la translació en els eixos  $x$  i  $y$  de la matriu de mesures.

Angles\_inici: angles inicials amb les rotacions que tindrà el cub al principi de la seqüència

**Output:**

W : matriu de mesures

Shape: matriu amb la forma original 3D de l'objecte

Motion: matriu de rotació real

- **Mostrar**

**Descripció:**

Mostra per pantalla l'estructura 3D entrada per paràmetre

**Input:**

S: matriu de forma

- **Mostrar2**

**Descripció:**

Mostra per pantalla dues estructures superposades

**Input:**

S1: matriu de forma 1

S2: matriu de forma 2

- **Mostrar\_cub\_rigid**

**Descripció:**

Mostra per pantalla el cub entrat per paràmetre

**Input:**

S: matriu de forma del cub a mostrar.

- **Klt\_converter**

**Descripció:**

Crea la matriu de mesures a partir dels punts que el mètode KLT ha trobat a la seqüència d'imatges 2D.

**Input:**

path: string que conté la ubicació del fitxer de sortida del mètode KLT.

**Output:**

W: matriu de mesures.

- **Rotation\_matrix2euler**

**Descripció:**

Dóna els angles eulerians d'una matriu de rotació

**Input:**

R: matriu de rotació 3x3

**Output:**

euler: vector 3x1 amb els angles originals de la rotació

- **Rotation\_euler2matriu**

**Descripció:**

Crea una matriu de rotacions a partir dels angles eulerians

**Input:**

euler: vector 3x1 amb els angles eulerians

**Output:**

R: matriu de rotació 3x3.

- **Control\_nan**

**Descripció:**

Elimina els valors NaN (Not a Number) de la matriu d'entrada

**Input:**

S: matriu de punts 3D amb valors NaN

**Output:**

S2: matriu de punts 3D sense valors NaN

- **Eliminar\_punts\_iguals**

**Descripció:**

Uneix dues matrius de punts 3D sense repetir els punts en comú

**Input:**

Q: matriu de punts 3D global

S: matriu de punts 3D nova

Thres: threshold. Diferència que hi pot haver entre els punts en comú.

**Output:**

Q2: unió de les matrius Q i S eliminant els punts en comú

- **For\_trans**

**Descripció:**

Mètode que genera una matriu a partir d'un vector

**Input:**

S: matriu de forma

T: vector de translació

**Output:**

Ts: matriu de 3xcolumnes de S on cada columna és el vector T

- **Mostrar\_2\_3D**

**Descripció:**

Mostra per pantalla dues estructures superposades

**Input:**

S1: matriu de forma 3D

S2: matriu de forma 3D

Q: matriu que alinea les dues anteriors



- **Mostrar\_3D**

**Descripció:**

Mostra qualsevol estructura 3D de punts

**Input:**

S: matriu de forma amb els punts 3D

Marker\_size: mida dels punts que es representaran

- **Mostrar\_seq**

**Descripció:**

Mostra la seqüència 2D de la matriu de mesures entrada per paràmetre

**Input:**

W: matriu de mesures

Marker\_size: tamany dels punts que es mostraran

- **Reconstruccio\_cub\_fm**

**Descripció:**

Mètode que reconstrueix un cub amb punts aleatoris o sense a partir dels mètode *Factorization Method* i ICP

**Input:**

W: matriu de mesures

**Output:**

Q: matriu de forma final. Reconstrucció 3D del cub

- **Reconstruccio\_cub\_pf**

**Descripció:**

Mètode que reconstrueix un cub amb punts aleatoris o sense a partir dels mètode *Power Factorization* i ICP

**Input:**

W: matriu de mesures

**Output:**

Q: matriu de forma final. Reconstrucció 3D del cub

**• Select\_points****Descripció:**

Mètode que permet seleccionar els punts que es considerin necessaris a partir d'una imatge 2D

**Input:**

W: matriu de mesures

**Output:**

pts: matriu de 3xpunts seleccionats on les dues primeres files són les components x i y dels punts seleccionats i la tercera representa la columna on està situat cada punt a W.