

MIRA

Documentación de Proyecto de Grado Ingeniería Eléctrica

Guillermo Carbajal
Gastón Marín
María Clara Pérez

Tutor: Álvaro Gómez

Facultad de Ingeniería
Universidad de la República

Montevideo, Uruguay
Febrero 2010

Agradecimientos

A nuestras familias.

Al Dr. Ramiro Lima por el generoso tiempo brindado y sus valiosos comentarios.

Al personal del Hospital de Tacuarembó por la buena predisposición a colaborar mostrada en todo momento.

Al Dr. Álvaro Villar.

Especialmente a Álvaro Gómez y Pablo Cancela por sus sabios consejos y compañerismo.

Tabla de Contenidos

Tabla de Contenidos	v
Lista de Figuras	XI
Lista de Tablas	XVII
1. Introducción	1
1.1. Motivación	1
1.2. Marco del proyecto	2
1.3. Objetivos del proyecto	2
1.4. Resumen del Proyecto	3
1.5. Organización de la documentación	9
2. Cirugía Guiada Por Imágenes	11
2.1. Introducción	11
2.2. Adquisición de información previo a la operación	12
2.3. Localización y seguimiento de instrumentos	12
2.4. Técnicas de Registrado	14
2.5. Visualización	16
2.5.1. Representación del estudio	16
2.5.2. Visualización en un sistema de guiado por imágenes	17
2.6. Actualización de información	18
3. Descripción de los equipos	21
3.1. Introducción	21
3.2. Descripción de los equipos en la sala de operaciones	22
3.2.1. Localizador Polaris Spectra	22
3.2.2. Microscopios	25
3.2.3. Tarjetas adquisidoras	27
3.3. Descripción de los equipos en el laboratorio	27
3.3.1. Cámaras del laboratorio	27

4. Calibración	29
4.1. Introducción	29
4.2. Modelo pinhole	29
4.3. Métodos de calibración de cámaras	35
4.3.1. Método de calibración de Zhang	36
4.3.2. Transformación Lineal Directa - Algoritmo DLT	41
4.3.3. Método de Bouguet	43
5. Implementación de la Calibración	47
5.1. Introducción	47
5.1.1. Elección del modelo pinhole	47
5.2. Particularidades de la calibración del microscopio	48
5.3. Métodos de calibración	48
5.3.1. Primera variante	50
5.3.2. Segunda variante	51
5.3.3. Tercera variante	52
5.4. Cronología del proceso de calibración	52
5.4.1. Sobre el método de evaluación de la calibración	55
5.4.2. Primeras decisiones sobre el modelo	55
5.4.3. Automatización de la adquisición de datos	56
5.5. Procedimiento de calibración	58
5.6. Resultados de Calibraciones	59
5.7. Evaluación del proceso de calibración	62
5.7.1. Descripción del procedimiento de evaluación	63
5.7.2. Experiencias con cámara de laboratorio	64
5.7.3. Experiencias con el microscopio	70
5.7.4. Conclusiones	78
6. Realidad Aumentada	81
6.1. Introducción	81
6.1.1. Definición	81
6.1.2. Aplicaciones	82
6.1.3. Software libre para RA	82
6.2. Creación del Mundo Virtual	83
6.2.1. Creación del mundo 3D	83
6.2.2. Escena OpenGL	84
6.2.3. Haciendo que parezca real	86
6.3. Alineación del mundo virtual con el real	87
6.3.1. Modelos de la cámaras	87
6.3.2. Correspondencia entre ambos modelos	87

7. Realidad Aumentada en el Microscopio	91
7.1. Introducción	91
7.2. Estado del arte	91
7.3. Descripción del sistema	92
7.4. Tareas fuera de sala	95
7.4.1. Colocación de fiduciaros	95
7.4.2. Realización del estudio	95
7.4.3. Segmentación del estudio	95
7.4.4. Calibración del puntero	96
7.4.5. Calibración del patrón	98
7.4.6. Calibración del microscopio	99
7.5. Tareas dentro de sala	99
7.5.1. Registrado del Paciente	99
7.5.2. Aplicación principal	100
7.6. Resultados	102
7.7. Reslicing	106
7.8. Fuentes de error	107
7.9. Métodos de validación de los resultados	107
7.10. Conclusiones	108
7.11. Trabajo para hacer en el futuro	108
8. Reconstrucción Estéreo	111
8.1. Introducción	111
8.2. Visión estéreo	111
8.2.1. No invertibilidad del proceso de formación de la imagen	111
8.3. Descripción de la geometría de un par estéreo	112
8.4. Matriz esencial y matriz fundamental	114
8.5. Reconstrucción estéreo y disparidad	117
8.6. Explicación detallada del proceso de reconstrucción	118
8.7. Calibración estéreo	119
8.7.1. Rectificación	120
8.8. Correspondientes	125
8.9. Algunos algoritmos para el cálculo de la disparidad	127
8.10. Triangulación	130
8.10.1. Disparidad y resolución de profundidad	131
8.11. Algoritmos utilizados para búsqueda de correspondientes	132
9. Implementación Estéreo	147
9.1. Introducción	147
9.2. Descripción de dispositivos estéreo	147
9.2.1. Par estéreo con cámaras web	147
9.2.2. Par estéreo con cámaras analógicas <i>distintas</i>	149
9.2.3. Par estéreo del Microscopio del Hospital de Tacuarembó	153
9.2.4. Una sola cámara + localizador Polaris	154

9.3. Problemas con cámaras distintas	154
9.4. Reconstrucción con una cámara + localizador Polaris	161
9.5. Reconstrucción de las superficies	163
9.6. Resultados	164
9.7. Evaluación de resultados	171
9.7.1. Error cometido en el Ejemplo 1	171
9.7.2. Error cometido en el Ejemplo 2	171
9.8. Conclusiones	173
10. IGSTK - Software para cirugía guiada por imágenes	175
10.1. IGSTK	175
10.2. Sistema de Seguimiento	178
10.2.1. Comunicación con el localizador Polaris	179
10.2.2. Creación de herramientas	180
10.3. Objetos y sus representaciones	181
10.4. Captura de Video	182
10.4.1. Video Imager component: una clase un poco más general	182
10.4.2. Modificaciones Realizadas	184
10.5. Reslicing	185
10.6. Lectura de parámetros	189
10.7. Integración con el Neuronavegador desarrollado por el IIE	189
10.7.1. Nueva clase: igstk::ViewAR	190
11. Conclusiones y trabajo a futuro	193
11.1. Calibración	193
11.2. Realidad aumentada	196
11.2.1. Trabajo para hacer en el futuro	196
11.3. Reconstrucción estéreo	197
A. Descripción del software implementado	199
A.1. Librerías utilizadas	199
A.1.1. OpenCV	199
A.1.2. IGSTK	199
A.2. Repositorio	200
A.3. Instalación	200
A.3.1. Páginas de descarga	201
A.4. Contenido de la solución	201
A.4.1. Aspectos generales de la Solución	202
A.4.2. CalibracionDamero	203
A.4.3. TestCalcularH	206
A.4.4. TestCalcularHvs2	209
A.4.5. CalibracionConCubo	209
A.4.6. RegistradoCabeza	213
A.4.7. HelloWorld	216

A.4.8. StereoUnaCamara	219
A.4.9. DisparidadyReconstruccion	222
A.4.10. EvalCalib	223
A.4.11. evalStereo	225
B. Calibración de patrones	227
C. Bases matemáticas	233
C.1. Representación de puntos y líneas en coordenadas homogéneas . . .	233
C.1.1. Método de Bouguet: cálculo de la estimación inicial de los parámetros intrínsecos	234
Bibliografía	239

Lista de Figuras

1.1. Esquema del sistema de realidad aumentada en el microscopio quirúrgico	3
1.2. Calibración del microscopio	4
1.3. Cráneo utilizado para realizar las pruebas	5
1.4. Visualización de la información en el sistema de Realidad Aumentada	6
1.5. Diagrama de pasos para realizar reconstrucción estéreo	6
1.6. Rotación y traslación relativas entre las cámaras - Imagen extraída de [23]	7
1.7. Ubicación de las cámaras de un par estéreo para que sus imágenes estén rectificadas - Imágenes extraídas de [40]	8
1.8. Imágenes rectificadas	8
1.9. Mapa de disparidad obtenido a partir de imágenes adquiridas con el microscopio del Hospital de Tacuarembó	8
1.10. Superficie 3D reconstruida a partir de imágenes tomadas con el microscopio del Hospital de Tacuarembó	9
2.1. Algunos de los instrumentos quirúrgicos que puede interesar seguir durante una intervención	14
2.2. Métodos de registrado	15
2.3. Representación 2D del estudio: cortes axial, sagital y coronal. Imagen extraída del visualizador Slicer [45].	17
2.4. Representación 3D del estudio en el visualizador Slicer	17
2.5. Visualización de la información en IGSTK	18
3.1. Disposición de los equipos en la sala de operaciones.	22
3.2. Vista del frente del localizador Polaris Spectra. Imagen extraída de [2]	22
3.3. Principio de funcionamiento del Polaris Spectra. Imagen extraída de [2]	23
3.4. Sistema de coordenadas asociado al localizador.	23
3.5. Dimensiones del volumen de trabajo del localizador. Imagen extraída de [2]	24
3.6. Herramientas y accesorios del localizador.	25
3.7. Microscopios quirúrgicos Carl Zeiss. 3.7(a) Microscopio OPMI Vario con estativo S8, Hospital de Clínicas. 3.7(b) Microscopio OPMI CS con estativo S4, Hospital Maciel. 3.7(c) Microscopio OPMI Vario con estativo S88, Hospital de Tacuarembó.	26
3.8. Tarjetas capturadoras de video.	27

3.9. Especificaciones de la cámara SSC-DC50P.	28
3.10. Especificaciones de la cámara DXC-107P.	28
4.1. Propiedades de las lentes delgadas.	30
4.2. Formación de una imagen a través de una lente delgada.	31
4.3. Modelo pinhole.	32
4.4. Modelo pinhole frontal.	32
4.5. Patrones de calibración.	35
4.6. Relación entre los diferentes sistemas de coordenadas, parámetros intrínsecos y parámetros extrínsecos.	36
5.1. Esquema de calibración.	49
5.2. Cálculo de parámetros intrínsecos utilizando sólo un damero	50
5.3. Cómputo de parámetros intrínsecos con un damero y el localizador Polaris. Las coordenadas de las esquinas están dadas en el sistema de referencia del localizador.	51
5.4. Detección automática de esquinas utilizando OpenCV	57
5.5. Calibración con patrón cúbico. 5.5(a) a 5.5(c) Segmentación de las caras izquierda, derecha y superior. 5.5(d) Detección automática de las esquinas del patrón mediante la función <code>cvFindChessCubeCorners</code> . Se obtienen las coordenadas de las esquinas de forma ordenada.	58
5.6. Imagen del patrón de calibración utilizada para el procedimiento de evaluación. Las posiciones detectadas de las esquinas se marcan con ×, mientras que con ⊗ se indican las reproyecciones. El error de reproyección en este caso fue de 1.52 píxeles.	63
5.7. Error en la reproyección de puntos, aumento 2, distancia de trabajo 300 mm, Conjunto de parámetros 1. Con el símbolo × se marcan las esquinas <i>detectadas</i> , mientras que el símbolo ⊗ corresponde a las esquinas <i>proyectadas</i> . 72	72
5.8. Error en la reproyección de puntos, aumento 2, distancia de trabajo 300 mm, Conjunto de parámetros 2. Con el símbolo × se marcan las esquinas <i>detectadas</i> , mientras que el símbolo ⊗ corresponde a las esquinas <i>proyectadas</i>	73
5.9. Error en la reproyección de puntos, aumento 3, distancia de trabajo 300 mm, Conjunto de parámetros 1. Con el símbolo × se marcan las esquinas <i>detectadas</i> , mientras que el símbolo ⊗ corresponde a las esquinas <i>proyectadas</i> . 75	75
5.10. Error en la reproyección de puntos, aumento 2, distancia de trabajo 300 mm, Conjunto de parámetros 2. Con el símbolo × se marcan las esquinas <i>detectadas</i> , mientras que el símbolo ⊗ corresponde a las esquinas <i>proyectadas</i> . 76	76
5.11. Imagen de un patrón de calibración tomada con el microscopio y cámara del Hospital de Clínicas.	78
6.1. Realidad de Milgram-Virtuality Continuum	82
6.2. Realidad Aumentada con Artoolkit	83

6.3.	Cadena de transformaciones que se aplican para generar la escena. Imagen extraída de [48]	84
6.4.	Parámetros que definen el área visible y la perspectiva utilizando OpenGL (frustum). Imagen extraída de [4].	85
6.5.	Modelos de las cámaras. Imágenes extraídas de [48]	88
6.6.	Ubicación de los ejes en la imagen	90
6.7.	Realidad Aumentada utilizando OpenCV y OpenGL	90
7.1.	Esquema del sistema de realidad aumentada en el microscopio quirúrgico .	93
7.2.	Cráneo utilizado para realizar las pruebas	95
7.3.	Creación de modelos de superficie en 3D Slicer [45]	96
7.4.	Calibración del puntero mediante pivoteo	97
7.5.	Calibración del microscopio utilizando un damero	99
7.6.	Posibles configuraciones para un modelo de la punta	101
7.7.	Grafo de la Escena	102
7.8.	Se observan el cráneo de pruebas, el puntero y una computadora que muestra el video con realidad aumentada	103
7.9.	Modelos añadidos virtualmente	104
7.10.	El puntero y la estructura que está tocando se encuentran dentro del cráneo cerrado	105
7.11.	Intersección entre el modelo 3D del tumor y el plano que pasa por la punta de la herramienta y es perpendicular al eje óptico para dos profundidades distintas	106
8.1.	Conceptos de la geometría epipolar	112
8.2.	Traslación y rotación que describen la posición relativa entre las cámaras. Imagen extraída de [23]	114
8.3.	Geometría de un par estéreo para la definición de la disparidad. Imagen extraída de [40]	117
8.4.	En la figura se ve un punto en el espacio y sus proyecciones sobre las imágenes; estas proyecciones son puntos correspondientes.	119
8.5.	Movimientos para escribir las coordenadas de un punto P referido al mundo en coordenadas normalizadas de las cámaras	120
8.6.	Cambios en las líneas epipolares al rectificar	121
8.7.	En las zonas negras se aprecia la pérdida de información al rectificar . . .	122
8.8.	Proceso de rectificación paso a paso	124
8.9.	Geometría de un par estéreo para la definición de la disparidad. Imagen extraída de [40].	125
8.10.	Mapa de disparidad y relación entre disparidad y profundidad	126
8.11.	Relación entre profundidad y disparidad, cortesía de Javier Preciozzi. . . .	131
8.12.	Resultados del prefiltrado que realiza el algoritmo Block Matching variando sus parámetros	135
8.13.	Resultados del algoritmo Block Matching con distintos tamaños de ventanas.	137
8.14.	Resultados del algoritmo Block Matching con distintos tamaños de ventanas.	138

8.15. Mapas de disparidad obtenidos con Graph Cut	140
8.16. Gruond Trurh y regiones de las imágenes sobre las que se analizan los errores	141
8.17. Regiones de los mapas de disparidad para la región A mostrada en la imagen 8.16(a)	142
8.18. Regiones de los mapas de disparidad para la región I mostrada en la imagen 8.16(a)	143
8.19. Mapas de disparidad de la región C mostrada en la imagen 8.16(b)	144
8.20. Mapas de disparidad de la región C mostrada en la imagen 8.16(b)	144
9.1. Par estéreo con cámaras web.	148
9.2. Ejemplos de imágenes capturadas con el par estero de cámaras web.	148
9.3. Rectificación de un par estéreo de cámaras web y mapa de disparidad.	149
9.4. Par estéreo con cámaras analógicas distintas.	150
9.5. Ejemplos de imágenes capturadas con el par estero de cámaras analógicas distintas.	151
9.6. Resultados utilizando cámaras analógicas: imágenes rectificadas - mapa de disparidad.	152
9.7. Resultados utilizando cámaras analógicas: imágenes rectificadas - mapa de disparidad. Baja resolución del mapa de disparidad.	152
9.8. Cámaras presentes en el microscopio del Hospital de Tacuarembó.	153
9.9. Imágenes que se utilizaron para realizar las pruebas a los algoritmos de búsqueda de correspondientes.	155
9.10. Cambio en el contraste de la imagen derecha.	155
9.11. Mapas de disparidad para las imágenes sin modificar.	156
9.12. Mapas de disparidad resultantes al variar el contraste de una imagen.	157
9.13. Mapas de disparidad utilizando como base las imágenes con ruido gaussiano.	158
9.14. Mapas de disparidad utilizando como base las imágenes con ruido gaussiano.	159
9.15. Análisis del efecto de una mala rectificación en los mapas de disparidad.	160
9.16. Se observa que sólo con la posición relativa entre los marcadores no se puede conocer la posición relativa entre las cámaras	162
9.17. Esquema de transformaciones y movimientos vinculados a la realización de reconstrucción estéreo con una sola cámara	163
9.18. Modelo utilizado para la reconstrucción estéreo.	164
9.19. Ejemplo 1 de imágenes rectificadas y el mapa de disparidad de la imagen izquierda utilizando el dispositivo estéreo armado en el Hospital de Tacuarembó.	165
9.20. Ejemplo 2 de imágenes rectificadas y el mapa de disparidad de la imagen derecha utilizando el dispositivo estéreo armado en el Hospital de Tacuarembó.	166
9.21. Reconstrucción de la superficie generada a partir del par de imágenes del Ejemplo 1 utilizando una nube de puntos con textura en Matlab	167
9.22. Reconstrucción de la superficie del Ejemplo 1 utilizando triangulación de una nube de puntos en Matlab	167
9.23. Reconstrucción de la superficie del Ejemplo 1 utilizando MeshLab	168

9.24. Reconstrucción de la superficie generada a partir del par de imágenes del Ejemplo 2 utilizando una nube de puntos con textura en Matlab.	169
9.25. Reconstrucción de la superficie del Ejemplo 2 utilizando triangulación de una nube de puntos en Matlab.	170
9.26. Reconstrucción de la superficie del Ejemplo 2 utilizando MeshLab.	170
10.1. Arquitectura utilizada por IGSTK. Figura extraída de [10].	176
10.2. Clases encargadas del seguimiento de herramientas en IGSTK. Imagen extraída de [10].	179
10.3. Herencia de algunos Objetos Espaciales en IGSTK. Imagen extraída de [10].	181
10.4. Ejemplo de representación de objetos en IGSTK.	182
10.5. Componentes de la clase VideoImager. Figura extraída de [10]	183
10.6. Jerarquía de clases de Reslicing	186
10.7. Intersección entre el modelo 3D del tumor y el plano que pasa por la punta de la herramienta y es perpendicular al eje óptico, para tres profundidades distintas.	188
10.8. Herencia de la clase ViewAR	191
11.1. Imagen de un patrón de calibración tomada con el microscopio y cámara del Hospital de Clínicas.	194
A.1. Arquitectura utilizada por IGSTK.	200
A.2. Diagrama de flujo del proyecto <i>CalibrarDamero</i>	205
A.3. Diagrama de flujo del proyecto <i>TestCalcularH</i>	208
A.4. Diagrama de flujo del proyecto <i>CalibracionConCubo</i>	212
A.5. Diagrama de flujo del proyecto <i>RegistradoCabeza</i>	215
A.6. Diagrama de flujo del proyecto <i>HelloWorld</i>	218
A.7. Diagrama de flujo del proyecto <i>StereoUnaCamara</i>	221
A.8. Diagrama de flujo del proyecto <i>DisparidadYReconstruccion</i>	223
B.1. Patrón plano y patrón tridimensional con herramientas adheridas.	228
B.2. Calibración de patrones.	229
B.3. Patrón plano de casillas de 6 mm y su sistema de coordenadas asociado. .	230
B.4. Patrón cúbico de casillas de 4 mm y su sistema de coordenadas asociado. .	231
C.1. Vista perspectiva del patrón de calibración, junto con los puntos de fuga y la recta de fuga asociada al plano del patrón.	235

Lista de Tablas

5.1. Resultados obtenidos al calibrar el microscopio del Hospital de Clínicas con un aumento de 3 para varios valores de distancia focal. Se utilizó el Camera Calibration Toolbox for Matlab.	53
5.2. Parámetros intrínsecos iniciales	54
5.3. Utilizando todo el modelo	54
5.4. Considerando sólo un coeficiente de distorsión radial	54
5.5. Sin considerar el modelo de distorsión	54
5.6. Considerando un coeficiente de distorsión y fijando el punto principal en el centro	54
5.7. Sin considerar el modelo de distorsión y fijando el punto principal en el centro	54
5.8. Calibración del microscopio del Hospital de Tacuarembó. Resultados de la Calibración 1. Aún partiendo de una estimación inicial mala para los parámetros intrínsecos se lograron buenos resultados	61
5.9. Calibración del microscopio del Hospital de Tacuarembó. Resultados de la Calibración 2.	61
5.10. Resultados de la Calibración 3	61
5.11. Error RMS en la reproyección de las esquinas del patrón de calibración, para una de las cámaras del laboratorio. La distancia del patrón a la cámara varía entre 25 y 50 <i>cm</i> , mientras se mantiene fijo el aumento en 12,5.	65
5.12. Error RMS en la reproyección de las esquinas del patrón de calibración, para una de las cámaras del laboratorio. La distancia del patrón a la cámara varía entre 25 y 55 <i>cm</i> , mientras se mantiene fijo el aumento en 15.	66
5.13. Error RMS en la reproyección de las esquinas del patrón de calibración, para una de las cámaras del laboratorio. La distancia del patrón a la cámara varía entre 40 y 60 <i>cm</i> , mientras se mantiene fijo el aumento en 20.	67
5.14. Error RMS en la reproyección de las esquinas del patrón de calibración, para una de las cámaras del laboratorio. La distancia del patrón a la cámara varía entre 25 y 55 <i>cm</i> , mientras se mantiene fijo el aumento en 30.	68
5.15. Parámetros intrínsecos A y transformación H resultantes de calibrar una cámara del laboratorio, con patrón plano y cúbico.	69
5.16. Error RMS en la reproyección de puntos, aumento 2, distancia de trabajo 300 mm, Conjunto de parámetros 1. Para estas imágenes, un pixel equivale aproximadamente a 0.089 mm.	71

5.17. Error RMS en la reproyección de puntos, aumento 2, distancia de trabajo 300 mm, Conjunto de parámetros 2. Para estas imágenes, un pixel equivale aproximadamente a 0.085 mm.	71
5.18. Error RMS en la reproyección de puntos, aumento 3, distancia de trabajo 300 mm, Conjunto de parámetros 1. Para estas imágenes, un pixel equivale aproximadamente a 0.062 mm.	74
5.19. Error RMS en la reproyección de puntos, aumento 3, distancia de trabajo 300 mm, Conjunto de parámetros 2. Para estas imágenes, un pixel equivale aproximadamente a 0.091 mm.	74
5.20. Parámetros de calibración correspondientes a las Configuraciones 1 y 2.	77
8.1. Error RMS.	145
8.2. Número de puntos sin cálculo de disparidad.	145
9.1. Primer Ejemplo	172
9.2. Segundo Ejemplo	172

Resumen

En este proyecto se presenta el desarrollo y puesta en funcionamiento de un sistema para guiar por imágenes utilizando realidad aumentada en un microscopio de Neurocirugía. Su propósito es asistir al neurocirujano para que la cirugía se torne más segura y menos invasiva. Para ello se superponen a la señal de video del microscopio modelos gráficos de la anatomía del paciente extraídos de estudios preoperatorios. La información obtenida de los estudios es válida hasta que el cerebro sufre ciertas deformaciones que se conocen como *brain-shift*. Una forma de evaluar estos desplazamientos, para luego actualizar la información preoperatoria es mediante mallas de la superficie del cerebro construidas a partir de imágenes del microscopio quirúrgico.

Se muestran y analizan los resultados obtenidos al aplicar realidad aumentada con el microscopio del Hospital de Tacuarembó a un cráneo de pruebas. También se exponen las superficies construidas.

Capítulo 1

Introducción

1.1. Motivación

Al momento de realizar una operación el neurocirujano cuenta con las imágenes de estudios realizados antes de la intervención, por ejemplo una selección de cortes de tomografía computada (TC) o resonancia nuclear magnética (RNM) impresos en un film. Con esta información el cirujano debe determinar la ubicación espacial de las estructuras a las que desea llegar y decidir la mejor trayectoria para alcanzarlas. La integración mental de toda esta información requiere una dosis importante de experiencia y ubicación espacial.

Ante este panorama, la navegación guiada por imágenes en cirugía se presenta como una efectiva herramienta para ayudar al cirujano en las operaciones neurológicas. Los neuronavegadores (sistemas de guiado por imágenes aplicados a neurocirugía) permiten desplegar la ubicación de los instrumentos quirúrgicos sobre las imágenes para guiar al cirujano en zonas de difícil visibilidad o acceso, brindando una realimentación visual de la trayectoria que se está siguiendo. De esta forma, contribuyen a lograr una apertura de los tejidos muy precisa y localizada en torno a la lesión, reduciendo así considerablemente la posibilidad de dañar involuntariamente algún órgano o tejido del cuerpo.

En ciertas intervenciones, los cirujanos utilizan el microscopio quirúrgico para la localización y la mejor visualización de estructuras cerebrales. En el entendido de que una fuerte correlación entre lo observado y la información extraída de los estudios contribuye a un mejor aprovechamiento de la información, es que surge como una interesante alternativa incorporar al microscopio quirúrgico como una herramienta más de guía en el proceso de neuronavegación. Así, por ejemplo, se podría mostrar en una señal de video extraída del microscopio la proyección en la imagen de una estructura anatómica que no es visible para el cirujano. O actualizar los cortes que se muestran al neurocirujano en el sistema de neuronavegación de acuerdo a lo que se está viendo en el microscopio.

La información que se brinda al neurocirujano en los sistemas de guiado por imágenes actuales se obtiene a partir de imágenes de TC y/o RNM que se realizan

antes de la operación. Esto hace que la información de posición extraída de los estudios sea válida en tanto la forma del cerebro no cambie sustancialmente. Por ello el neuronavegador es útil en intervenciones con pequeñas aperturas de cráneo o en las primeras etapas de una intervención mayor, antes de que el cerebro sufra desplazamientos (conocidos en la literatura médica como *brain-shift*). La compensación del *brain-shift* para poder asistir al neurocirujano a lo largo de toda una intervención es tema de investigación en diversas partes del mundo. Una alternativa es construir una malla de la superficie del cerebro a partir de imágenes obtenidas con el microscopio quirúrgico y luego utilizar estos datos para estimar el corrimiento del cerebro y actualizar así la información mostrada al neurocirujano.

Los dispositivos de guiado a través de imágenes médicas son utilizados frecuentemente en diversos hospitales del mundo, fundamentalmente en países desarrollados y con un gran potencial económico. Si bien el beneficio de su utilización en algunas intervenciones está fuera de discusión, en Uruguay no existe hasta el momento, ningún navegador por imágenes médicas. La razón es que su costo es tan elevado que resulta prohibitivo para nuestro medio.

1.2. Marco del proyecto

Actualmente, en el marco de un proyecto CSIC (Comisión Sectorial de Investigación Científica), el Grupo de Tratamiento de Imágenes (GTI) del Instituto de Ingeniería Eléctrica (IIE) está desarrollando un neuronavegador. Dicho proyecto es financiado conjuntamente por la CSIC y por ASSE (Administración de los Servicios de Salud del Estado). Cuenta además con el Hospital de Clínicas como parte interesada en el desarrollo del mismo.

El proyecto surge a raíz del deseo de neurocirujanos del Hospital de Clínicas y del Hospital de Tacuarembó de contar con un neuronavegador a un costo apreciablemente menor que el que se ofrece comercialmente. Esta disminución considerable en el precio es posible debido a que los costos asociados al neuronavegador son de desarrollo, en gran parte de software, y a que es un producto con un potencial mercado reducido.

Este proyecto de fin de carrera se enmarca dentro del proyecto Neuronavegador y por consiguiente comparte la misma fuente de financiación.

1.3. Objetivos del proyecto

El objetivo principal de este proyecto es la incorporación del microscopio quirúrgico al sistema de neuronavegación que se está desarrollando en el IIE. Con ese fin se trazaron inicialmente dos objetivos:

- Desarrollo y puesta en funcionamiento de un sistema para guiar por imágenes utilizando realidad aumentada en un microscopio de neurocirugía.

Se extraerá del microscopio una señal de video con lo visto por uno de sus oculares. Se añadirá a la misma, virtualmente y en tiempo real, información que sea de utilidad para el neurocirujano durante la intervención.

- Reconstrucción de la superficie cerebral del paciente a partir de imágenes extraídas del microscopio quirúrgico.

Esta es la primera etapa del proceso de compensación del *brain-shift* utilizando el microscopio. Las imágenes serán extraídas de un par de cámaras que registran lo visto por los oculares del microscopio.

1.4. Resumen del Proyecto

Se desarrolló e implementó un sistema que permite agregar realidad aumentada a los microscopios que se utilizan en neurocirugía. El esquema del mismo se muestra en la figura 7.1.

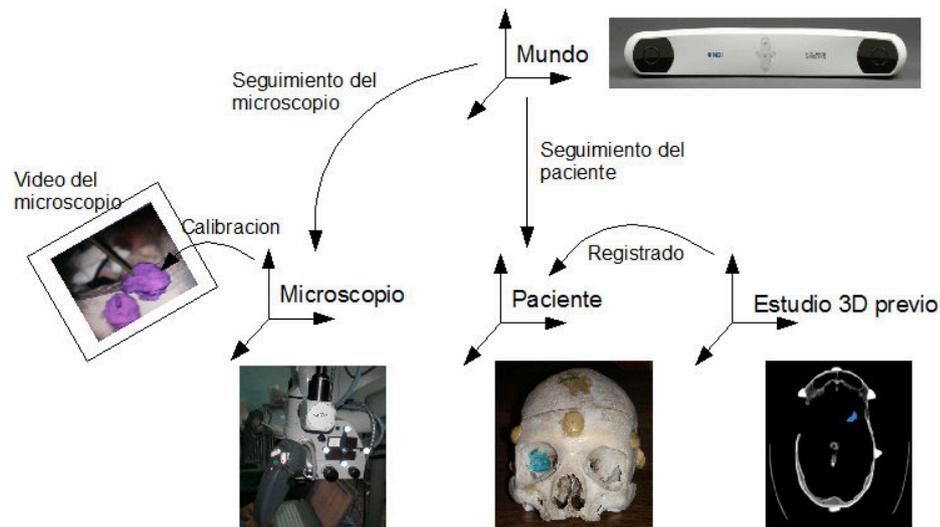


Figura 1.1: Esquema del sistema de realidad aumentada en el microscopio quirúrgico

El problema de agregar información virtual a lo visto por un microscopio quirúrgico puede ser escindido en los siguientes:

- Adquisición de la señal de video de la escena vista en un ocular
- Calibración de la óptica del microscopio

Durante el proceso de calibración se extraen los parámetros intrínsecos y extrínsecos del microscopio. Los parámetros intrínsecos (A) determinan la forma en que un punto del espacio es proyectado en la imagen del microscopio. Los parámetros extrínsecos (H) ubican el centro óptico de la cámara del microscopio respecto a un sistema de referencia solidario al microscopio.

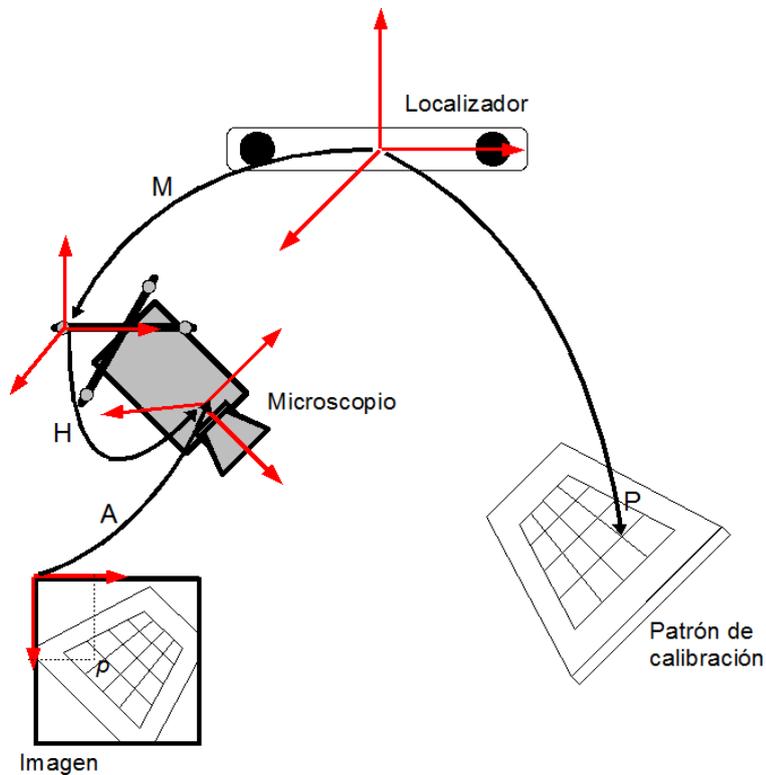


Figura 1.2: Calibración del microscopio

Para calibrar el sistema se utilizó un patrón con forma de damero del cual se puede detectar automáticamente la posición de las esquinas. Variando la posición relativa entre el microscopio y el patrón se adquieren fotos del patrón y en simultáneo, se extraen las coordenadas espaciales de las esquinas utilizando el localizador Polaris Spectra. A partir de las correspondencia entre los puntos espaciales y los puntos en la imagen, se extraen los parámetros del modelo pinhole de la cámara utilizando el algoritmo de Bouguet [7].

- Seguimiento del paciente y del microscopio

Para actualizar en tiempo real la información brindada al cirujano es fundamental una precisa y continua localización de las herramientas que participan en la escena. En particular para el microscopio, mediante la calibración se relaciona el sistema de coordenadas solidario al microscopio con el centro óptico del mismo; sin embargo esto no es suficiente ya que el microscopio estará en movimiento durante la intervención.

Para conocer la posición espacial de las distintas herramientas se utilizó el localizador Polaris Spectra. La comunicación con el Polaris y el cálculo de las transformaciones que relacionan a los distintos elementos de la escena se realizó utilizando la librería IGSTK (Image Guided Surgery Toolkit) [35].

- Registrado del estudio pre operatorio con la escena

La información que se agrega virtualmente al video es extraída de estudios preoperatorios. El estudio se encuentra referido a un sistema de coordenadas que no coincide con el sistema de referencia del paciente en el momento de la intervención. Si bien el estudio es realizado antes de la intervención, la rigidez del cráneo permite suponer que la posición relativa de las distintas estructuras que aparecen en el estudio se mantiene, es decir, se puede alinear la información obtenida a partir de los estudios antes de la operación con el paciente mediante una simple transformación rígida. Este mapeo entre el estudio del paciente y la ubicación del mismo en la sala de operaciones es lo que se conoce como registrado. Para realizar las pruebas se utilizaron estudios del cráneo que se muestra en la Figura 1.3



Figura 1.3: Cráneo utilizado para realizar las pruebas

- Visualización de la información

Se añadieron virtualmente a la señal de video extraída del microscopio modelos tridimensionales de estructuras anatómicas generadas a partir de estudios tomográficos. La Figura 1.4 muestra el cráneo utilizado para las pruebas y la señal de video a la cual se le superpusieron las estructuras anatómicas obtenidas a partir del estudio. En el video también se observa el modelo de la herramienta que está siendo utilizada como puntero.

Además de desarrollar e implementar el sistema se integró al microscopio como una herramienta más de guía en el neuronavegador que está siendo desarrollado por el IIE.



Figura 1.4: Visualización de la información en el sistema de Realidad Aumentada

Reconstrucción de una superficie cerebral

En lo que respecta a la reconstrucción de la superficie cerebral se implementó una plataforma capaz de realizar la misma. Además de realizarse la reconstrucción estéreo, durante el transcurso del proyecto se vio la posibilidad de reconstruir una superficie utilizando una sola cámara y el localizador Polaris. El diagrama de bloques de módulos de reconstrucción estéreo se muestra en la figura 1.5

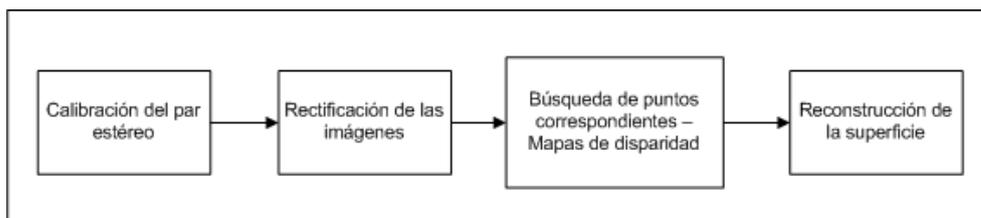


Figura 1.5: Diagrama de pasos para realizar reconstrucción estéreo

A continuación se describen los módulos:

- Calibración del par estéreo

Se determinó la posición relativa entre las cámaras y la forma en que éstas proyectan los puntos del espacio en la imagen. Esta posición relativa se compone de una traslación y una rotación que ubica a la cámara derecha respecto de la cámara izquierda; la rotación y traslación vinculadas se observan

en la Figura 1.6. Se implementaron dos formas de realizar la calibración estéreo, una de ellas fue utilizando las librerías de OpenCV [59] que tienen implementado el algoritmo de calibración estéreo de Bouget [7]. Esta forma de calibrar fue útil cuando se realizó estéreo con dos cámaras, pero por problemas que luego se describen se resolvió implementar la reconstrucción estéreo con una sola cámara. En el caso de reconstrucción utilizando una sola cámara se utilizó el localizador Polaris para determinar la posición relativa entre ambas imágenes.

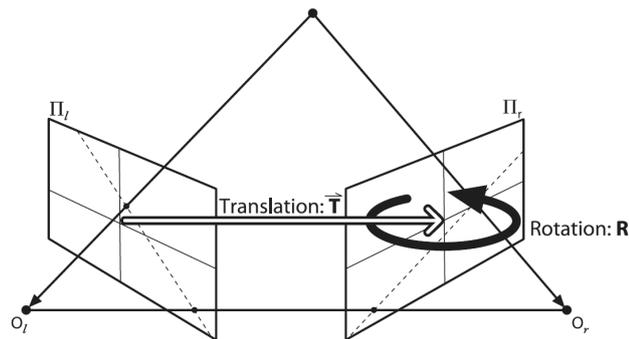


Figura 1.6: Rotación y traslación relativas entre las cámaras - Imagen extraída de [23]

■ Búsqueda de puntos correspondientes

Para recuperar la posición espacial de un punto a partir de dos imágenes es necesario conocer su proyección en ambas imágenes. Para facilitar la búsqueda de correspondientes suele ser conveniente rectificar las imágenes; de esta forma se restringe la búsqueda de correspondientes a puntos que tienen la misma coordenada en y . Rectificar las imágenes es proyectar las mismas como si las cámaras estuvieran con sus ejes ópticos alineados y sus filas estén alineadas; para realizar la rectificación es necesario conocer previamente la posición relativa entre las dos cámaras, es decir conocer la calibración del par estéreo. En la Figura 1.7 se observa cómo quedan las cámaras para que las imágenes queden rectificadas mientras que en la Figura 1.8 se observa el resultado del proceso de rectificación de imágenes.

A partir las imágenes rectificadas es posible calcular mapas de disparidad que brindan información de la profundidad de un punto, recuperándose de esta manera la información perdida en el proceso de formación de la imagen. Para el cálculo de la disparidad es necesario poner en correspondencia los puntos de ambas imágenes. Los algoritmos utilizados para la búsqueda de correspondientes fueron Block Matching y Graph Cut, ambos implementados en OpenCV; en la Figura 1.9 se ve cómo queda un mapa de disparidad.

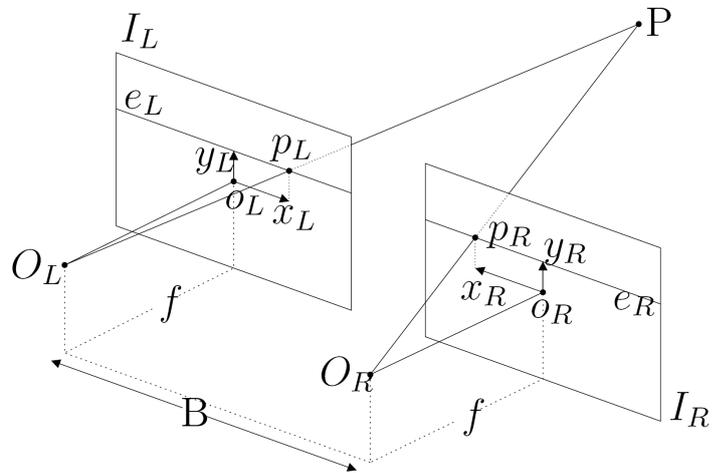


Figura 1.7: Ubicación de las cámaras de un par estéreo para que sus imágenes estén rectificadas - Imágenes extraídas de [40]

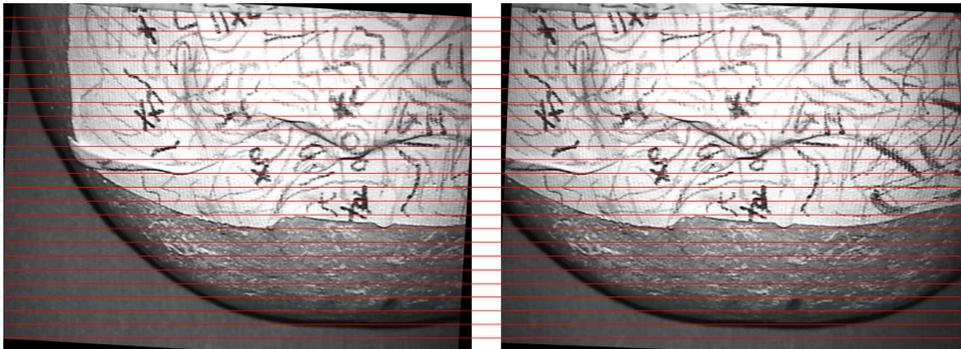


Figura 1.8: Imágenes rectificadas

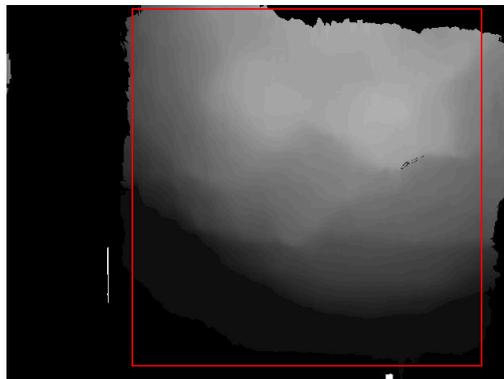


Figura 1.9: Mapa de disparidad obtenido a partir de imágenes adquiridas con el microscopio del Hospital de Tacuarembó

- **Reconstrucción de superficie**

Una vez calculados los mapas de disparidad, se puede realizar la reconstrucción de la escena. Para esto es necesario realizar la triangulación de cada punto de las imágenes para los que se conoce su correspondiente y también conocer la geometría estéreo del par (calibración). En la Figura 1.10 se muestra el resultado de la reconstrucción de una superficie.

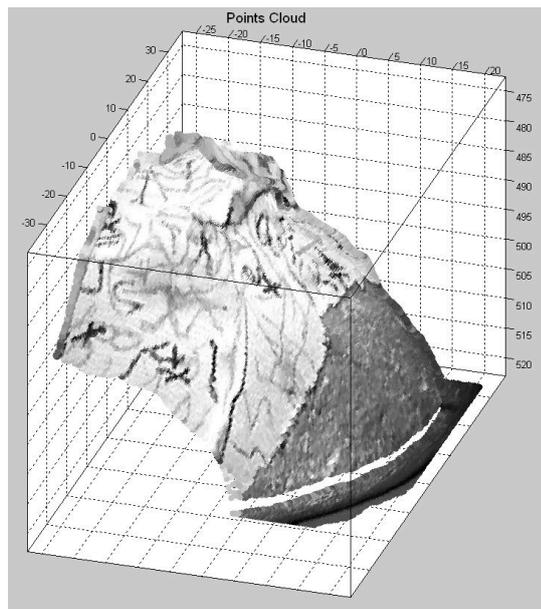


Figura 1.10: Superficie 3D reconstruida a partir de imágenes tomadas con el microscopio del Hospital de Tacuarembó

1.5. Organización de la documentación

Capítulo 2 - Cirugía Guiada por imágenes Se brinda un panorama global de las tareas que involucra desarrollar un sistema de guiado mediante imágenes. Se repasan los diferentes métodos que se han utilizado para localizar los instrumentos quirúrgicos en la sala de operaciones, las técnicas de registro del paciente con el estudio y las distintas maneras de visualizar la información que se emplean.

Capítulo 3 - Componentes del Sistema Se muestran los elementos de hardware que se utilizaron en este proyecto.

Capítulo 4 - Calibración Se ofrece el marco teórico del tema. Se describe el modelo de cámara y los algoritmos utilizados para encontrar los parámetros de dicho modelo.

Capítulo 5 - Implementación de la Calibración Se presenta la aplicación de los algoritmos de calibración en el caso del microscopio quirúrgico. Se analizan las particularidades del caso y se muestran los resultados obtenidos.

Capítulo 6 - Realidad Aumentada Se detalla el proceso que se debe seguir si se quiere añadir virtualmente información a una escena real. Se desarrolla el método mediante el cual se ubica a la cámara de la escena virtual en la misma posición y con el mismo lente que la real.

Capítulo 7 - Microscopio Quirúrgico - Una aplicación de Realidad Aumentada Se muestra una de las aplicaciones de la realidad aumentada: asistir al neurocirujano durante una intervención quirúrgica. Se detalla el sistema y procedimiento desarrollado. Se muestran los resultados obtenidos.

Capítulo 8 - Reconstrucción Estéreo Se resumen los principales conceptos en lo que respecta al tema de reconstrucción estéreo, comenzando con una breve introducción a los conceptos básicos de la geometría estéreo. Luego se detalla paso a paso el proceso de reconstrucción siguiendo lo que se puede llamar la secuencia “natural” de pasos que deben darse, comenzando por la calibración de las cámaras, rectificación, búsqueda de correspondientes hasta llegar a la triangulación. Por último se hace un pequeño análisis de las características de los dos algoritmos utilizados para la búsqueda de puntos correspondientes en los pares estéreo rectificados.

Capítulo 9 - Implementación Estéreo Análisis de la evolución del proyecto estéreo y de los dispositivos que se utilizaron para resolver los distintos problemas. También se muestran resultados de laboratorio y en el Hospital de Tacuarembó con lo que es el último dispositivo implementado, con el que se obtuvieron los mejores resultados. Por último se presenta una reseña de las técnicas utilizadas para visualización de mallas.

Capítulo 10 - IGSTK - Software para cirugía guiada por imágenes Se describe la biblioteca de uso médico IGSTK (Image Guided Surgery Toolkit). Se muestra cómo se realizó la integración del módulo del microscopio con el neuronavegador que está desarrollando el GTI del IIE.

Capítulo 2

Cirugía Guiada Por Imágenes

2.1. Introducción

Los procedimientos quirúrgicos han evolucionado considerablemente en los últimos veinte años. Un factor que ha influido notoriamente en este avance es el desarrollo de las diferentes modalidades de imagenología médica, desde los rayos X, pasando por Tomografías Computadas (TC) y Resonancias Magnéticas (RM) o las más recientes técnicas de medicina nuclear, como Single Positron Emission Tomography (SPECT) y Positron Emission Tomography (PET) por citar algunas. El uso de estas imágenes ha contribuido a mejorar la seguridad y efectividad de los diferentes procedimientos. Las imágenes generadas mediante estos procesos, dependiendo de la modalidad, brindan una descripción detallada de la anatomía del paciente (TC, RM, rayos X) o de la actividad metabólica de una cierta zona del cuerpo (PET, SPECT). Al aportar información complementaria, se pueden combinar mediante técnicas de registro. Esta información es utilizada por el neurocirujano, haciendo que la cirugía sea más segura, menos invasiva y más efectiva.

Para utilizar la información que aparece en las imágenes médicas durante una intervención es necesario que el cirujano realice un mapeo mental que asocie las estructuras que observó en los estudios (en algunos cortes) con las que observa en el paciente durante la intervención. Esto implica realizar una reconstrucción mental de las estructuras observadas en los cortes, lo que depende en gran medida de la experiencia con la que cuenta el cirujano.

La cirugía guiada por imágenes [50, 51, 75] se basa en determinar las coordenadas de los instrumentos de cirugía en la mesa de operaciones y mostrar su posición en las imágenes de estudios previos para guiar al cirujano en zonas de difícil visibilidad o acceso. Brinda una realimentación visual de la posición del instrumento y la trayectoria que se está siguiendo, lo que favorece y simplifica la toma de decisiones.

Para lograr este objetivo se suele dividir el proceso en cinco subprocesos [51]:

1. Adquisición de la información preoperatoria, en alguna de sus modalidades.

2. Localización y seguimiento de la posición de los instrumentos quirúrgicos y/o de los dispositivos médicos.
3. Registrado del espacio quirúrgico con la información del estudio.
4. Visualización de la posición de los instrumentos de cirugía con respecto a las estructuras de interés presentes en el estudio.
5. Estimación de las diferencias entre la información presente en el estudio previo y las observadas durante la operación.

2.2. Adquisición de información previo a la operación

Antes de la operación se debe realizar al menos un estudio al paciente. En caso de realizarse más de uno es necesario integrar las distintas informaciones mediante registrado. Una vez que se tiene toda la información referida a un mismo sistema de coordenadas se comienza con la planificación de la operación. Durante la planificación se segmentan estructuras de interés y se generan modelos de superficie 3D de las mismas.

Los métodos de segmentación varían desde aquellos que son completamente manuales hasta los automáticos. El método más preciso es el manual aunque es también el que mayor tiempo insume. Las regiones de interés son marcadas, en cada corte, por un experto. Existen una multiplicidad de algoritmos tendientes a reducir el tiempo de la segmentación manual. Los más básicos se basan en métodos de umbralización de nivel de gris y morfología matemática. Permiten, por ejemplo, discriminar un tejido a partir de un nivel de gris (por ejemplo, piel) o segmentar una estructura encontrando todos aquellos puntos de una región conexas que comparten alguna característica.

En esta etapa es posible, por ejemplo, colorear alguna estructura de interés para el neurocirujano de forma de resaltar su ubicación en los distintos cortes del estudio. Además se pueden realizar anotaciones en los cortes, indicando por ejemplo trayectorias a seguir, puntos de acceso o de localización de lesiones.

2.3. Localización y seguimiento de instrumentos

La precisa localización y seguimiento de los instrumentos que pertenecen a la escena operatoria es de fundamental importancia. A lo largo de los años se han utilizado diferentes principios físicos para determinar la posición de los distintos instrumentos. A continuación se presenta brevemente algunos de los métodos utilizados y sus principales características.

En 1985 Roberts et al.[56] utilizaron un transductor de ultrasonido para seguir un microscopio y mostrar datos extraídos de estudios tomográficos junto con la escena. Para seguir el microscopio se colocaron en el mismo receptores de ultrasonido en posiciones conocidas. Una fuente *s* emitía pulsos de sonido a una frecuencia mayor

a la del rango audible y a partir del tiempo que demoraba la señal en viajar desde el emisor a los receptores se deducía la posición de los marcadores.

Los sistemas de posicionamiento que basan su funcionamiento en el sonido sufren de dos problemas fundamentales. El primero es que si bien la la velocidad del sonido a niveles de presiones y temperatura estándares es $330m/s$, éste valor se puede alterar al variar la temperatura y la humedad, causando de esta manera errores en la localización de los receptores. Para mitigar este problema se podría utilizar más de un emisor, liberando de esta manera la suposición de que la velocidad del sonido es constante. Esto conduce a un segundo problema, la separación de tiempo que debe existir entre la emisión de los pulsos. Entre la emisión de un emisor y otro se debe aguardar un intervalo de tiempo lo suficientemente largo como para que el eco de uno no se confunda con la emisión del siguiente. Esto hace que no se pueda seguir a las herramientas de manera “continua”.

Otro de los principios físicos que se puede utilizar para la construcción de dispositivos de seguimientos es el óptico. Una de las posibilidades es utilizar diodos emisores de luz infrarroja (IREDs por sus siglas en inglés). La luz emitida por estos diodos es capturada por un sensor CCD (Charge Coupled Device). Colocando un filtro de luz visible frente al sensor, se pueden distinguir los IREDs de otras fuentes de luz presentes en la sala. Basados en este principio surgieron dos sistemas comerciales, el Optotrak 3020 (Nothern Digital Inc, Waterloo, ON, Canada) [2] y el Flashpoint 3D [63]. Ambos sistemas fueron utilizados en dispositivos de guiado por imágenes [62, 41].

Otra posibilidad es la utilización de dos cámaras y algún patrón conocido, cuyas características sean fácilmente extraíbles de las imágenes. Dentro de estos sistemas se puede destacar el proyecto VISLAN desarrollado por el GUY's Hospital [18] [43] o más recientemente el localizador comercial Micron Tracker de la firma Claron Technology [33]. La mayor ventaja de estos sistemas es el bajo costo ya que los patrones que se utilizan para determinar la ubicación espacial de los objetos suelen ser dibujos impresos en papel. El mayor problema que presentan es que, al trabajar con luz visible, el ambiente puede hacer que el sistema se confunda y no pueda detectar perfectamente las características del patrón introduciendo así un error en la localización de los objetos. En un intento por mantener las ventajas de los sistemas pasivos y fortalecer sus debilidades es que Nothern Digital ha desarrollado una versión de Polaris que usa unas esferas reflectoras de infrarrojo para poder seguir las distintas herramientas.

Todos los sistemas ópticos, ya sea activos o pasivos, necesitan contar con línea de vista entre el dispositivo emisor y el o los receptores. Esto, junto a la imposibilidad de utilizarlo en herramientas flexibles, son las principales debilidades de los sistemas ópticos.

Algunos investigadores han utilizado sistemas electromagnéticos para la localización de las herramientas [76]. Un ejemplo de localizador electromagnético es el Aurora, desarrollado por Nothern Digital [13]. La principal ventaja de estos sistemas es que la señal electromagnética puede viajar a través del cuerpo por lo que se puede colocar un sensor en la punta de una herramienta que ingresa al cuerpo.

Las principales desventajas son la precisión, que aún no alcanza la de los dispositivos ópticos y la interferencia que se genera con dispositivos médicos presentes en la sala de operaciones.

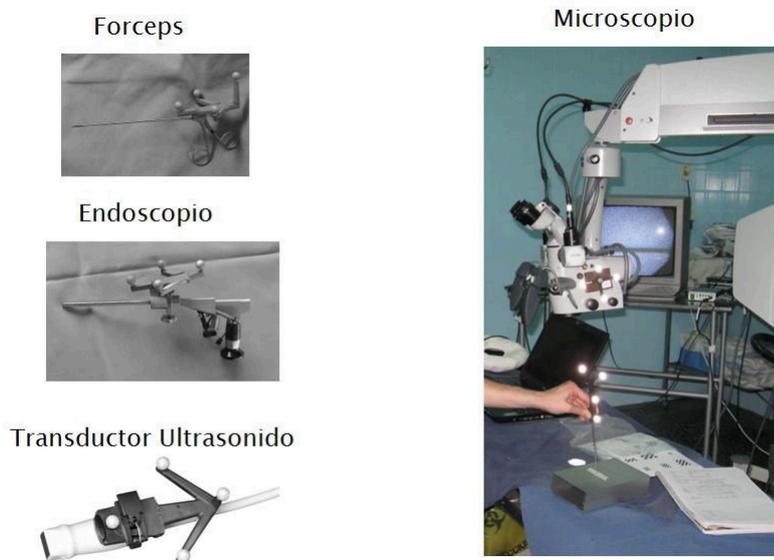


Figura 2.1: Algunos de los instrumentos quirúrgicos que puede interesar seguir durante una intervención

2.4. Técnicas de Registrado

El desarrollo de las diferentes modalidades de imagenología médica ha brindado información tridimensional de la anatomía del paciente, y en particular de la localización y la extensión de las lesiones. Por otro lado, los dispositivos localizadores o *trackers* han permitido desarrollar técnicas que permiten poner en correspondencia el espacio del estudio con el de la sala de operación. Este proceso es lo que se conoce como registrado del espacio quirúrgico.

Cuando se conoce la relación matemática que vincula un punto referido al sistema de coordenadas del estudio con el punto correspondiente referido al sistema de coordenadas de la sala de operaciones se dice que los espacios están registrados. Si la relación se puede reducir a una simple rotación y traslación se dice que el registrado es rígido.

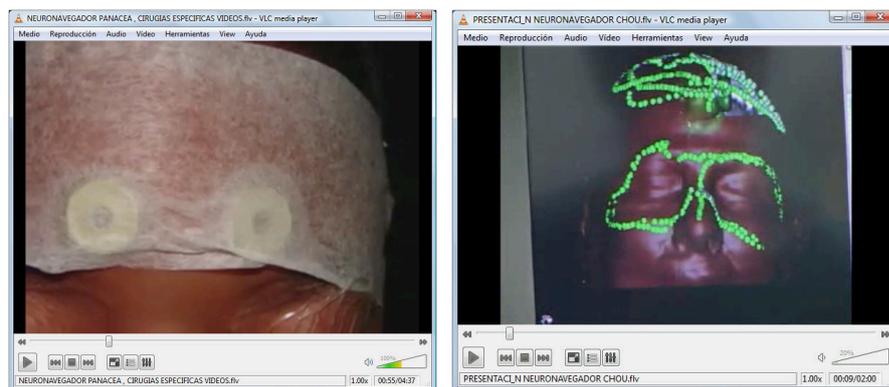
La rigidez del cráneo permite suponer de manera razonable que las estructuras localizadas en el estudio antes de la operación no sufrieron deformaciones, al menos en las primeras etapas de la intervención. Esta suposición simplifica enormemente la puesta en correspondencia entre las coordenadas de un punto referido al estudio y referido a la sala de operaciones.

El método obvio de registrado consiste en identificar puntos correspondientes

del espacio del estudio y del espacio físico. A partir de estos pares de puntos y de técnicas como la desarrollada por Arun y Horn [5, 31] de minimización del error (en el sentido de mínimos cuadrados) en la transformación que lleva un conjunto de puntos en el otro se puede rápidamente registrar los espacios. El principal inconveniente de este método es el de poder identificar de manera precisa puntos correspondientes en el estudio y en la sala de operaciones, ya que características fácilmente identificables y puntualmente localizables en un espacio pueden no serlo en el otro. Los puntos presentes en los estudios suelen tener además una incertidumbre asociada al grosor del corte.

Para establecer las correspondencias se pueden utilizar estructuras del cuerpo conocidas, que en general van a presentar las dificultades mencionadas anteriormente, o se pueden utilizar objetos no pertenecientes a la anatomía, especialmente diseñados para facilitar la puesta en correspondencia, y que pueden colocarse dentro o en la superficie del cuerpo. Estos objetos, que se utilizan como marcadores, son los habitualmente llamados fiduciales. El que se conozcan exactamente el tamaño y las dimensiones de los fiduciales y el que sean diseñados para que sean visibles en más de un corte del estudio hace que se reduzca la incertidumbre en la localización de los puntos correspondientes. Un tratamiento matemático sobre el registrado utilizando marcadores y el diseño de los fiduciales puede verse en [44].

En los últimos años se han desarrollado técnicas de registrado que llevan nubes de puntos en nubes de puntos. Se identifican puntos en la superficie de la piel (tocando con un puntero, o con un láser) y en el estudio (segmentando a partir del nivel de gris de la piel) y se halla la transformación que lleva una nube en la otra [54]. La gran ventaja que presenta esta técnica es que no es necesario colocar fiduciales en el paciente antes de realizar el estudio, con las molestias que esto genera. Aún no han logrado la precisión de los sistemas basados en fiduciales implantados en los huesos pero la precisión es comparable con la de fiduciales adheridos a la piel.



(a) Fiduciales adheridos a la piel del paciente (b) Puntos marcados con el puntero durante el registrado por superficie

Figura 2.2: Métodos de registrado

2.5. Visualización

El propósito de la visualización en los sistemas de guiado mediante imágenes es representar de forma útil y confiable al paciente y a los instrumentos quirúrgicos de manera de guiar al neurocirujano durante la intervención. Se podría decir que en la visualización es donde se integran las múltiples y relacionadas fuentes de información. Allí se combinan los datos obtenidos durante los estudios previos con la información de posicionamiento de los instrumentos quirúrgicos que se obtienen durante la intervención. La integración y el adecuado manejo de la información es en sí mismo un desafío. Una adecuada representación de la información puede ser la diferencia entre el éxito o fracaso de la aplicación, siendo tan o más importante que la precisión lograda por todo el sistema.

Es importante actualizar la información a una tasa que permita hacer un buen uso de ella y además es fundamental la sincronización que exista al desplegar las distintas fuentes de información. Asimismo es necesario advertir al profesional, mediante alguna señal, en caso de que la información que se esté mostrando no sea actual.

2.5.1. Representación del estudio

Una representación 2D de los cortes no requiere demasiados recursos computacionales y es por lo tanto la forma más habitual de representar la información. La principal ventaja que presenta la representación en 2D es la baja latencia con que se muestran los cortes. Las representaciones en 3D y 4D requieren de mayores recursos computacionales por lo que el período de actualización de la información suele ser mayor.

Representaciones 2D: Axial, Sagital y Coronal Si se toma al cuerpo como referencia, en anatomía se definen las siguiente direcciones:

- Anterior: delante del cuerpo
- Posterior: atrás del cuerpo
- Superior: hacia la cabeza
- Inferior: hacia los pies
- Izquierda: hacia la izquierda
- Derecha: hacia la derecha

En el formato DICOM, por ejemplo, que es el estándar más aceptado en la actualidad, las coordenadas 3D de un estudio se representan de la siguiente manera:

La dimensión más baja, por ejemplo x , al variar los cortes crece de derecha a izquierda. La dimensión y crece de anterior a posterior y la dimensión z de inferior a superior. A esta forma de orientar los datos se le llama sistema Left Posterior Superior (LPS).

La manera estándar de representar un estudio médico es mediante tres cortes ortogonales, a saber: axial, sagital y coronal. El corte axial es perpendicular al eje inferior-superior, el plano sagital es perpendicular al eje izquierda-derecha y el plano coronal es perpendicular al eje anterior-posterior.

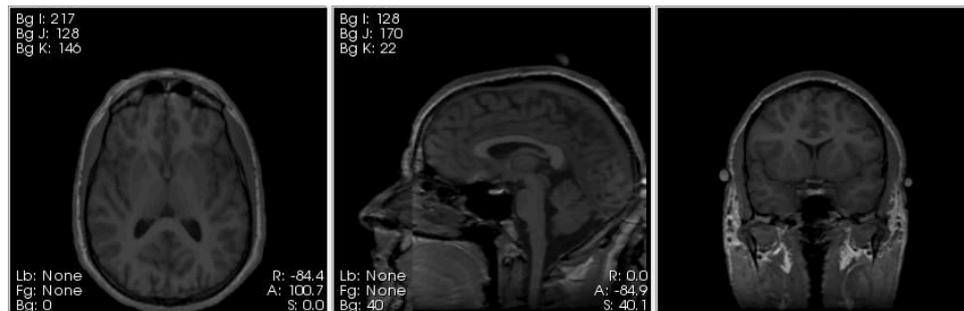


Figura 2.3: Representación 2D del estudio: cortes axial, sagital y coronal. Imagen extraída del visualizador Slicer [45].

Representación 3D Se suelen mostrar los cortes axial, sagital y coronal que pasan por el punto espacial de interés. Además de los tres cortes se puede mostrar alguna estructura previamente segmentada.

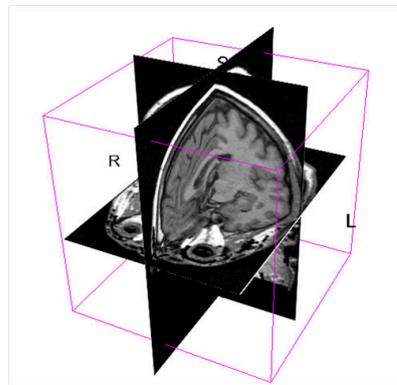


Figura 2.4: Representación 3D del estudio en el visualizador Slicer

2.5.2. Visualización en un sistema de guiado por imágenes

La figura 2.5 muestra la forma típica en que se combinan las diversas fuentes de información en un sistema de guiado por imágenes. Se observan los cortes Axial, Sagital y Coronal y dentro de ellos se aprecian:

En amarillo: la proyección de la herramienta del cirujano en cada uno de los cortes.

En azul: la proyección del corte coronal en los cortes axial y sagital.

En verde: la proyección del corte sagital en los cortes axial y coronal.

En rojo: la proyección del corte axial en los cortes sagital y coronal.

En la representación 3D se ven los tres cortes que pasan por la posición de la herramienta del cirujano y en azul se observa un modelo virtual de la herramienta.

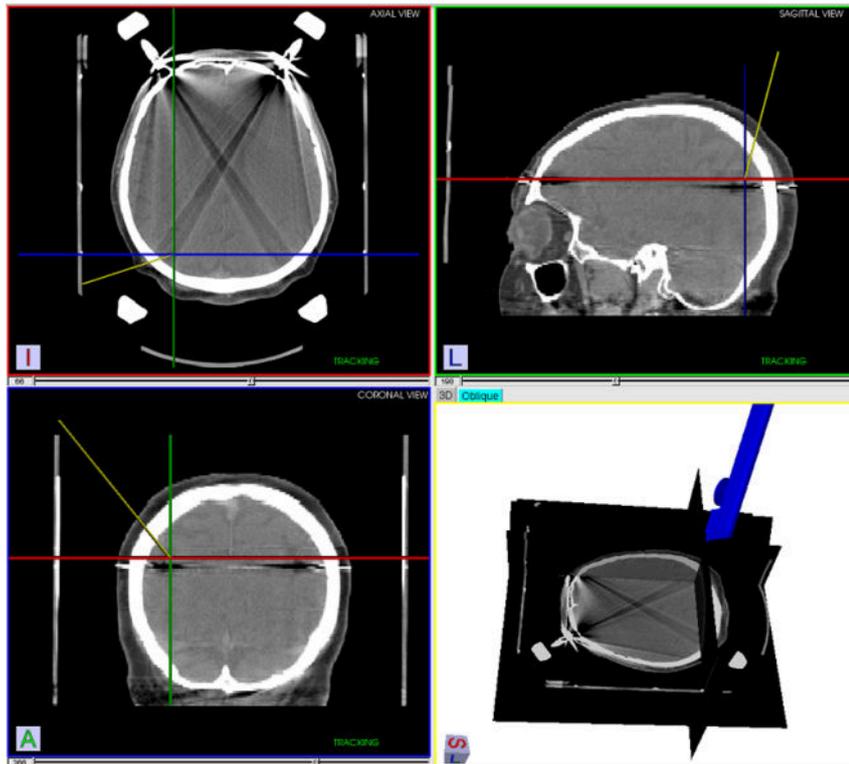


Figura 2.5: Visualización de la información en IGSTK

2.6. Actualización de información

Los estudios previos son una fuente precisa de información; sin embargo, durante la intervención el cerebro sufre deformaciones del entorno de los 5-10 mm [51]. La cuantificación de estos desplazamientos se puede ver en [19, 28]. Las causas de este desplazamiento del cerebro son la liberación de fluido cerebro espinal al abrir la duramadre, la respuesta fisiológica ante la anestesia, la manipulación del tejido cerebral que realiza el cirujano y principalmente la acción de la gravedad [55].

Estas deformaciones provocan que no sea válida la suposición inicial de que el cerebro es una estructura rígida y por lo tanto el registrado que se realizó al comienzo de la intervención deja de ser válido. Para seguir guiando al neurocirujano es necesario actualizar la información preoperatoria de alguna manera.

La teoría de la elasticidad provee de una serie de ecuaciones derivadas parciales que describen las condiciones de equilibrio de un cuerpo en función de variables como la presión interna, la gravedad y la deformación interna. Basado en esta teoría se construye un modelo del cerebro compuesto por un conjunto de resortes con ciertas propiedades biomecánicas [51].

Mediante la adquisición de imágenes intraoperatorias (no tan precisas como las que se realizan antes de la operación) es posible actualizar este modelo y corregir así la información obtenida durante la operación y por lo tanto la que se brinda al neurocirujano.

Existen diversos métodos para adquirir información sobre la estructura cerebral durante la intervención. La forma más difundida en la actualidad es la utilización de ultrasonido, que si bien no es tan precisa como la realización de estudios de tomografía o resonancia intraoperatorios, presenta dos grandes ventajas. La primera es el costo, que es sensiblemente menor, y la segunda es el espacio que ocupa en la sala de operaciones un transductor de ultrasonido. A modo de ejemplo, si a Uruguay le regalaran un tomógrafo o un resonador para realizar estudios durante la operación no habría ninguna sala de operaciones capaz de ingresarlo.

Otra de las técnicas posibles es utilizar el microscopio para extraer la ubicación de la superficie del cerebro visible por el mismo y utilizar esta información para actualizar el modelo del cerebro. La principal ventaja de este método es que si se cuenta con un microscopio no es necesario realizar ninguna inversión extra. Con respecto a la utilización de ultrasonido, presenta como desventaja que sólo es posible utilizar la posición de la superficie del cerebro para corregir el modelo.

Si se modela correctamente el cerebro y se adapta dicho modelo a partir de imágenes obtenidas durante la intervención se pueden reducir considerablemente los errores debido a las deformaciones.

Capítulo 3

Descripción de los equipos

3.1. Introducción

En los capítulos anteriores se describió el marco y los objetivos principales del proyecto. A modo de resumen, el trabajo consistió en implementar un sistema que agregue realidad aumentada a un microscopio quirúrgico, así como reconstruir la superficie cerebral a partir de la señal de video proveniente del microscopio. En concreto, se trabajó en las instalaciones del Hospital de Tacuarembó, Hospital de Clínicas y Hospital Maciel, utilizando los microscopios quirúrgicos con los que cuentan dichos hospitales.

La Figura 3.1 muestra la disposición de los equipos durante el trabajo en la sala de operaciones. En el presente capítulo se describirá brevemente la función y las características de cada uno de los componentes fundamentales del sistema, a saber:

- Sistema localizador y sus herramientas
- Microscopio quirúrgico
- Cámaras de video y adaptadores
- Tarjetas capturadoras de video

Sin embargo, los algoritmos implementados fueron desarrollados fundamentalmente en el laboratorio, destinando las visitas a los Hospitales principalmente a la evaluación y validación del trabajo. Por esto resulta importante también describir los elementos utilizados en el laboratorio.



Figura 3.1: Disposición de los equipos en la sala de operaciones.

3.2. Descripción de los equipos en la sala de operaciones

3.2.1. Localizador Polaris Spectra

El localizador Polaris Spectra utilizado es un sistema de tipo óptico pasivo perteneciente a la familia Polaris, de Northern Digital Inc. Este equipo es capaz de medir la posición 3D de marcadores pasivos adheridos a herramientas específicas de la aplicación, y por consiguiente determinar la posición y orientación de estas herramientas dentro de cierto volumen de trabajo.



Figura 3.2: Vista del frente del localizador Polaris Spectra. Imagen extraída de [2]

Sensor de posición

Constituye el componente principal del equipo, y su funcionamiento puede resumirse como se describe a continuación. El sistema cuenta con dos sensores de posición formados por una lente y un CCD, y cada uno de ellos está rodeado por un arreglo de IREDs (Infra Red Emitting Diode) [69], que emiten luz infrarroja sobre todo el volumen de trabajo. Dentro de este volumen se encuentran las herramientas con sus marcadores pasivos, los cuales tienen la propiedad de reflejar la luz proveniente de los IREDs. La luz reflejada es recibida por los dos sensores, y a partir de esta información, el sistema es capaz de calcular la posición 3D de cada

marcador mediante triangulación.

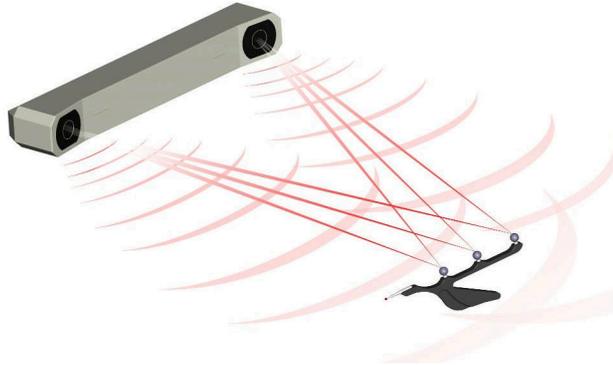


Figura 3.3: Principio de funcionamiento del Polaris Spectra. Imagen extraída de [2]

La información de posición de los marcadores es luego transmitida a través de una interfaz USB, junto con información del estado del sistema, de manera tal que si el localizador se conecta a una computadora esta información puede ser recogida y almacenada, desplegada o tratada de alguna forma. Las coordenadas 3D y transformaciones estarán referidas a un sistema de coordenadas asociado al localizador, tal como se muestra en la Figura 3.4.

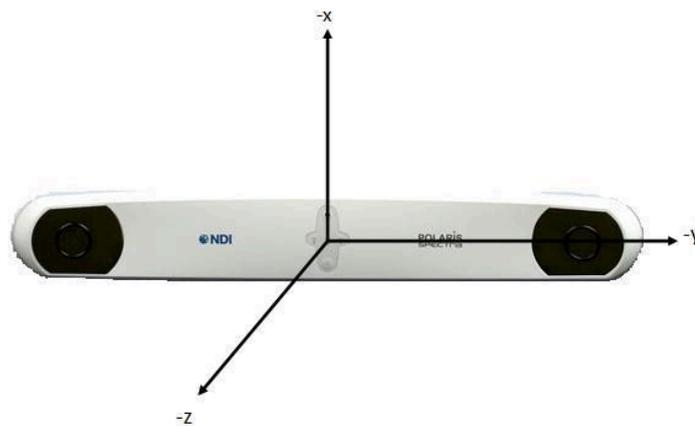


Figura 3.4: Sistema de coordenadas asociado al localizador.

Como se mencionó, asociado al localizador existe un volumen de trabajo en forma de pirámide, dentro del cual deben ubicarse las herramientas para ser visibles por el equipo. Sus dimensiones aproximadas pueden consultarse en la Figura 3.5. Dentro de la región identificada como Pyramid Volume, el fabricante declara una precisión de 0.25 mm RMS con un intervalo de confianza del 95 % de 0.5 mm, mientras que dentro del volumen Extended Pyramid Volume la precisión es de 0.30

mm RMS con un intervalo de confianza del 95 % de 0.6 mm.¹ Estos resultados están basados en la detección de un único marcador situado en más de 900 posiciones dentro del volumen de trabajo, tomando un promedio de 30 muestras en cada posición.

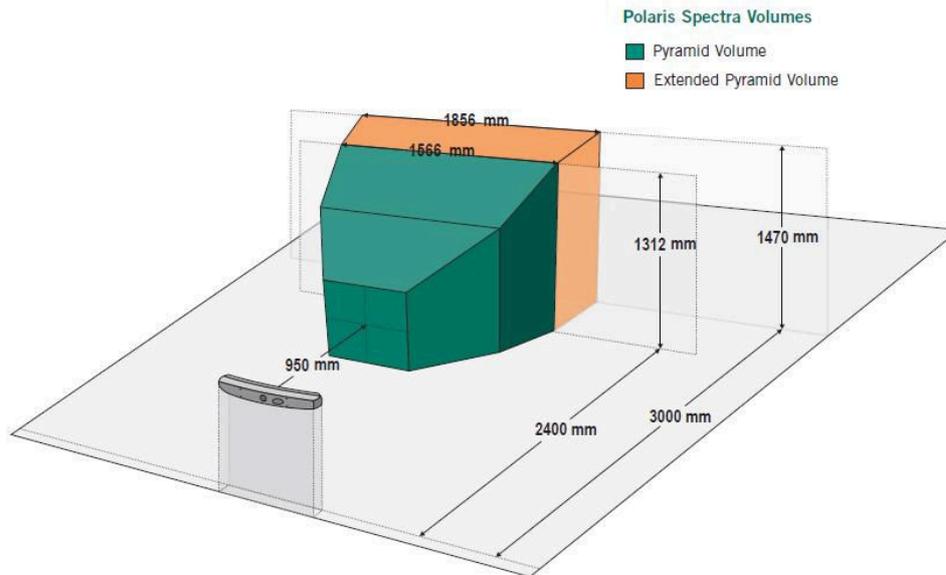


Figura 3.5: Dimensiones del volumen de trabajo del localizador. Imagen extraída de [2]

Herramientas y marcadores

El localizador viene acompañado de tres herramientas con marcadores pasivos, dos de ellas en forma de cruz y una tipo puntero. Cada herramienta es un cuerpo rígido que incorpora 4 marcadores. Básicamente estos son esferas de 11.5 mm de diámetro cubiertas por un material retrorreflectivo. Están ubicados en la herramienta en posiciones fijas, sin movimiento entre ellos, y su geometría en cada una de las herramientas es única, lo que hace posible que el localizador identifique cada una de ellas y pueda observar y seguir más de una a la vez. En la Figura 3.6 se muestran las herramientas junto con el resto de los accesorios que acompañan al localizador:

- 25 marcadores pasivos
- dos pinzas ajustables para fijar las herramientas a varillas cilíndricas de distintos diámetros
- dos llaves tipo Allen para el montaje de las herramientas sobre las pinzas y de las pinzas sobre la varilla

¹El valor de precisión dado para la región Extended Pyramid Volume refiere al volumen completo.



Figura 3.6: Herramientas y accesorios del localizador.

3.2.2. Microscopios

Fabricados por la firma Carl Zeiss [1], los modelos de los microscopios quirúrgicos de los que se dispone en los Hospitales y con los cuales se trabajó en el desarrollo del Proyecto son:

- Hospital Regional de Tacuarembó: Microscopio OPMI Vario sobre estativo de suelo S88
- Hospital Maciel: Microscopio OPMI CS sobre estativo de suelo S4
- Hospital de Clínicas: Microscopio OPMI Vario sobre estativo de suelo S8

OPMI Vario es un microscopio que cuenta con sistema de aumento motorizado con relación de 1:6 y sistema de enfoque interno motorizado de forma continua, con un margen de trabajo variable entre 200 y 415 mm. Incorpora una luz de Xenón de 180 W con carácter de luz diurna, y dos palancas de control con botones que permiten ajustar los parámetros del microscopio, además de liberar los frenos para colocarlo en la posición deseada. Tanto el estativo de suelo S88 en el caso del equipo del Hospital de Tacuarembó como el S8 en el Hospital de Clínicas incorporan un display con teclas que permiten desplegar y ajustar todos los parámetros del OPMI Vario.

El microscopio OPMI CS es un modelo anterior al OPMI Vario, que también cuenta con sistema de aumento motorizado con relación de 1:6, sistema de enfoque motorizado y luz de Xenón de 100 W. No dispone de las palancas de control, pero sí de un panel de mando con pedales para ajustar los parámetros. El ajuste de parámetros también puede hacerse desde los botones y perillas incorporadas en el estativo de suelo S4. A diferencia de los anteriores, éste no cuenta con display.

Los tres microscopios tienen la posibilidad de extraer la imagen de cada ocular por medio de cámaras de video que se acoplan al microscopio a través de adaptadores.



(a)



(b)



(c)

Figura 3.7: Microscopios quirúrgicos Carl Zeiss. 3.7(a) Microscopio OPMI Vario con estativo S8, Hospital de Clínicas. 3.7(b) Microscopio OPMI CS con estativo S4, Hospital Maciel. 3.7(c) Microscopio OPMI Vario con estativo S88, Hospital de Tacuarembó.

El Hospital de Tacuarembó cuenta con la posibilidad de utilizar dos adaptadores y cámaras MediLive originales de Carl Zeiss, como puede verse en la Figura 3.7(c). Generalmente tienen disponible solo un juego adaptador-cámara.

El adaptador disponible en el Hospital Maciel es de la marca D.F. Vasconcellos [57] y se utiliza junto con una cámara de la marca Video Domain. Finalmente, la cámara disponible en el Hospital de Clínicas es marca Rainbow CCTV y se utiliza con un adaptador de la marca D.F Vasconcellos.

3.2.3. Tarjetas adquisidoras

Para recoger el video proveniente de las cámaras de los microscopios y poder procesarlo posteriormente en la computadora, es necesario conectar la salida de video compuesto de las cámaras a una tarjeta capturadora de video. Cono este objetivo, las tarjetas utilizadas fueron:

- PCTV USB2, fabricada por Pinnacle [52].
- TV BOX 2.0, fabricante desconocido.

Ambas reciben la señal de video compuesto de las cámaras y tienen como salida la señal digitalizada a través de una interfaz USB mediante la cual se conectan a la computadora.



Figura 3.8: Tarjetas capturadoras de video.

3.3. Descripción de los equipos en el laboratorio

Durante el trabajo en el laboratorio los equipos utilizados fueron los mismos que en la sala de operaciones, a excepción del microscopio, que debió ser sustituido por cámaras. A continuación se presentan sus principales características técnicas.

3.3.1. Cámaras del laboratorio

Las cámaras de video que sustituyeron al microscopio en el laboratorio fueron:

- Cámara Sony modelo SSC-DC50P, más lente TV zoom lens 12.5 - 75 mm, F1.8.
- Cámara Sony modelo DXC-107P, más lente Sony TV zoom lens 12.5 - 75 mm, F1.8.

Las tablas siguientes resumen sus especificaciones:

	SSC-DC50AP	SSC-DC54AP
Image device:	1/2-inch Interline Transfer Exwave HAD CCD	
Picture elements:	752 (H) × 582 (V)	
Sensing area:	6.3 × 4.7 mm	
Signal system:	PAL standard	
Scanning system:	625 lines, 2:1 interlace	
Sync system:	Internal or external with VBS/VIS or MPX-VIS	Internal or external with
Phase control:	H/SC phase adjustment (H phase: ±0.12H, SC phase: 360° with 0/180° switch) V phase adjustment (±90°) for AC line lock (SSC-DC54AP/DC58AP only)	
Horizontal resolution:	470 TV lines	
Lens mount:	C/CS mount adjustable	
Minimum illumination:	AGC ON (TURBO mode) 0.4 lx at F1.2 (30 IRE) 0.8 lx at F1.2 (50 IRE) 3.0 lx at F1.2 (100 IRE)	
Aperture control:	SHARP/NORMAL switchable	
Automatic gain control (AGC):	TURBO/NORMAL/OFF switchable	
Electronic shutter:	1/50, 1/120, 1/250, 1/500, 1/1000, 1/2000, 1/4000, 1/10000 s	
CCD IRIS control:	ON/OFF switchable, 1/50 to 1/1000000 s	
White balance:	ATWpro/ATW/AWB (one-push)/5600 K switchable	
Backlight compensation:	BLC ON/OFF switchable (Eight AE spot is selectable)	
Signal-to-noise ratio:	More than 50 dB (Weight ON, AGC OFF)	
Video out:	BNC: 1.0 Vp-p, 75Ω, sync negative Y/C: Y: 1.0 Vp-p, 75Ω, sync negative C: 0.286 Vp-p, at burst level, 75Ω	
Operating temperature:	-10 to 50°C (14 to 122°F)	
Storage temperature:	-40 to 60°C (-40 to 140°F)	
Power requirements:	1) Multiplexing with YS-W150P/W250P 2) DC 12 V from DC 12 V power supply	AC 24 V, 50 Hz
Power consumption:	1) 5.5 W supplied from YS-W150P/W250P 2) 4.5 W at DC 12 V	6.0 W

Figura 3.9: Especificaciones de la cámara SSC-DC50P.

DXC-107A

Image device:	1/2-inch Hyper HAD Interline Transfer CCD (× 1)
Picture elements:	768(H) × 494(V)
Sensing area:	6.4mm × 4.8mm (1/2-inch)
Signal system:	EIA standard, NTSC color system
Scanning system:	525 lines, 2 : 1 interlace
Sync system:	Internal/external with VS
Horizontal resolution:	470 TV lines
Lens mount:	C mount
Sensitivity:	2000 lx with F5.6
Minimum illumination:	4.5 lx with F1.2
Gain control:	AGC (ON/OFF selectable)
Signal-to-noise ratio:	48dB
White balance:	AUTO: ATW or AWB PRESET: 3200K or 5600K
Electronic shutter:	OFF (1/60s), 1/100 (FL), 1/250, 1/1000, 1/2000, 1/4000, 1/10000 (s)
Phase control:	H phase control
CCD IRIS control:	BACKLIGHT/STANDARD/SPOTLIGHT selectable
Camera number ID:	Four-digit Characters: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, -, (SPACE)
Video out:	VBS: 1.0Vp-p, 75Ω, sync negative Y/C: Y: 1.0Vp-p, 75Ω, sync negative C: 0.286Vp-p, 75Ω, without sync
Operating temperature:	0°C to 40°C (32°F to 104°F)
Storage temperature:	-30°C to 60°C (-22°F to 140°F)
Power requirements:	DC 10.5 to 16V using CCMC-12P multi-core cable (supplied from CMA-D2/D7) DC 25 to 30V using co-axial cable (supplied from CMA-D7/YS-W230)
Power consumption:	3.3W using CCMC-12P multi-core cable 6.3W using co-axial cable
Maximum cable length:	300m using SC-2V co-axial cable and CMA-D7/YS-W230 100m using CCDC-100A cable and CMA-D2 25m using CCMC-12P25 multi-core cable and CMA-D2/D7

Figura 3.10: Especificaciones de la cámara DXC-107P.

Capítulo 4

Calibración

4.1. Introducción

El término *calibración de cámaras* refiere al proceso de caracterizar la óptica de una cámara, es decir, encontrar el conjunto de parámetros intrínsecos y extrínsecos, o simplemente de intrínsecos, que modelen completamente el proceso de formación de la imagen a través de la óptica en cuestión, dado un modelo matemático de la misma. En términos geométricos, este proceso consiste en asociar biunívocamente un rayo que pasa por el centro óptico de la cámara con un punto del plano de la imagen. Esto se hace adquiriendo imágenes de un *patrón de calibración* de estructura conocida y poniendo en correspondencia ciertos puntos de su estructura con su proyección en las imágenes.

Básicamente, la calibración de una cámara se lleva a cabo con uno de los siguientes objetivos, o con ambos:

- Conocer la proyección 2D de un objeto sobre una imagen a partir de su ubicación espacial.
- Extraer información espacial de un objeto a partir de las coordenadas de su proyección sobre una imagen.

En este capítulo se deducen en primer lugar las ecuaciones del modelo de cámara utilizado (*pinhole*) y luego se exponen los conceptos detrás de los diferentes métodos de calibración.

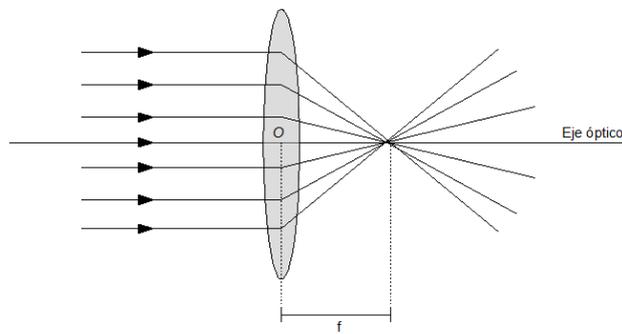
4.2. Modelo pinhole

En general, un sistema óptico, como una cámara, está formado por un conjunto de lentes cuya función es dirigir la luz, esto es, controlar la dirección en que ésta se propaga. El modelo más simple de lente es el de “lente delgada” [42], que consiste en un eje, llamado *eje óptico*, y un plano perpendicular al eje, llamado *plano focal*, con una apertura circular centrada en el *centro óptico*, es decir, en la intersección del eje

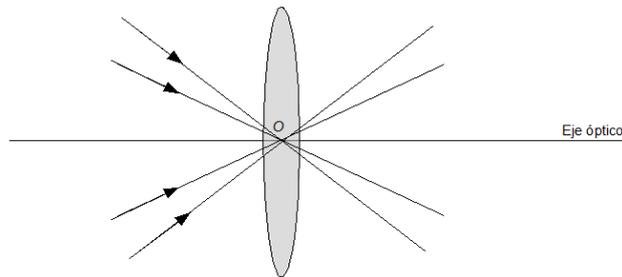
óptico con el plano focal. La lente delgada tiene dos parámetros: la distancia focal f y el radio r , y su función se puede describir mediante dos propiedades:

- Todos los rayos que inciden sobre la lente paralelos al eje óptico, convergen en el eje óptico a una distancia f del centro óptico. Este punto es el *foco* de la lente.
- Todos los rayos que pasan por el centro óptico se propagan a través de la lente sin cambiar su dirección.

La Figura 4.1 ilustra estas propiedades.



(a) Rayos paralelos al eje óptico.



(b) Rayos que pasan por el centro óptico.

Figura 4.1: Propiedades de las lentes delgadas.

Sea $p \in \mathbb{R}^3$ un punto en el espacio, a una distancia Z del centro óptico a lo largo del eje óptico, y sea x el punto en que se cortan dos rayos que pasan por p e inciden sobre la lente, uno paralelo al eje óptico y el otro pasando por el centro óptico. Descomponiendo los rayos que pasan por p en estas dos componentes, se puede demostrar que todos los rayos que pasan por p convergen en x al pasar por la lente.

Sea z la distancia de x al centro óptico en la dirección de eje óptico. Entonces los valores de Z y z están relacionados con f a través de la fórmula:

$$\frac{1}{Z} + \frac{1}{z} = \frac{1}{f}$$

que puede deducirse de la Figura 4.2 usando triángulos semejantes. El punto x es la imagen del punto p .

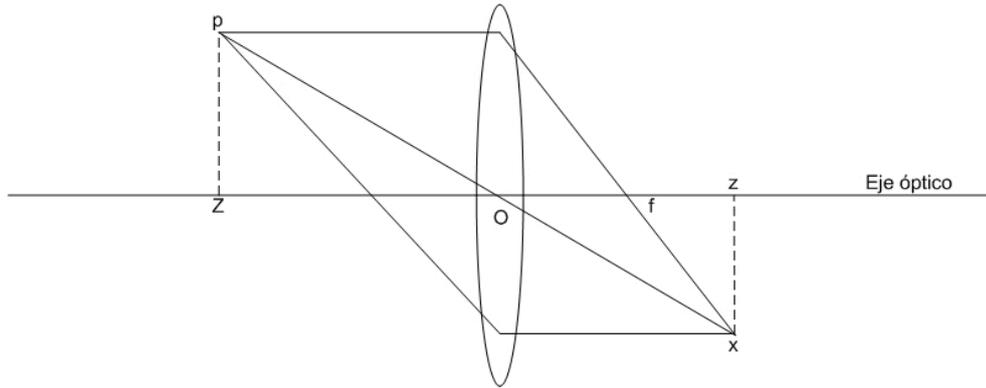


Figura 4.2: Formación de una imagen a través de una lente delgada.

Si ahora se hace tender a cero el radio de la lente delgada, todos los rayos pasarán por el centro óptico O , y por lo tanto seguirán en la misma dirección, sin desviarse. Por lo tanto, los únicos puntos que contribuyen a la imagen del punto p son los que están sobre la recta que pasa por el centro óptico O . Entonces, si se tiene el punto p de coordenadas $\mathbf{X} = [X, Y, Z]^T$ en el sistema de referencia con origen en el centro óptico y eje z igual al eje óptico, es inmediato que las coordenadas de p y su imagen x están relacionadas mediante

$$x = -f \frac{X}{Z}, \quad y = -f \frac{Y}{Z} \quad (4.1)$$

como puede deducirse de la Figura 4.3 utilizando triángulos semejantes. Este modelo de formación de la imagen se denomina *modelo pinhole ideal*.¹

Obsérvese que hay un signo de menos delante de las ecuaciones 4.1, lo cual hace que la imagen de un objeto se vea invertida en el plano de la imagen. Para eliminar este efecto, se puede suponer que el plano de la imagen está adelante del centro óptico, a una distancia $z = f$ en lugar de $z = -f$. De aquí en adelante se adoptará este modelo (*pinhole frontal*), como se muestra en la Figura 4.4. Las ecuaciones que relacionan las coordenadas del punto p y su imagen son ahora:

$$x = f \frac{X}{Z}, \quad y = f \frac{Y}{Z} \quad (4.2)$$

¹El modelo pinhole es una idealización del modelo de lente delgada, ya que al disminuir el radio de la apertura, la difracción pasa a ser dominante y por lo tanto el modelo de lente delgada puramente refractiva deja de ser válido.

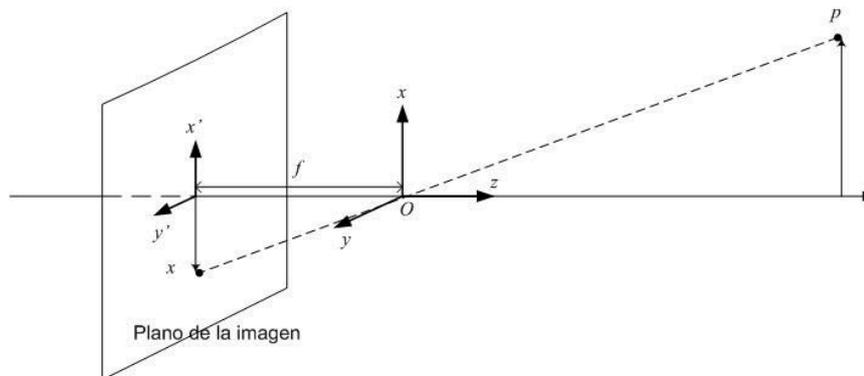


Figura 4.3: Modelo pinhole.

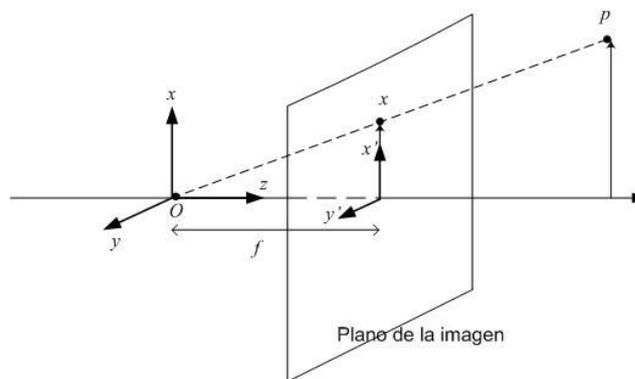


Figura 4.4: Modelo pinhole frontal.

A continuación se deducirá un modelo matemático que establece una correspondencia precisa entre puntos en el espacio (3D) y su proyección en el plano de la imagen (2D) [42, 29]. En forma general, el modelo debe considerar tres tipos de transformación:

1. transformaciones entre el sistema de coordenadas del mundo y el sistema de coordenadas de la cámara, dado que en general, un punto estará referido a un sistema de coordenadas que no necesariamente coincidirá con el de la cámara;
2. proyección de las coordenadas espaciales 3D a coordenadas en el plano de la imagen 2D
3. transformaciones entre los distintos sistemas posibles para expresar las coordenadas en el plano de la imagen

En primer lugar, sea $\mathbf{X}_0 = [X_0, Y_0, Z_0] \in \mathbb{R}^3$ un punto referido al sistema de coordenadas del mundo. Las coordenadas $\mathbf{X} = [X, Y, Z]$ del mismo punto en el

sistema de coordenadas de la cámara estarán relacionadas con \mathbf{X}_0 por un movimiento rígido

$$\mathbf{X} = R\mathbf{X}_0 + T \quad (4.3)$$

Escribiendo las ecuaciones 4.2 en forma matricial se tiene:

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X \\ Y \end{bmatrix}$$

y si además se escriben \mathbf{x} y \mathbf{X} en coordenadas homogéneas (ver Apéndice C):

$$Z \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (4.4)$$

Esta expresión asume que el origen de coordenadas de la imagen se encuentra en el *punto principal*, es decir, en la intersección del eje óptico con el plano de la imagen. Sin embargo, cuando se toman imágenes con cámaras CCD, es habitual expresar las coordenadas de un punto considerando el origen de coordenadas en el vértice superior izquierdo de la imagen. Además, es usual trabajar con coordenadas en *píxeles*, en lugar de expresarlas en unidades de longitud.

Para transformar las coordenadas a píxeles, es necesario multiplicar por un factor de escala que representa la cantidad de píxeles por unidad de longitud. En general, la dimensión de un pixel en la dirección x no es igual a la dimensión en la dirección y , por lo que se definen dos factores de escala s_x y s_y tales que, llamando x_s y y_s a las coordenadas en píxeles resulta:

$$\begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Una vez hecho esto, es necesario trasladar el origen de coordenadas al vértice superior izquierdo de la imagen; si (c_x, c_y) son las coordenadas del punto principal, entonces llamando x' y y' a estas nuevas coordenadas:

$$\begin{aligned} x' &= x_s + c_x \\ y' &= y_s + c_y \end{aligned}$$

y reescribiendo las relaciones anteriores en forma matricial:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & c_x \\ 0 & s_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Finalmente, para el caso en que los píxeles no son rectangulares, se introduce el factor de *skew*, s_θ que tiene en cuenta el ángulo θ entre los ejes x_s y y_s :

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & s_\theta & c_x \\ 0 & s_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4.5)$$

Combinando las ecuaciones 4.3, 4.4 y 4.5 se tiene el modelo completo que transforma las coordenadas homogéneas de un punto en el espacio en las coordenadas en píxeles de su proyección sobre el plano de la imagen:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} s_x & s_\theta & c_x \\ 0 & s_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix} \quad (4.6)$$

Habitualmente se combinan las dos primeras matrices, para dar lugar a la *matriz de parámetros intrínsecos*, A :

$$A = \begin{bmatrix} s_x & s_\theta & c_x \\ 0 & s_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_x \\ 0 & 0 & 1 \end{bmatrix}$$

donde $f_x = fs_x$ y $f_y = fs_y$ son la distancia focal expresadas en píxeles en la dirección x e y respectivamente. La tercera matriz se denomina *matriz de parámetros extrínsecos* y da la rotación y la traslación necesarias para llevar coordenadas en el sistema de referencia del mundo al sistema de referencia de la cámara. A las coordenadas expresadas en el sistema de la cámara se les denomina *coordenadas normalizadas*.

Modelado de la distorsión radial

Hasta ahora se asumió que la proyección a través de la cámara de los puntos espaciales sobre el plano de la imagen puede modelarse como una proyección perspectiva. Esto implica que dado cualquier punto del espacio, éste estará alineado con su imagen en el plano de la imagen y el centro de la cámara. Sin embargo, en las cámaras reales esta suposición puede no ser válida, ya que pueden presentar distorsión, fundamentalmente del tipo radial, en la que el punto real proyectado se relaciona con el ideal mediante un desplazamiento radial:

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = L(\tilde{r}, \tilde{x}, \tilde{y})$$

donde

- (\tilde{x}, \tilde{y}) es la posición ideal de los puntos proyectados (proyección perspectiva)
- (x_d, y_d) es la proyección real de los puntos afectados por la distorsión
- \tilde{r} es la distancia del punto (\tilde{x}, \tilde{y}) al punto principal
- $L(\tilde{r})$ es un factor de distorsión únicamente dependiente del radio \tilde{r}

4.3. Métodos de calibración de cámaras

Una vez obtenido el modelo de cámara, el proceso de calibración busca encontrar el conjunto de parámetros (f_x, f_y, s, c_x, c_y) , coeficientes de distorsión) que mejor describan el comportamiento de la cámara. En general, los algoritmos de calibración trabajan sobre imágenes de un objeto de estructura conocida, o un *patrón*, para poner en correspondencia un conjunto de puntos de referencia, cuya ubicación en el patrón es conocida con precisión, con su proyección sobre la imagen. A partir de esta información es posible extraer los valores de los parámetros, como se verá en las secciones siguientes. En la Figura 4.5 se muestran diferentes tipos de patrones: planos y tridimensionales, donde los puntos de referencia son las esquinas de un damero, o círculos, etc.

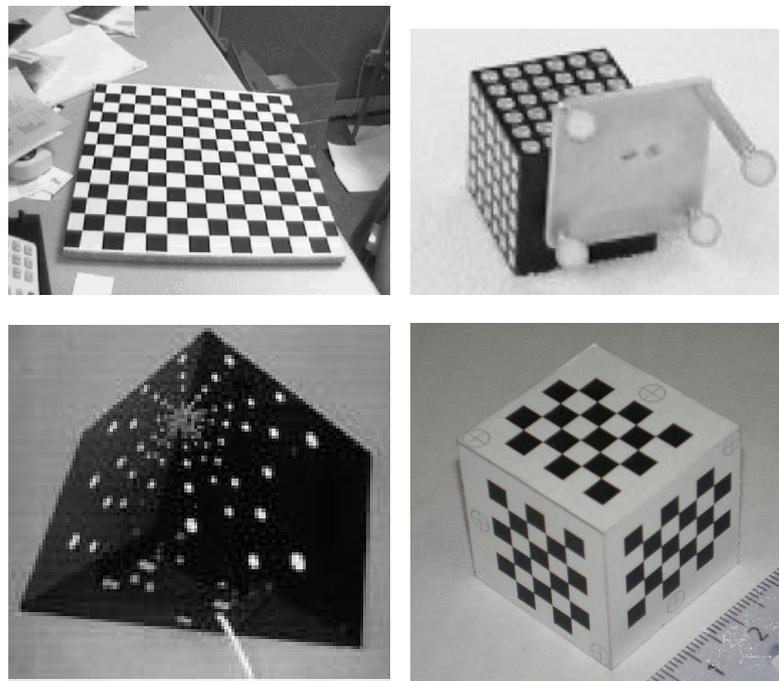


Figura 4.5: Patrones de calibración.

Como se explicó anteriormente, a través del proceso de calibración se obtiene la matriz de parámetros intrínsecos A , así como los parámetros extrínsecos, dados por una rotación R y una traslación t , que relacionan los sistemas de coordenadas del mundo y de la cámara. La Figura 4.6 muestra la relación entre los distintos sistemas de coordenadas y las matrices que resultan de la calibración.

En las siguientes subsecciones se describen los algoritmos utilizados. El método propuesto por Zhengyou Zhang [77] implementa una calibración en dos etapas a través de un patrón de estructura plana, considerando un modelo de cámara pinhole con dos coeficientes de distorsión radial. En la primera etapa se calcula una

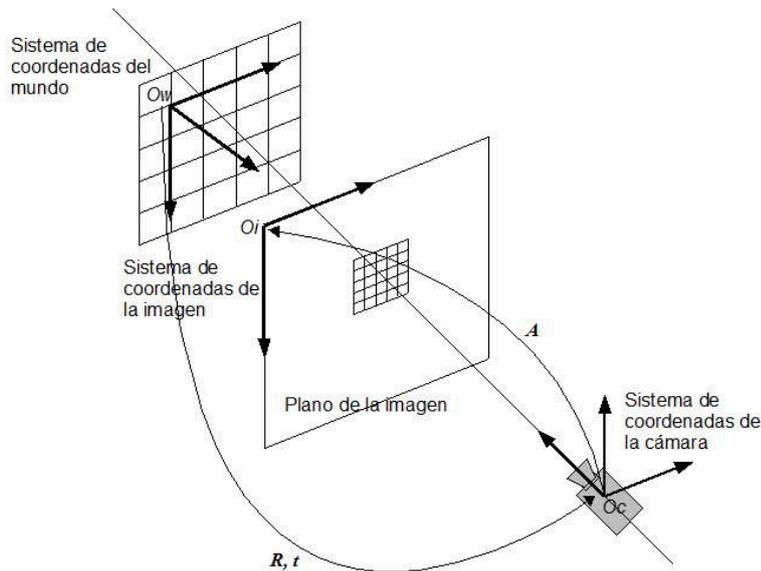


Figura 4.6: Relación entre los diferentes sistemas de coordenadas, parámetros intrínsecos y parámetros extrínsecos.

estimación inicial de los parámetros, la cual es tomada como semilla de un algoritmo iterativo que optimiza los parámetros para minimizar el error de reproyección de los puntos de referencia.

Por otro lado, el método de Transformación Lineal Directa o DLT (Direct Linear Transform) es un método lineal para encontrar una transformación que vincula conjuntos de puntos correspondientes, lo cual puede ser utilizado para encontrar la matriz de proyección de una cámara dada una cantidad suficiente de correspondencias entre puntos espaciales y su proyección en la imagen. Este método tiene la ventaja de ser general, por lo que puede aplicarse tanto para patrones planos como para patrones tridimensionales.

Finalmente, el método desarrollado por Jean Yves Bouguet [7] combina ambos métodos en un algoritmo también en dos etapas, donde la estimación inicial de los parámetros se hace mediante el método de Zhang en el caso de estar trabajando con un patrón plano, o mediante DLT en caso contrario. Bouguet introduce además los conceptos de puntos y líneas de fuga en el cálculo de la estimación inicial de los parámetros intrínsecos. Considera el modelo de cámara propuesto por Janne Heikkilä y Olli Silvén [30], el cual incorpora dos coeficientes de distorsión adicionales para modelar la distorsión tangencial.

4.3.1. Método de calibración de Zhang

El método de calibración propuesto por Zengyou Zhang [77], se caracteriza por su flexibilidad, dado que sólo requiere que la cámara observe un patrón plano en diferentes posiciones espaciales. Esto puede hacerse moviendo el patrón o la cámara

libremente, y no es necesario conocer el movimiento. Debido a esto, y dado que el método contempla la distorsión provocada por el sistema óptico de la cámara, resulta ser un algoritmo adecuado para el uso con cámaras estándar. Por otra parte, el hecho de utilizar un patrón plano reduce los costos, al no tener que contar con patrones tridimensionales extremadamente precisos.

La técnica consta de dos etapas diferentes: en la primera, se calcula una aproximación inicial de todos los parámetros intrínsecos de la cámara; en la segunda, estos valores se refinan minimizando la distancia entre los puntos conocidos sobre la imagen del patrón y los que se calculan a partir de los parámetros hallados.

A continuación se plantean las ecuaciones básicas que describen el problema, así como la solución propuesta por este método.

Ecuaciones básicas

Sea $m = [u \ v]^T$ un punto 2D, $M = [X \ Y \ Z]^T$ un punto 3D, y $\tilde{m} = [u \ v \ 1]^T$ y $\tilde{M} = [X \ Y \ Z \ 1]^T$ los mismos puntos expresados en coordenadas homogéneas. Si se utiliza un modelo pinhole para la cámara, entonces un punto en 3D, M , estará relacionado con su proyección en la imagen, m , mediante la ecuación:

$$s\tilde{m} = A [R \ t] \tilde{M}$$

donde

$$A = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

es la matriz de parámetros intrínsecos de la cámara, s es un factor de escala arbitrario y R y t los parámetros extrínsecos.

Se puede asumir, sin perder generalidad, que en cada orientación del patrón plano que vaya a utilizarse para calibrar, este se encuentra contenido en el plano $Z = 0$ del sistema de coordenadas del mundo. En estas condiciones, llamando r_i a las columnas de R , se tiene que

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A [r_1 \ r_2 \ r_3 \ t] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = A [r_1 \ r_2 \ t] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (4.7)$$

de donde se deduce que existe una homografía, H , que relaciona un punto en el espacio, M , con su correspondiente en la imagen, m :

$$s\tilde{m} = H\tilde{M} \quad (4.8)$$

con

$$H = A [r_1 \ r_2 \ t]$$

Esta matriz H es 3×3 y está definida a menos de un factor de escala.

Cálculo de la homografía y restricciones sobre los parámetros

Dada una imagen del patrón plano, puede calcularse la homografía H de la manera que se describe a continuación. Sean M_i y m_i los puntos del patrón y los puntos de la imagen respectivamente; idealmente estos puntos deberían satisfacer la ecuación $s\tilde{m}_i = HM_i$, pero en la práctica esto no se cumple dado que los puntos extraídos de la imagen se ven afectados por ruido. Lo que se busca entonces es encontrar una matriz H que sea óptima, es decir, que minimice la distancia $\sum_i \|m_i - \hat{m}_i\|^2$ entre los puntos conocidos sobre la imagen del patrón, m_i , y los calculados mediante la Ecuación 4.8 a partir de la homografía que se busca, \hat{m}_i . Este es un problema de mínimos cuadrados no lineal; para resolverlo pueden aplicarse métodos iterativos como el algoritmo de Levenberg - Marquardt, que es el enfoque propuesto por Zhang. Una explicación detallada de este algoritmo puede encontrarse en [29]. Para dar una idea, este es un algoritmo iterativo de estimación de parámetros similar al método de Newton, con algunas modificaciones que le permiten una convergencia más rápida, especialmente en el caso de tener que estimar un número elevado de parámetros.

Cálculo de la estimación inicial

Una vez calculada H a partir de los pares de puntos correspondientes, llamando h_i a sus columnas se tiene que

$$\begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = \lambda A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$

donde λ es una constante escalar arbitraria. Dado que R es una matriz de rotación, sus columnas son vectores ortonormales, por lo que pueden plantearse las siguientes ecuaciones relativas a su norma y al producto interno entre ellas:

$$r_1^T r_2 = 0 \Rightarrow (A^{-1}h_1)^T (A^{-1}h_2) = 0 \Rightarrow h_1^T A^{-T} A^{-1} h_2 = 0 \quad (4.9)$$

$$\|r_1\|^2 = \|r_2\|^2 = 1 \Rightarrow r_1^T r_1 = r_2^T r_2 \Rightarrow h_1^T A^{-T} A^{-1} h_1 = h_2^T A^{-T} A^{-1} h_2 \quad (4.10)$$

Éstas son las restricciones básicas sobre los parámetros intrínsecos que se obtienen dada una homografía. Dado que una homografía tiene 8 grados de libertad y que 6 de los parámetros son extrínsecos (3 de rotación y 3 de traslación) sólo se obtienen dos ecuaciones sobre los parámetros intrínsecos.

Cálculo de los parámetros

Como se mencionó, el método consiste en una primera etapa en la que se encuentra una aproximación inicial de los parámetros de la cámara, seguida de una etapa de optimización de la solución.

Sea $B = A^{-T}A^{-1} \equiv \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix}$. Entonces las ecuaciones anteriores quedan:

$$h_1^T B h_2 = 0$$

$$h_1^T B h_1 = h_2^T B h_2$$

Calculando B en función de los parámetros, se llega a la siguiente expresión para la matriz:

$$B = \begin{bmatrix} \frac{1}{f_x^2} & -\frac{s}{f_x^2 f_y} & \frac{sc_y - c_x f_y}{f_x^2 f_y} \\ \frac{-s}{f_x^2 f_y} & \frac{s^2}{f_x^2 f_y^2} + \frac{1}{f_y^2} & -\frac{s(sc_y - c_x f_y)}{f_x^2 f_y^2} - \frac{c_y}{f_y^2} \\ \frac{sc_y - c_x f_y}{f_x^2 f_y} & -\frac{s(sc_y - c_x f_y)}{f_x^2 f_y^2} - \frac{c_y}{f_y^2} & \frac{(sc_y - c_x f_y)^2}{f_x^2 f_y^2} + \frac{c_y^2}{f_y^2} + 1 \end{bmatrix} \quad (4.11)$$

Se observa que B es una matriz simétrica, por lo que puede representarse por un vector de 6 elementos $b = [B_{11} \ B_{12} \ B_{13} \ B_{21} \ B_{22} \ B_{33}]^T$. Habiendo definido este vector, y operando con la matriz B , los términos de la forma $h_i^T B h_j$ pueden escribirse:

$$h_i^T B h_j = v_{ij}^T$$

donde

$$v_{ij} = [h_{i1}h_{j1} \quad h_{i1}h_{j2} + h_{i2}h_{j1} \quad h_{i2}h_{j2} \quad h_{i3}h_{j1} + h_{i1}h_{j3} \quad h_{i3}h_{j2} + h_{i2}h_{j3} \quad h_{i3}h_{j3}]$$

Por lo tanto, las ecuaciones 4.9 y 4.10 obtenidas para los parámetros intrínsecos pueden reescribirse como ecuaciones homogéneas en b :

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0 \quad (4.12)$$

Si se toman n imágenes del patrón plano y se escriben estas ecuaciones en forma matricial se llega a una ecuación del tipo $Vb = 0$, donde V es una matriz de $2n \times 6$. Una vez estimado el vector b que cumple con esta relación, se pueden calcular los parámetros intrínsecos como sigue:

$$\begin{aligned} c_y &= (B_{12}B_{13} - B_{11}B_{23}) / (B_{11}B_{22} - B_{12}^2) \\ \lambda &= B_{33} - [B_{13}^2 + c_y(B_{12}B_{13} - B_{11}B_{23})] / B_{11} \\ f_x &= \sqrt{\lambda / B_{11}} \\ f_y &= \sqrt{\lambda B_{11} / (B_{11}B_{22} - B_{12}^2)} \\ s &= -B_{12} f_x^2 f_y / \lambda \\ c_x &= sc_y / f_x - B_{13} f_x^2 / \lambda \end{aligned}$$

Estimación de máxima similitud

La solución anterior para los parámetros intrínsecos de la cámara se obtuvo minimizando una distancia algebraica que no tiene ningún significado físico. El siguiente paso del método consiste en refinar la solución minimizando una nueva distancia, como se describe a continuación.

Suponiendo que se toman n imágenes del patrón y en cada una de ellas se conocen m puntos del modelo, entonces se busca minimizar

$$\sum_{i=1}^n \sum_{j=1}^m \|m_{ij} - \hat{m}(A, R_i, t_i, M_j)\|^2$$

donde $\hat{m}(P, R_i, t_i, M_j)$ es la proyección del punto M_j en la imagen i de acuerdo con la Ecuación 4.8. Nuevamente, este es un problema de minimización no lineal que puede resolverse con métodos como el de Levenberg - Marquardt y utilizando como estimación inicial la calculada anteriormente.

Estimación de la distorsión radial

En los desarrollos anteriores no se ha considerado la distorsión introducida por los lentes de la cámara. Este método de calibración contempla este efecto, pero considera sólo los dos primeros términos de la distorsión radial, con el argumento de que de acuerdo con la literatura existente sobre el tema, la función de distorsión está completamente dominada por estos coeficientes, en especial por el primero; más aún, el hecho de considerar coeficientes de orden superior podría causar inestabilidad numérica.

Sean (u, v) los puntos ideales (sin distorsión) en coordenadas de la imagen y (\tilde{u}, \tilde{v}) los puntos correspondientes observados en la imagen. De igual manera, sean (x, y) y (\tilde{x}, \tilde{y}) los puntos ideales y con distorsión en coordenadas normalizadas. Con esta notación, se tiene que:

$$\begin{aligned}\tilde{x} &= x + x \left[k_1 (x^2 + y^2) + k_2 (x^2 + y^2)^2 \right] \\ \tilde{y} &= y + y \left[k_1 (x^2 + y^2) + k_2 (x^2 + y^2)^2 \right]\end{aligned}$$

donde k_1 y k_2 son los coeficientes de distorsión. Dado que $u = c_x + x$, $v = c_y + y$, $\tilde{u} = c_x + \tilde{x}$ y $\tilde{v} = c_y + \tilde{y}$, las ecuaciones anteriores se pueden escribir como:

$$\tilde{u} = u + (u - c_x) \left\{ k_1 [(u - c_x)^2 + (v - c_y)^2] + k_2 [(u - c_x)^2 + (v - c_y)^2]^2 \right\} \quad (4.13)$$

$$\tilde{v} = v + (v - c_y) \left\{ k_1 [(u - c_x)^2 + (v - c_y)^2] + k_2 [(u - c_x)^2 + (v - c_y)^2]^2 \right\} \quad (4.14)$$

Una vez más, para resolver los parámetros incluyendo los coeficientes de distorsión, se propone minimizar la función:

$$\sum_{i=1}^n \sum_{j=1}^m \|m_{ij} - \hat{m}(P, k_1, k_2, R_i, t_i, M_j)\|^2$$

donde $\hat{m}(P, k_1, k_2, R_i, t_i, M_j)$ es la proyección del punto M_j en la imagen i según la Ecuación 4.8, seguida de la corrección por la distorsión dada por las Ecuaciones 4.13 y 4.14.

4.3.2. Transformación Lineal Directa - Algoritmo DLT

El método DLT (Direct Linear Transform) fue propuesto por Abdel-Aziz y Karara y es uno de los métodos más usados para calibración de cámaras, aunque es un algoritmo general que permite encontrar mapeos lineales entre dos conjuntos de datos cualesquiera, dada una cierta cantidad de correspondencias entre ambos [29, 6]. En esta sección se describen los conceptos básicos, aplicados al problema de encontrar una homografía que transforma un conjunto de puntos \mathbf{x}_i del espacio proyectivo \mathbb{P}^2 (ver Apéndice C) en el conjunto de puntos $\mathbf{x}'_i \in \mathbb{P}^2$. En una situación práctica, \mathbf{x}_i y \mathbf{x}'_i son puntos en dos imágenes.

Dado un conjunto de correspondencias $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ se quiere encontrar una matriz H que haga la transformación $\mathbf{x}'_i = H\mathbf{x}_i$, pero ¿cuál es la cantidad mínima de correspondencias necesaria? Por un lado, la matriz H debe ser una matriz 3×3 , de nueve componentes, pero está definida a menos de un factor de escala, con lo cual el número de grados de libertad de H es ocho. Por otro lado, cada una de las correspondencias impone dos condiciones, una por cada coordenada. En consecuencia, es necesario contar con un mínimo de cuatro correspondencias para determinar H .

El primer paso consiste en escribir la ecuación $\mathbf{x}'_i = H\mathbf{x}_i$ en términos del producto vectorial $\mathbf{x}'_i \times H\mathbf{x}_i = 0$, lo cual dará lugar a una solución más sencilla para H . Llamando h^{jT} a la fila j -ésima de H , se puede ver que

$$H\mathbf{x}_i = \begin{pmatrix} h^{1T}\mathbf{x}_i \\ h^{2T}\mathbf{x}_i \\ h^{3T}\mathbf{x}_i \end{pmatrix}$$

Escribiendo explícitamente el producto vectorial en función de las coordenadas de $\mathbf{x}'_i = (x'_i, y'_i, w'_i)^T$:

$$\mathbf{x}'_i \times H\mathbf{x}_i = \begin{pmatrix} y'_i h^{3T}\mathbf{x}_i - w'_i h^{2T}\mathbf{x}_i \\ w'_i h^{1T}\mathbf{x}_i - x'_i h^{3T}\mathbf{x}_i \\ x'_i h^{2T}\mathbf{x}_i - y'_i h^{1T}\mathbf{x}_i \end{pmatrix}$$

y dado que $h^{jT}\mathbf{x}_i = \mathbf{x}_i h^{jT}$, la ecuación anterior se puede reescribir en la forma $A_i h = 0$, donde A_i es una matriz 3×9 y h un vector de 9 componentes formado por las entradas de H :

$$A_i h = \begin{bmatrix} \mathbf{0}^T & -w'_i x_i^T & y'_i x_i^T \\ w'_i x_i^T & \mathbf{0}^T & -x'_i x_i^T \\ -y'_i x_i^T & x'_i x_i^T & \mathbf{0}^T \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0 \quad (4.15)$$

Si se observa la ecuación 4.15 se puede ver que la tercera fila es una combinación lineal de las dos primeras, con lo cual se tienen sólo dos ecuaciones linealmente independientes por cada correspondencia y la tercera puede eliminarse:

$$A_i h = \begin{bmatrix} \mathbf{0}^T & -w'_i x_i^T & y'_i x_i^T \\ w'_i x_i^T & \mathbf{0}^T & -x'_i x_i^T \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0 \quad (4.16)$$

donde ahora A_i es una matriz de 2×9 .

Para cada correspondencia se tienen una ecuación del tipo de la Ecuación 4.16, y agrupando todas las ecuaciones se llega a una ecuación lineal en h de la forma $Ah = 0$.

Hasta ahora las ecuaciones fueron desarrolladas para encontrar la homografía que transforma un conjunto de puntos 2D en otro conjunto de puntos 2D. Pero el método puede extenderse y utilizarse también para calcular la matriz de proyección de una cámara, es decir, una matriz P de 3×4 que transforme un conjunto X_i de puntos 3D en el espacio en un conjunto x_i puntos 2D en el plano de la imagen. En este caso, planteando los mismos pasos que en el caso anterior se llega a la ecuación:

$$A_i p = \begin{bmatrix} \mathbf{0}^T & -w_i X_i^T & y_i X_i^T \\ w_i X_i^T & \mathbf{0}^T & -x_i X_i^T \\ -y_i X_i^T & x_i X_i^T & \mathbf{0}^T \end{bmatrix} \begin{pmatrix} p^1 \\ p^2 \\ p^3 \end{pmatrix} = 0 \quad (4.17)$$

donde p es un vector de doce elementos formado por las entradas de P , p^j es la fila j -ésima de la matriz P y A_i es una matriz de 3×12 . Al igual que en el caso anterior, la tercera ecuación es una combinación lineal de las dos primeras y por lo tanto puede eliminarse, resultando para cada correspondencia una ecuación del tipo:

$$\begin{bmatrix} \mathbf{0}^T & -w_i X_i^T & y_i X_i^T \\ w_i X_i^T & \mathbf{0}^T & -x_i X_i^T \end{bmatrix} \begin{pmatrix} p^1 \\ p^2 \\ p^3 \end{pmatrix} = 0 \quad (4.18)$$

Dado que la matriz P tiene 11 grados de libertad (sin tomar en cuenta la escala), es posible encontrar una solución para p contando con un mínimo de 6 correspondencias.¹ Sin embargo, si se dan $n \geq 6$ correspondencias, y dado que en la práctica las coordenadas no son ideales sino que están afectadas por ruido, el sistema $Ap = 0$ es sobredeterminado y no existe una solución exacta por lo que deberá buscarse una solución estimada.² Un criterio para esto es minimizar el

¹Estrictamente, son necesarias once ecuaciones, con lo cual alcanza con utilizar cinco correspondencias y sólo una de las ecuaciones de la sexta.

²Si las posiciones de los puntos fueran ideales, entonces la matriz A aún tendría rango 11 y el sistema no sería sobredeterminado.

error algebraico $\|Ap\|$; además, para evitar la solución nula es necesario imponer una condición adicional, por ejemplo $\|p\| = 1$.

Con estas restricciones, el problema puede resolverse como sigue: sea $A = UDV^T$ la descomposición en valores singulares de A , y sea $y = V^T p$. Entonces el problema consiste en minimizar $\|UDy\|$. Pero dado que $\|Ux\| = \|x\|$ para cualquier vector x , se tiene que $\|UDy\| = \|Dy\|$. Asimismo, $\|V^T p\| = \|p\|$, con lo cual el problema es equivalente a minimizar $\|Dy\|$ sujeto a $\|y\| = 1$. Finalmente, como D es una matriz diagonal con sus elementos en orden descendente, se concluye que la solución es $y = (0 \ 0 \ \dots \ 0 \ 1)$ y por lo tanto $p = Vy$ es la última columna de V .

4.3.3. Método de Bouguet

El método de calibración de cámaras desarrollado por Jean Yves Bouguet [7, 8] combina los métodos de Zhang (pensado únicamente para trabajar con patrones planos) y DLT (de uso más general) para contemplar tanto el uso de patrones planos como de patrones tridimensionales. Al igual que el método de Zhang, Bouguet propone la resolución del problema en dos etapas: una primera etapa de inicialización de los parámetros intrínsecos y extrínsecos, seguida de la optimización de todos los parámetros para minimizar el error de reproyección de los puntos de referencia sobre el plano de la imagen.

Dadas las correspondencias de puntos en el espacio y su proyección en la imagen, el primer paso consiste en determinar si el patrón utilizado tiene estructura tridimensional o si se trata de un patrón plano, ya que la forma de calcular los parámetros es diferente en ambos casos.

Patrón plano. En caso de estar trabajando con un patrón plano, la inicialización de los parámetros extrínsecos (homografía H) es idéntica a la propuesta por Zhang, es decir, se minimiza la distancia entre las coordenadas 2D de la proyección de los puntos de referencia sobre la imagen, m_i , y la proyección \hat{m}_i obtenida a partir de las coordenadas 3D de los mismos puntos utilizando la homografía H (Ecuación 4.8). En concreto, se resuelve

$$\min_H \sum_i \|m_i - \hat{m}_i\|^2$$

a través del algoritmo de Levenberg-Marquardt.

Sin embargo, la estimación de los parámetros intrínsecos se obtiene de una forma completamente distinta respecto al algoritmo de Zhang, ya que se hace uso de la geometría del espacio B-dual, en particular de los conceptos de puntos y líneas de fuga. Estos conceptos permiten derivar expresiones cerradas para los parámetros, utilizando herramientas matemáticas sencillas y compactas, a la vez que se aprovecha toda la información geométrica presente en la escena a través del patrón de calibración [8]. En el Apéndice C se muestra cómo se pueden calcular los parámetros intrínsecos de esta manera; la deducción detallada de las expresiones puede consultarse en [8]. En esta primera etapa, el punto principal se fija en el centro

de la imagen, y se consideran nulos todos los coeficientes de distorsión y el factor de *skew*, por lo tanto los únicos parámetros que se estiman son f_x y f_y .

Patrón tridimensional. Si se detecta que el patrón tiene estructura tridimensional, la estimación inicial de los parámetros extrínsecos se hace utilizando directamente el método DLT que se introdujo en la Subsección 4.3.2.

En cuanto a los parámetros intrínsecos, la inicialización en este caso es “manual”, fijándose

$$f_x = f_y = \frac{n_x/2}{\tan \frac{\pi \cdot 35}{360}}$$

donde n_x es el tamaño de la imagen en la dirección horizontal. Esta expresión asume que el ángulo de visión de la cámara es de 35 grados para dar una aproximación de la distancia focal. La relación entre el campo de visión de la cámara y sus parámetros intrínsecos puede consultarse en el Capítulo 6, Sección 6.3.2. Al igual que en el caso de patrón plano, el punto principal, los coeficientes de distorsión y el factor de *skew* inicialmente no se estiman y se fijan a cero.

En la etapa de optimización, se busca refinar el valor de todos los parámetros, incluyendo el punto principal, coeficientes de distorsión y factor de *skew*, con el fin de minimizar el error de reproyección de los puntos. Para la minimización, el método utiliza el algoritmo de Levenberg-Marquardt.

Modelo de distorsión

El método de calibración de Bouguet también contempla el efecto de la distorsión en las cámaras. Además de considerar los dos primeros términos de la distorsión radial como lo hace Zhang, considera coeficientes que modelan una posible distorsión tangencial.

Sean (u, v) los puntos ideales (sin distorsión) en coordenadas de la imagen y (\tilde{u}, \tilde{v}) los puntos correspondientes observados en la imagen. De igual manera, sean (x, y) y (\tilde{x}, \tilde{y}) los puntos ideales y con distorsión en coordenadas normalizadas. Con esta notación, se tiene que:

$$\begin{aligned}\tilde{x} &= x + x \left[k_1 (x^2 + y^2) + k_2 (x^2 + y^2)^2 \right] + dx \\ \tilde{y} &= y + y \left[k_1 (x^2 + y^2) + k_2 (x^2 + y^2)^2 \right] + dy\end{aligned}$$

donde k_1 y k_2 son los coeficientes de distorsión del modelo de distorsión radial descrito en la Sección 4.3.1 y $d = \begin{pmatrix} dx & dy \end{pmatrix}^T$ es un vector que modela la distorsión tangencial:

$$d = \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} 2 \cdot k_3 \cdot x \cdot y + k_4 \left((x^2 + y^2)^2 + 2x^2 \right) \\ k_3 \left((x^2 + y^2)^2 + 2y^2 \right) + 2 \cdot k_4 \cdot x \cdot y \end{pmatrix} \quad (4.19)$$

Una vez más, para resolver los parámetros se propone minimizar la función:

$$\sum_{i=1}^n \sum_{j=1}^m \|m_{ij} - \hat{m}(P, k_1, k_2, k_3, k_4, R_i, t_i, M_j)\|^2$$

pero ahora se agregan como parámetros los coeficientes que modelan la distorsión tangencial.

Capítulo 5

Implementación de la Calibración

5.1. Introducción

La calibración del microscopio es un paso indispensable y fundamental en el proceso de realidad aumentada. Se entiende por calibrar al microscopio, encontrar los parámetros intrínsecos que modelan la óptica del mismo y los parámetros extrínsecos que lo ubican en el espacio respecto a un sistema de referencia conocido. En la práctica ambos procesos se desarrollan simultáneamente a partir de la asociación, a través de un cierto modelo, de puntos espaciales con puntos en la imagen.

La técnica de calibración consistió en realizar varias observaciones con el microscopio de un patrón de geometría conocida, variando la posición relativa entre el microscopio y el patrón entre observaciones. Para cada observación se obtuvieron las posiciones espaciales y en la imagen de algunos puntos característicos y se determinaron los parámetros del modelo a partir de los algoritmos vistos en el Capítulo 4.

El presente capítulo describe cómo se fueron modificando el método de calibración y los parámetros del modelo de la cámara durante este proyecto, hasta llegar al procedimiento de calibración finalmente implementado.

5.1.1. Elección del modelo pinhole

La elección del modelo de cámara pinhole para describir el comportamiento de la óptica del microscopio y de las cámaras de laboratorio se debió fundamentalmente a dos razones:

- Es un modelo muy simple y existen diversos algoritmos de calibración que se basan en dicho modelo, en particular los que presentan en el Capítulo 4
- Existen sistemas de realidad aumentada para microscopios quirúrgicos actualmente en uso que consideran el modelo pinhole para todo el rango de aumento, con buenos resultados [21, 15].

5.2. Particularidades de la calibración del microscopio

En el Capítulo 4 se vio que los algoritmos de calibración cuentan con dos etapas. En la primera se determinan los parámetros de calibración resolviendo un sistema lineal y luego se realiza un refinamiento de los mismos mediante un algoritmo iterativo no lineal que toma como semilla la solución inicial y que, a partir de ella, recalcula los parámetros en pos de minimizar una función de costo.

La obtención de una buena estimación inicial es muy importante ya que la solución de la minimización no lineal depende fuertemente de la semilla inicial. Si se parte de parámetros que poco tienen que ver con la óptica de la cámara seguramente se obtengan valores que minimicen “matemáticamente” el error en un cierto entorno de la función de costo pero que carezcan de sentido físico.

En comparación con la calibración de cámaras convencionales la calibración del microscopio presenta una característica que hace que el procedimiento resulte más difícil: el reducido campo de visión en la dirección del eje óptico.

Si se calibra con un patrón plano, por ejemplo, resulta difícil encontrar posiciones diferentes del patrón sin que éste se vaya de foco. El hecho de que las posiciones sean parecidas hace que se brinde al algoritmo muy poca información espacial: es casi como darle la misma información repetida muchas veces. En general, una buena práctica de calibración con un patrón plano debe cumplir con las siguientes características:

- El patrón debe ocupar la mayor región posible de la imagen. Notar que debido al reducido campo de visión del microscopio un patrón que ocupa mucho espacio en la imagen, es difícil de girar sin que se vaya de foco.
- Se deben tomar imágenes del patrón con diversos ángulos. Notar que si se coloca al patrón en una posición paralela al plano de la imagen es imposible distinguir una variación en la posición de la cámara en la dirección del eje óptico (T_z) de una variación en el aumento del lente (f).
- Se deben tomar imágenes del patrón a distintas profundidades. Esto permite apreciar los efectos de la proyección en perspectiva. Se recomienda que la distancia entre los puntos utilizados para calibrar más cercanos y lejanos sea comparable con la distancia entre la cámara y los puntos de calibración. Evidentemente esto es imposible de lograr con el microscopio.

Vale la pena mencionar que a medida que se incrementa el aumento del microscopio, las condiciones de trabajo hacen que el problema se vuelva peor condicionado.

5.3. Métodos de calibración

En la Figura 5.1 se muestran las transformaciones involucradas en el proceso de calibración:

- **A**: Es la matriz de parámetros intrínsecos. Se supone que esta transformación no varía con el tiempo si no se varían los valores de aumento y distancia de trabajo del microscopio.
- **H**: Es la ubicación de la cámara respecto a los marcadores. Se supone que esta transformación no varía con el tiempo si no se varían los valores de aumento y distancia de trabajo del microscopio.
- **M**: Es la ubicación de los marcadores adheridos al microscopio respecto al localizador.
- **P**: Es la posición de un punto en el espacio respecto al localizador.

Si se llama p a la proyección en la imagen del punto P , entonces se deduce a partir del diagrama la relación entre estos:

$$p = AH^{-1}M^{-1}P \quad (5.1)$$

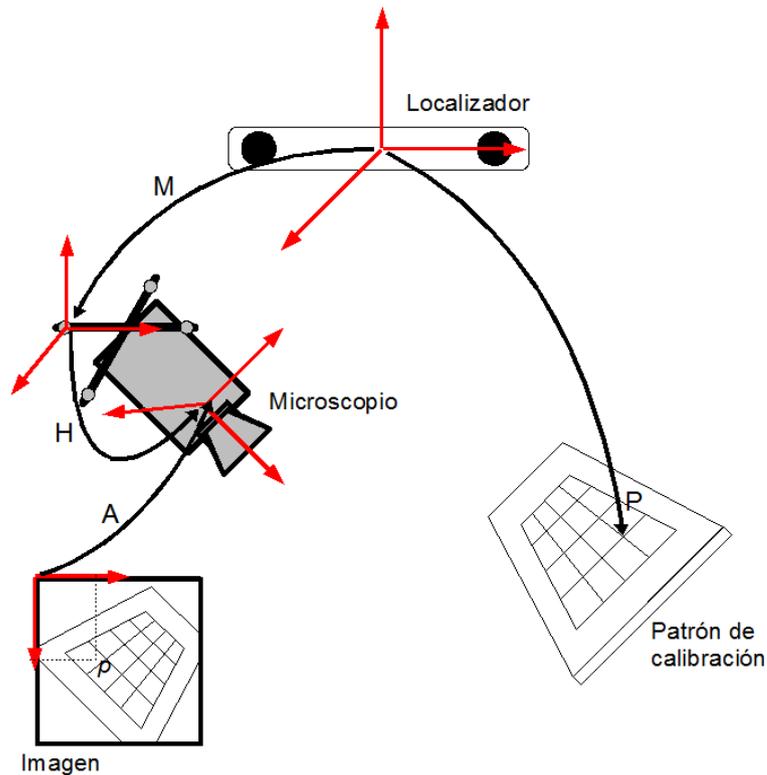


Figura 5.1: Esquema de calibración.

El objetivo del proceso de calibración es encontrar las matrices A y H . A la acción conjunta de las matrices A y H se le denomina matriz de proyección. Se implementaron tres variantes; las mismas se explican a continuación:

5.3.1. Primera variante

Se calculan los parámetros intrínsecos utilizando un damero. Las coordenadas espaciales del damero están referidas a un sistema de referencia solidario al mismo (ver Figura 5.2). Para realizar el cálculo de los parámetros intrínsecos se debe estimar la posición del damero respecto a la cámara para cada imagen. El refinamiento de la solución inicial involucra una minimización no lineal que cuenta con una cantidad de parámetros N , con

$$N = 6 \times \text{número de imágenes} + \text{número de parámetros intrínsecos}$$

El factor multiplicativo 6 representa los seis grados de libertad que se tienen para ubicar el patrón en el espacio (3 ángulos de la rotación R más las 3 componentes de la traslación t).

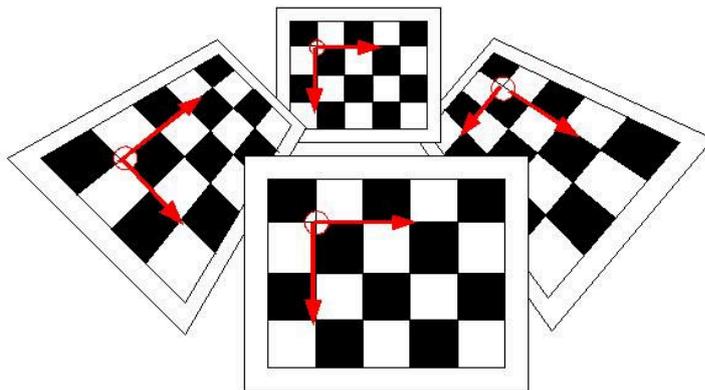


Figura 5.2: Cálculo de parámetros intrínsecos utilizando sólo un damero

Una vez determinados los parámetros intrínsecos se halla la posición de la cámara respecto a los marcadores adheridos al microscopio. Para ello se realiza el siguiente procedimiento:

- Se coloca el damero en una posición donde se puedan detectar las esquinas en la imagen y se puedan determinar sus coordenadas espaciales utilizando el localizador.
- A partir de la posición de las esquinas del damero en el espacio y en la imagen se calcula la posición de la cámara respecto al localizador.
- Sin mover el microscopio, se adquiere la transformación M que representa la posición de los marcadores solidarios al microscopio respecto al localizador.
- Con la posición de los marcadores y de la cámara del microscopio se halla H .

5.3.2. Segunda variante

En esta variante también se utilizan diversas posiciones de un damero, pero las coordenadas espaciales de las esquinas están dadas en el sistema de coordenadas del localizador, como se esquematiza en la Figura 5.3. De esta forma se está utilizando el localizador para “simular” un patrón tridimensional. Para determinar la posición espacial de las esquinas respecto al localizador es necesario previamente calibrar el patrón; los detalles de este procedimiento se explican en el Apéndice B.

La principal ventaja que presenta esta variante respecto a la anterior es que ahora todos los puntos están referidos a un mismo sistema de coordenadas, esto provoca que la minimización no lineal se haga en función de una cantidad de parámetros $N = 6 + \text{número de parámetros intrínsecos}$, reduciendo así considerablemente el número de parámetros intervinientes en la minimización. En este caso, se deben determinar sólo 6 parámetros extrínsecos que dan la ubicación del sistema de coordenadas del mundo (localizador) respecto a la cámara.

Una ventaja adicional es que para estimar H se utilizan n veces más puntos espaciales que en la primera variante, con n igual a la cantidad de imágenes, por lo que se es más robusto al ruido introducido por el sensor Polaris.

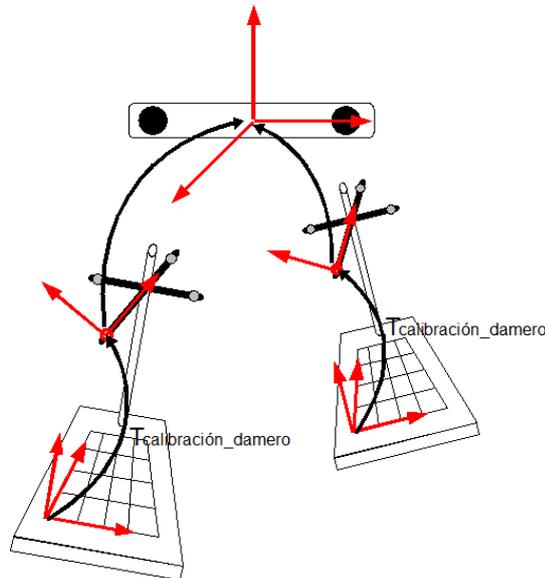


Figura 5.3: Cómputo de parámetros intrínsecos con un damero y el localizador Polaris. Las coordenadas de las esquinas están dadas en el sistema de referencia del localizador.

Como ahora se está calibrando con un patrón “tridimensional”, en vez de uno plano, se utiliza el algoritmo DLT en vez del cálculo de homografías para determinar los parámetros extrínsecos. Los intrínsecos por su parte no pueden ser calculados a partir de los puntos de fuga por lo que se debe indicar una matriz de parámetros inicial. Dicha matriz son los parámetros intrínsecos obtenidos en la primera variante.

5.3.3. Tercera variante

Un inconveniente que se presenta al simular la estructura 3D con el localizador es que el tiempo que transcurre entre el momento de adquirir la imagen y el de obtener las coordenadas espaciales puede no ser despreciable. Es decir, se pueden producir movimientos en el patrón plano que hagan que la correspondencia obtenida no sea buena. Por dicho motivo se resolvió crear un patrón tridimensional con forma de cubo con el que se pudieran adquirir puntos sin necesidad de mover el patrón.

5.4. Cronología del proceso de calibración

En las etapas iniciales del Proyecto se evaluaron las imágenes obtenidas desde el microscopio y se realizaron las primeras calibraciones utilizando la primera variante.

Durante estas pruebas se utilizó la biblioteca Camera Calibration Toolbox for Matlab (disponible en [7]); se trata de una librería para Matlab que implementa el método de calibración de Bouguet que se explicó en el Capítulo 4. La decisión se fundamentó en que es una librería de uso simple e intuitivo que ha sido utilizada en varios trabajos a nivel académico y es reconocida por sus buenos resultados.

Luego de las primeras experiencias quedó en evidencia la dificultad del proceso de calibración ya que los resultados obtenidos no fueron nada alentadores. Tanto los parámetros intrínsecos como los coeficientes de distorsión adquirieron valores que carecían de sentido físico. En las Tablas 5.1 y 5.2 se muestran como ejemplo algunos valores de los parámetros. Los mismos fueron obtenidos calibrando el microscopio del Hospital de Clínicas con el Camera Calibration Toolbox.

La Tabla 5.1 corresponde a la calibración del microscopio manteniendo el valor del aumento mientras se varía la distancia de trabajo desde 200 a 400 mm, y utilizando un modelo de cámara que incluye los cuatro coeficientes de distorsión del modelo de Bouguet. La resolución de las imágenes adquiridas fue de 640×480 . El error que se muestra es bajo, tanto en la dirección x como en la dirección y , siendo en todos los casos menor al píxel. Este error se calcula proyectando los puntos que se utilizaron en la calibración y comparando esta posición con su ubicación original en la imagen. Como podía suponerse, los coeficientes que modelan la distorsión tangencial (k_3, k_4) fueron muy pequeños. Sin embargo no sucedió lo mismo con los coeficientes que modelan la distorsión radial; en particular se observa que el coeficiente k_2 aumenta considerablemente al aumentar la distancia de trabajo. Este comportamiento no tiene explicación física. Dado que el segundo coeficiente de distorsión k_2 multiplica al término dependiente de r^4 (ver Subsección 4.3.3), es de esperar que su valor sea menor al del primer coeficiente k_1 , pero esto claramente no sucede.

La Tabla 5.2 muestra los resultados que se obtuvieron al calibrar el microscopio con un aumento de 7 a una distancia de trabajo de 200 mm. Al igual que en el caso anterior, los errores de reproyección son muy bajos en todos los casos. Sin embargo se obtuvieron valores sin ningún sentido físico: se ven por ejemplo puntos principales negativos o que superan el tamaño de la imagen.

Tabla 5.1: Resultados obtenidos al calibrar el microscopio del Hospital de Clínicas con un aumento de 3 para varios valores de distancia focal. Se utilizó el Camera Calibration Toolbox for Matlab.

Distancia de trabajo	Punto Principal		Largo Focal	
	c_x	c_y	f_x	f_y
200	251.97	349.02	2880.09	2845.31
240	339.81	472.71	3368.08	3342.26
280	169.79	373.78	4042.97	4004.85
320	312.57	276.83	4725.10	4657.59
360	426.17	361.21	5265.31	5215.15
400	450.73	296.48	5965.33	5905.25
Modelo de distorsión				
Distancia de trabajo	k_1	k_2	k_3	k_4
200	-0.88	0.091	0.01	-0.00
240	-0.43	-18.65	-0.01	-0.02
280	-1.59	26.68	0.00	0.02
320	-1.59	91.88	0.02	-0.00
360	-2.33	140.87	-0.00	-0.03
400	-2.98	360.78	0.01	-0.03
Error (píxeles)				
Distancia de trabajo	e_x	e_y		
200	0.19	0.30		
240	0.16	0.30		
280	0.16	0.30		
320	0.16	0.30		
360	0.16	0.30		
400	0.17	0.25		

Tabla 5.2: Parámetros intrínsecos iniciales

Largo Focal	: fc = [11532.61	11676.44]		
Punto Principal:	cc = [319.5	239.5]		
Distorsión	: kc = [0.00	0.00	0.00	0.00]

Tabla 5.3: Utilizando todo el modelo

Largo Focal	: fc = [12996.10	12835.17]	±	[1080.76	970.99]
Punto Principal:	cc = [378.64	-482.51]	±	[0.00	0.00]
Distorsión	: kc = [-3.25	235.98	0.058	-0.050]	
Error (píxeles)	: err = [0.32670	0.42647]			

Tabla 5.4: Considerando sólo un coeficiente de distorsión radial

Largo Focal	: fc = [13757.86	13034.18]	±	[1085.34	961.36]
Punto Principal:	cc = [450.39	-487.94]	±	[0.00	0.00]
Distorsión	: kc = [0.70419	0.00	0.00	0.00]	
Error (píxeles)	: err = [0.33440	0.43834]			

Tabla 5.5: Sin considerar el modelo de distorsión

Largo Focal	: fc = [11738.27	11822.43]	±	[616.71	637.08]
Punto Principal:	cc = [668.42	-13.86]	±	[0.00	0.00]
Distorsión	: kc = [0.00	0.00	0.00	0.00]	
Error (píxeles)	: err = [0.33408	0.38807]			

Tabla 5.6: Considerando un coeficiente de distorsión y fijando el punto principal en el centro

Largo Focal	: fc = [11532.61	11676.44]	±	[649.51	670.05]
Punto Principal:	cc = [319.5	239.5]	±	[0.00	0.00]
Distorsión	: kc = [0.00	0.00	0.00	0.00]	
Error (píxeles)	: err = [0.33408	0.38807]			

Tabla 5.7: Sin considerar el modelo de distorsión y fijando el punto principal en el centro

Largo Focal	: fc = [11735.03	11771.28]	±	[731.30	743.19]
Punto Principal:	cc = [319.5	239.5]	±	[0.00	0.00]
Distorsión	: kc = [0.00	0.00	0.00	0.00]	
Error (píxeles)	: err = [0.33301	0.44627]			

5.4.1. Sobre el método de evaluación de la calibración

Al mostrar los resultados de la Tabla 5.2 se dijo que los parámetros obtenidos carecían de sentido físico, pero sin embargo se muestra en todos los casos un error subpíxelico. ¿Cómo es esto posible?

La respuesta es inherente a la forma de calcular el error que implementa Bouguet. Al calibrar se estiman los parámetros intrínsecos pero también se ajustan los parámetros extrínsecos, es decir la ubicación del sistema de referencia de cada uno de los daderos respecto al sistema de coordenadas de la cámara. La práctica mostró que en aquellos casos en que no se logra una buena estimación de los parámetros intrínsecos, tampoco se consiguen parámetros extrínsecos que se adecuen a la realidad.

Habitualmente, para validar el procedimiento de calibración, se suele tomar una imagen del patrón que no haya sido utilizada para calibrar, y sobre ella se re proyectan las posiciones espaciales de las esquinas haciendo uso de los parámetros intrínsecos encontrados. Sin embargo, las posiciones espaciales se obtienen a partir de la imagen calculando el conjunto de parámetros extrínsecos que minimice el error de reproyección. Dado que se deja variar libremente los parámetros extrínsecos (siempre y cuando se trate de un movimiento rígido), aún cuando la estimación de los parámetros intrínsecos no sea la adecuada es posible encontrar una posición del damero relativa a la cámara que, combinada con esos intrínsecos, resulte en un bajo error de reproyección.

Como experiencia se tomó una imagen del patrón que no había sido utilizada para calibrar y se calculó el error de reproyección de la forma descripta, obteniéndose un error subpíxelico. Luego se modificaron algunos parámetros intrínsecos y utilizando la misma imagen se volvió a calcular el error de reproyección. Esta vez también se obtuvo un error subpíxelico. Esta experiencia pone en evidencia la fragilidad del método de evaluación de la calibración. Se constató por ejemplo que una variación en los valores del punto principal se compensa con una traslación en el plano de la imagen, o que un aumento en los largos focales tiene como contrapartida un acercamiento a lo largo del eje óptico.

En la Sección 5.7 se describe el procedimiento utilizado para evaluar el método de calibración implementado en este proyecto. La principal diferencia con el método habitual es que no se calculan los parámetros extrínsecos, sino que se utiliza la posición real de las esquinas del patrón dada por el localizador. Esto evita los problemas mencionados, dado que ahora los parámetros extrínsecos están fijos.

5.4.2. Primeras decisiones sobre el modelo

Basado en la observación de las imágenes obtenidas del microscopio se resolvió no considerar el modelo distorsión de la cámara. La consulta bibliográfica realizada en ese sentido [7] mostró que esta es una decisión que se suele tomar, debido a dos razones:

- la distorsión provocada por el microscopio es muy baja.

- en el procedimiento de realidad aumentada es necesario actualizar la información en tiempo real; una corrección de la imagen debido a la distorsión provoca una reducción en la tasa de refresco de pantalla.

Otra decisión que se tomó fue la de fijar el punto principal en el centro de la imagen. Al realizar pruebas en el laboratorio con cámaras “domésticas” se obtenían modelos razonables de distorsión y valores de puntos principales cercanos al centro de la imagen. No obstante, como se mostró en la Tabla 5.2 en las pruebas realizadas con el microscopio se obtenían valores de puntos principales que carecían totalmente de sentido. Se realizó una consulta bibliográfica al respecto [7], que mostró que el punto principal es el parámetro más difícil de estimar del modelo y se recomienda fijarlo en el centro de la imagen si no se cuenta con buena información.

5.4.3. Automatización de la adquisición de datos

El procesamiento de la información de calibración insumía mucho tiempo por lo que se resolvió automatizar el proceso; esto conlleva las siguientes ventajas: reducción del tiempo de calibración, posibilidad de analizar los resultados y realizar modificaciones durante las visitas a los hospitales. Para la automatización se utilizó la librería OpenCV, disponible para descargar en [59]. Ésta es una librería de visión por computadora, de código abierto, escrita en C y C++. Entre otras aplicaciones, OpenCV es útil para calibración de cámaras, ya que incluye una implementación del algoritmo de Bouguet que se introdujo en el Capítulo 4.

Para la automatización, lo que se hizo fue:

- Extraer del video las imágenes que se utilizan para calibrar en tiempo real. Hasta el momento se debía guardar el video y luego extraer del mismo los cuadros que se utilizarían para la calibración. Actualmente se extraen las imágenes del video presionando una tecla; se evita de esta forma todo el procesamiento posterior de la señal de video.
- Detectar automáticamente los puntos de calibración de cada imagen. El Toolbox de Bouguet no tiene detección automática de esquinas, sino que para cada imagen del damero que se vaya a utilizar en la calibración hay que marcar con el ratón las cuatro esquinas exteriores.

Detección de esquinas con el patrón plano

Se utilizó la función `cvFindChessBoardCorners`; dicha función recibe como parámetro una imagen de un damero en escala de grises y el número de esquinas interiores del mismo. Si se piensa en un tablero de ajedrez, por ejemplo, este tiene 7×7 esquinas interiores. Devuelve la ubicación aproximada de las esquinas pero no de cualquier forma, sino ordenadas, como se muestra en la Figura 5.4. Esto es muy importante para poder establecer la correspondencia con la ubicación espacial de los puntos.

Para alcanzar precisión subpíxelica en la detección se utiliza el algoritmo de detección de esquinas de Harris [27] que se implementa en la función `cvFindCornerSubPix`.

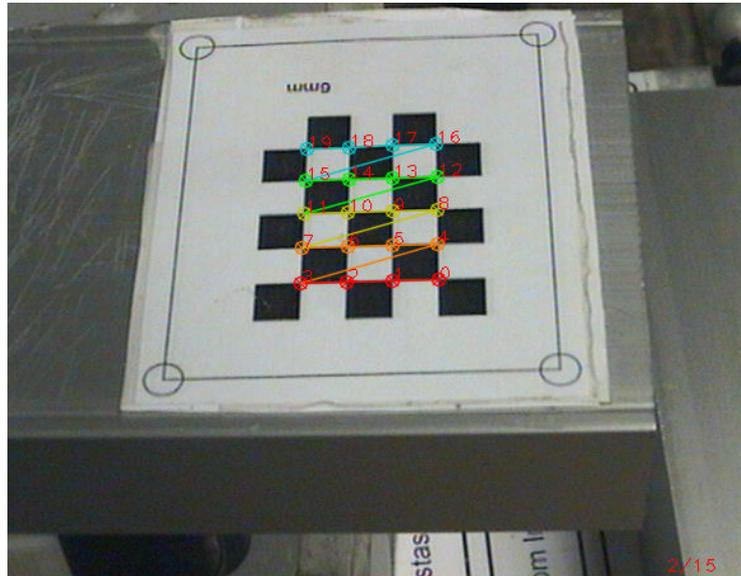


Figura 5.4: Detección automática de esquinas utilizando OpenCV

Adaptación de la función de OpenCV para detectar esquinas de un patrón cúbico

Se implementó la función `cvFindChessCubeCorners` que extrae en forma semiautomática las esquinas de los dameros ubicados en las caras de un patrón de calibración cúbico. Recibe como parámetro una foto del patrón y el número de esquinas presentes en cada damero y devuelve la ubicación de las esquinas en píxeles. El algoritmo consta de los siguientes pasos:

1. Se transforma la imagen a blanco y negro.
2. Se marcan con el ratón cuatro vértices de las caras visibles del cubo en el siguiente orden: vértice superior izquierdo, vértice superior central, vértice superior derecho y vértice inferior central.
3. A partir de la imagen original se segmentan tres regiones, conteniendo cada una de ellas una cara del cubo.
4. Para cada región se construye una imagen con la región segmentada y se rellena el resto de la imagen con el valor medio de gris de la región.
5. Para cada una de las imágenes construidas se detectan las esquinas utilizando la función `cvFindChessBoardCorners` de OpenCV.

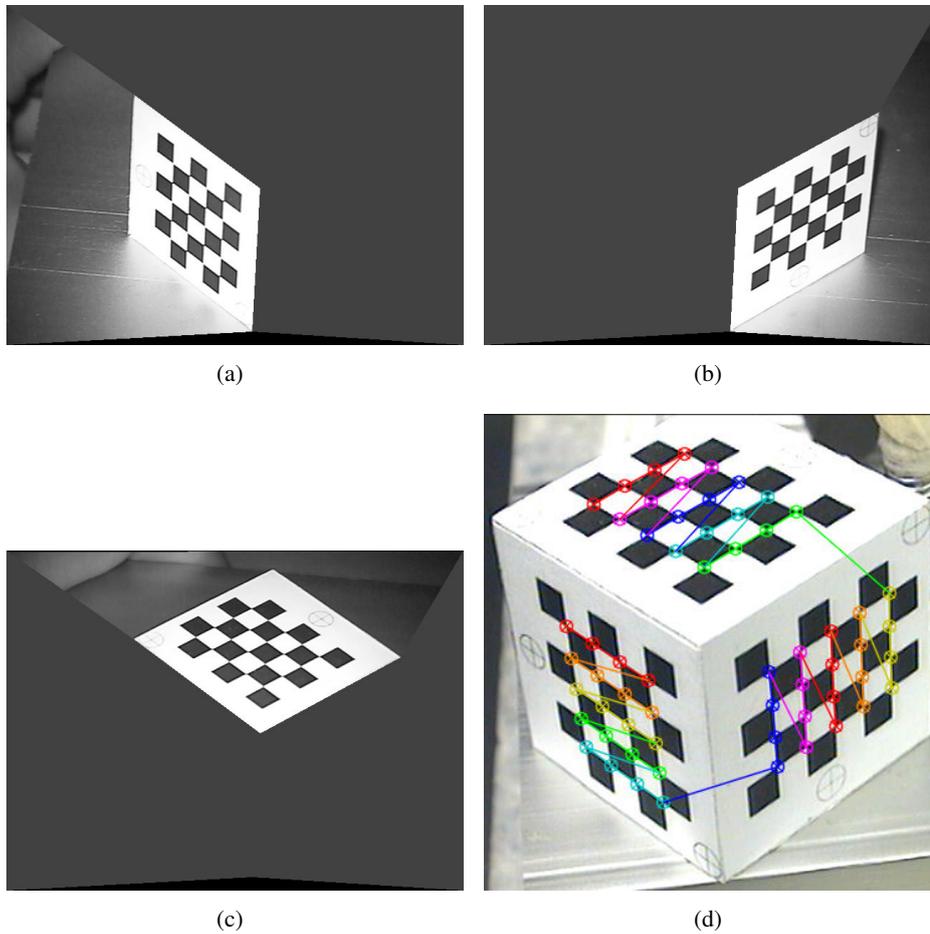


Figura 5.5: Calibración con patrón cúbico. 5.5(a) a 5.5(c) Segmentación de las caras izquierda, derecha y superior. 5.5(d) Detección automática de las esquinas del patrón mediante la función `cvFindChessCubeCorners`. Se obtienen las coordenadas de las esquinas de forma ordenada.

5.5. Procedimiento de calibración

El procedimiento de calibración finalmente establecido es el siguiente:

- Se selecciona entre el patrón plano y el patrón cúbico para realizar la calibración.
- Se calibra el patrón para conocer la posición de sus esquinas a partir de la posición de sus marcadores adheridos (por los detalles, ver Apéndice B).
- Se establece la correspondencia entre los puntos espaciales y los de la imagen. Para cada calibración, un número de imágenes son adquiridas desde diferentes posiciones relativas entre el microscopio y el patrón. Esto es necesario por

diferentes razones: incrementar el número de puntos correspondientes usados en la calibración, para reducir errores debido al ruido introducido en el seguimiento, para obtener una mejor dispersión de los puntos de calibración a lo largo del eje óptico y asegurar que el procedimiento de calibración sea más estable desde el punto de vista numérico.

- Se validan los resultados.

5.6. Resultados de Calibraciones

Las Tablas 5.8 a 5.10 muestran los resultados obtenidos al calibrar el microscopio del Hospital de Tacuarembó. Se realizaron tres pruebas manteniendo fijos los valores de aumento, distancia de trabajo y la posición de los marcadores adheridos al microscopio.

En la Tabla 5.8 se observa cómo la estimación inicial de los parámetros intrínsecos no es buena, es decir, no se logró una buena estimación de los intrínsecos a partir del cálculo de las homografías que llevan el damero detectado a un damero ideal. Sin embargo, tomando esa solución como semilla se lograron parámetros de calibración muy razonables; este ejemplo es una clara muestra de que, en general, la calibración con objetos no coplanares funciona mejor que la calibración con planos.

Las Tablas 5.9 y 5.10 muestran calibraciones en las que sí se logró una buena estimación inicial de los parámetros intrínsecos; en general sucede cuando se toman posiciones de los daderos desde distintos ángulos y que barren el espacio de la imagen; aunque no siempre es así.

Un aspecto que vale la pena mencionar es que durante la calibración se determina la forma en que la cámara proyecta los puntos del espacio, es decir la matriz de proyección, que es la acción conjunta de los parámetros intrínsecos (A) y de los extrínsecos (que coinciden con H^{-1} si se considera como sistema de referencia la posición de los marcadores adheridos al microscopio). Estas dos matrices están muy relacionadas y no es sencillo, dada las particularidades del microscopio, desacoplar la acción de cada una de ellas. Si bien aquí se presentan sólo tres calibraciones, los resultados sugieren lo siguiente: las coordenadas x y z de las traslaciones obtenidas en las matrices H (última columna) varían en menor medida que la coordenada y . Este es un fenómeno característico; la razón por la que sucede es que, dado el escaso campo de visión del microscopio, no se pueden recolectar muchos datos en la dirección del eje óptico (que en este caso coincide con el eje y del sistema de referencia de los marcadores). Esta escasa variación de profundidad en los objetos no permite que el modelo discierna si las dimensiones de los objetos en la imagen se deben a la posición de la cámara o al aumento del lente. Pensar en el siguiente ejemplo: es posible obtener exactamente la misma representación de un objeto en la imagen si se lo mira a dos metros de distancia con un determinado aumento que si se lo mira a cuatro metros con el doble de aumento... ¿cómo saber cuál es la situación? La solución pasa por variar la profundidad del objeto: si se mueve el mismo en la dirección del eje óptico ya no se verán iguales; cuanto más se mueva, más diferencias

van a haber y por lo tanto se va a poder desacoplar mejor entre la posición de la cámara y el aumento de la misma.

Si se observan las traslaciones en y obtenidas y los valores de largo focal del modelo pinhole se ve que cuanto más cerca se ubica a la cámara del objeto (mayor coordenada y) menores son los valores de largo focal obtenidos.

Intrínsecos iniciales			Intrínsecos		
14133	0	319.5	4353.3	0	319.5
0	14346	239.5	0	4312.7	239.5
0	0	1	0	0	1
H					
	0.996932	0.034357	0.070331	-74.67	
	0.070842	-0.013903	-0.997391	-9.91	
	-0.033289	0.999313	-0.016294	118.50	

Tabla 5.8: Calibración del microscopio del Hospital de Tacuarembó. Resultados de la Calibración 1. Aún partiendo de una estimación inicial mala para los parámetros intrínsecos se lograron buenos resultados

Intrínsecos iniciales			Intrínsecos		
3991.80	0	319.5	4182.70	0	319.5
0	3974.65	239.5	0	4173.17	239.5
0	0	1	0	1	0
H					
	0.996752	0.038075	0.070965	-75.03	
	0.076720	-0.003633	-0.997046	7.15	
	-0.037074	0.999194	-0.015368	118.66	

Tabla 5.9: Calibración del microscopio del Hospital de Tacuarembó. Resultados de la Calibración 2.

Intrínsecos iniciales			Intrínsecos		
4182.70	0	319.5	4130.73	0	319.5
0	4173.17	239.5	0	4122.33	239.5
0	0	1	0	1	0
H					
	0.995619	0.042798	0.083134	-77.70	
	0.083761	-0.013070	-0.996400	10.83	
	-0.041558	0.998998	-0.016597	119.54	

Tabla 5.10: Resultados de la Calibración 3

5.7. Evaluación del proceso de calibración

Una vez calibradas las cámaras o el microscopio, se hizo necesario definir un procedimiento que permitiera validar los resultados obtenidos y tener una medida del error cometido al modelar la óptica con los parámetros encontrados. En términos generales, el procedimiento consistió en comparar la posición sobre una imagen de ciertos puntos de control (las esquinas del patrón de calibración plano) con su proyección calculada a partir de la posición espacial de los mismos puntos, dada por el localizador ¹, y haciendo uso de la matriz obtenida por medio del proceso de calibración.

Las cámaras de laboratorio fueron calibradas con el patrón plano y con el patrón cúbico. Los resultados obtenidos en ambos casos fueron comparables, con errores en el entono de los 3 píxeles. En la Subsección 5.7.2 se presentan en detalle las experiencias realizadas con las cámaras.

La Subsección 5.7.3 resume el trabajo con el microscopio. Todos los datos corresponden a pruebas hechas sobre el del Hospital de Tacuarembó, dado que la calibración de los otros dos presentó varios problemas, como se verá al final de este capítulo. Con el procedimiento propuesto, los errores de reproyección se evaluaron para dos configuraciones diferentes del microscopio:

Configuración 1 Aumento 2, Distancia de trabajo 300 mm.

Configuración 2 Aumento 3, Distancia de trabajo 300 mm.

Para las dos configuraciones, los errores de reproyección de las esquinas del patrón estuvieron en general en torno a los 3 o 4 píxeles, aunque se lograron en algunos casos errores menores e incluso menores al píxel. La Figura 5.6 muestra la posición detectada de las esquinas junto con las reproyecciones de las mismas, para una de las experiencias en las que los resultados fueron satisfactorios.

No se presentan resultados de la calibración con patrón cúbico; su uso con el microscopio fue descartado dado que no se lograron buenos resultados.

En el resto de la sección se describe detalladamente el procedimiento seguido para la evaluación y se reúnen algunos valores representativos de los resultados obtenidos durante el trabajo con los microscopios y cámaras de laboratorio. Finalmente se resumen las principales conclusiones sobre el trabajo realizado.

¹Como se comentó en la Subsección 5.4.1, el hecho de utilizar la posición espacial del patrón dada por el localizador diferencia a este método de la forma habitual en que se evalúa la calibración, en la cual los parámetros extrínsecos son calculados de manera de minimizar el error de reproyección.

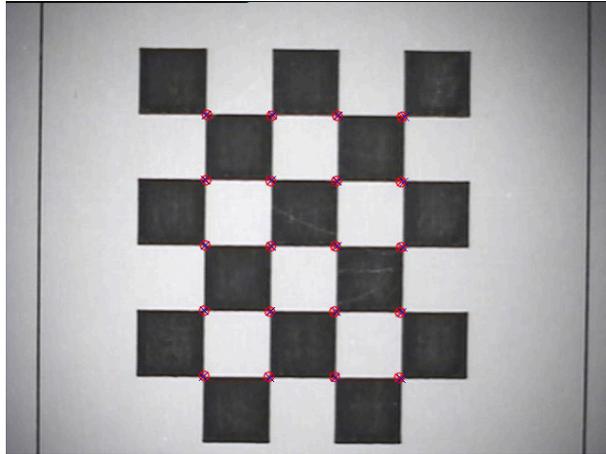


Figura 5.6: Imagen del patrón de calibración utilizada para el procedimiento de evaluación. Las posiciones detectadas de las esquinas se marcan con \times , mientras que con \otimes se indican las reproyecciones. El error de reproyección en este caso fue de 1.52 píxeles.

5.7.1. Descripción del procedimiento de evaluación

El algoritmo de calibración fue evaluado comparando la posición de ciertos puntos de control sobre una imagen con su reproyección, calculada a partir de la posición espacial de los mismos puntos y haciendo uso de la matriz de proyección obtenida por medio del proceso de calibración.

Como objeto de referencia se utilizó un patrón plano tipo damero y como puntos de control sus esquinas interiores. Sobre una imagen de este patrón tomada con la óptica calibrada, se detectaron las esquinas por medio de la función `cvFindChessboardCorners`, dando lugar al conjunto de coordenadas "detectadas" (x_{id}, y_{id}) .

Como se describió en las secciones anteriores, a través del proceso de calibración se obtienen dos matrices:

- la matriz de parámetros intrínsecos A , tal que al multiplicar las coordenadas normalizadas de un punto por esta matriz se tienen las coordenadas de la proyección del mismo sobre el plano de la imagen.
- la matriz de transformación H que relaciona los sistemas de coordenadas de la cámara y de los marcadores adheridos a ella.

Por lo tanto, teniendo las coordenadas de las esquinas del damero en el sistema de coordenadas de la herramienta de la cámara, y haciendo uso de la matriz H , se obtienen las coordenadas en el sistema de referencia de la cámara; luego a través de la matriz A se proyectan las esquinas sobre el plano de la imagen, dando lugar a las coordenadas (x_{ip}, y_{ip}) .

La posición espacial de cada esquina del damero se extrajo de dos formas diferentes:

- a través del puntero, tocando cada esquina con la punta calibrada, con lo cual se obtienen directamente las coordenadas de los puntos en el sistema de referencia asociado al localizador.
- a través de una herramienta adherida de forma rígida al patrón. Con esta herramienta ubicada dentro del volumen visible por el localizador, la posición espacial de la misma es conocida, y teniendo el conjunto damero-herramienta previamente calibrado se conoce la posición espacial de las esquinas del damero referidas al sistema de coordenadas del localizador. En el Apéndice B se explica el procedimiento de calibración damero-herramienta.

En ambos casos, dado que se cuenta con la posición de la herramienta adherida a la cámara en el sistema de coordenadas del localizador, se conoce la posición de las esquinas en el sistema de coordenadas asociado a esta herramienta.

Idealmente, las coordenadas (x_{ip}, y_{ip}) deberían coincidir con las coordenadas "detectadas" sobre la imagen (x_{id}, y_{id}) , pero es justamente la diferencia entre ellas lo que se busca evaluar. Los algoritmos de calibración buscan minimizar esta diferencia sobre las imágenes que se le pasan como información, por lo que es de esperar que sobre ellas el error sea muy pequeño. Sin embargo, lo importante es evaluar la validez de la calibración en posiciones arbitrarias dentro del campo de visión de la cámara, por lo que se hace necesario para la evaluación tomar imágenes del patrón en posiciones que no hayan sido utilizadas para calibrar.

Siguiendo el procedimiento anterior, para cada imagen del patrón se obtuvieron dos conjuntos de coordenadas (en píxeles) correspondientes: $(x_{id}, y_{id}) \leftrightarrow (x_{ip}, y_{ip})$, con $i = 1 \dots N$, siendo N la cantidad de esquinas interiores del damero utilizado y donde el subíndice d corresponde a las coordenadas *detectadas* sobre la imagen y el subíndice p a las *proyectadas*. El error RMS entre ambos conjuntos, dado por:

$$E_{RMS} = \sqrt{\frac{\sum_{i=1}^N (x_{id} - x_{ip})^2 + (y_{id} - y_{ip})^2}{N}}$$

fue el parámetro utilizado como medida del desempeño de los algoritmos de calibración. Debe notarse que en la ecuación anterior todas las coordenadas involucradas están medidas en píxeles, por lo que el error RMS también está dado en píxeles.

5.7.2. Experiencias con cámara de laboratorio

En la evaluación de la calibración de la cámara de laboratorio se utilizó como objeto de referencia un patrón de 4×5 esquinas interiores con casillas de 8 mm de lado y se aplicó el procedimiento descrito sobre 10 a 15 imágenes del patrón ubicado a diferentes distancias de la cámara, aproximadamente paralelo al plano de la imagen y centrado en éste, de forma de barrer la zona en foco. La prueba se repitió para distintos aumentos de la cámara.

Para calibrar con el patrón plano se utilizaron 15 imágenes del mismo, y en el caso del patrón tridimensional se tomaron imágenes en tres posiciones diferentes; en ambos casos la resolución de las imágenes fue de 640×480 píxeles. Las Tablas 5.11 a 5.14 resumen los resultados obtenidos con ambos métodos. La primera columna muestra la distancia aproximada del patrón al centro de la cámara y las siguientes el error RMS de reproyección obtenido a esa distancia. Los valores de los parámetros de calibración resultantes para cada aumento y cada tipo de patrón se resumen en la Tabla 5.15

Aumento: 12.5		
Distancia aproximada al centro de la cámara (cm)	Calibración con damero plano	Calibración con cubo
26	1.73	12.55
30	1.12	9.49
32	0.82	7.90
34	1.75	4.68
36	1.69	2.197
37	2.52	1.805
38	2.99	2.31
39	1.86	2.33
40	2.44	1.25
41	2.35	2.06
42	1.79	1.44
44	1.50	1.39
45	2.39	2.30
47	2.27	2.87
50	2.36	3.90

Tabla 5.11: Error RMS en la reproyección de las esquinas del patrón de calibración, para una de las cámaras del laboratorio. La distancia del patrón a la cámara varía entre 25 y 50 *cm*, mientras se mantiene fijo el aumento en 12,5.

Aumento: 15		
Distancia aproximada al centro de la cámara (cm)	Calibración con damero plano	Calibración con cubo
26	3.78	4.75
30	3.87	1.96
31	2.02	1.87
32	3.23	3.33
33	3.17	2.27
34	2.14	4.24
35	1.87	1.82
36	2.17	3.54
37	2.63	0.84
38	1.85	4.10
39	2.94	2.46
40	3.42	2.46
41	2.93	3.32
42	3.70	2.90
43	3.91	3.87
45	3.19	2.73
49	2.77	3.12
51	6.92	1.76
55	4.89	4.47

Tabla 5.12: Error RMS en la reproyección de las esquinas del patrón de calibración, para una de las cámaras del laboratorio. La distancia del patrón a la cámara varía entre 25 y 55 *cm*, mientras se mantiene fijo el aumento en 15.

Aumento: 20		
Distancia aproximada al centro de la cámara (cm)	Calibración con damero plano	Calibración con cubo
40	1.68	2.19
42	2.79	3.40
44	1.61	0.96
46	2.56	2.60
48	3.18	2.34
50	3.60	3.58
51	3.40	1.06
52	3.49	3.11
53	3.72	3.33
54	1.92	6.44
55	3.20	6.20
56	3.40	6.97
57	3.18	5.67
58	1.28	7.39
60	2.66	7.97

Tabla 5.13: Error RMS en la reproyección de las esquinas del patrón de calibración, para una de las cámaras del laboratorio. La distancia del patrón a la cámara varía entre 40 y 60 *cm*, mientras se mantiene fijo el aumento en 20.

Aumento: 30		
Distancia aproximada al centro de la cámara (cm)	Calibración con damero plano	Calibración con cubo
60	4.87	7.36
62	1.85	8.74
63	3.32	10.43
64	3.04	8.35
65	3.32	5.33
67	1.42	7.68
69	3.80	7.23
70	4.07	3.28
72	1.11	7.18
71	3.19	4.90
73	1.18	4.89
75	5.30	5.20
77	2.71	9.25
76	1.50	7.26
80	4.11	2.70
79	6.08	7.12
83	2.15	6.21

Tabla 5.14: Error RMS en la reproyección de las esquinas del patrón de calibración, para una de las cámaras del laboratorio. La distancia del patrón a la cámara varía entre 25 y 55 *cm*, mientras se mantiene fijo el aumento en 30.

Calibración con patrón plano						
Aumento 12.5						
<i>A</i>			<i>H</i>			
1485.23	0	319.5	-0.009033	-0.999841	0.01538	-50.06
0	1484.92	239.5	0.003928	-0.015416	-0.999873	-83.35
0	0	1	0.999951	-0.008971	0.004067	44.05
Aumento 15						
<i>A</i>			<i>H</i>			
1872.42	0	319.5	-0.008543	-0.999816	0.017202	-51.06
0	1887.59	239.5	0.00249	-0.017224	-0.999849	-58.65
0	0	1	0.99996	-0.008498	0.002636	44.50
Aumento 20						
<i>A</i>			<i>H</i>			
2658.71	0	319.5	-0.008508	-0.999714	0.022356	-53.56
0	2706.42	239.5	0.006451	-0.022411	-0.999728	-22.32
0	0	1	0.999943	-0.008362	0.00664	42.62
Aumento 30						
<i>A</i>			<i>H</i>			
4109.96	0	319.5	-0.002475	-0.999741	0.022624	-53.44
0	4112.50	239.5	0.001886	-0.022628	-0.999742	-25.77
0	0	1	0.999995	-0.002432	0.001942	45.88
Calibración con patrón cúbico						
Aumento 12.5						
<i>A</i>			<i>H</i>			
1402.17	0	319.5	-0.008309	-0.999852	0.015049	-51.36
0	1400.38	239.5	0.018832	-0.015204	-0.999707	-105.60
0	0	1	0.999788	-0.008023	0.018955	40.37
Aumento 15						
<i>A</i>			<i>H</i>			
1709.45	0	319.5	-0.003407	-0.999929	0.011418	-50.79
0	1706.97	239.5	0.006079	-0.011439	-0.999916	-90.80
0	0	1	0.999976	-0.003338	0.006117	43.65
Aumento 20						
<i>A</i>			<i>H</i>			
2438.77	0	319.5	-0.010273	-0.999779	0.018318	-50.89
0	2435.73	239.5	0.003171	-0.018351	-0.999827	-71.40
0	0	1	0.999942	-0.010213	0.003359	44.14
Aumento 30						
<i>A</i>			<i>H</i>			
3903.67	0	319.5	-0.007309	-0.999767	0.020295	-50.92
0	3894.21	239.5	0.006326	-0.020341	-0.999773	-62.79
0	0	1	0.999953	-0.007179	0.006474	41.83

Tabla 5.15: Parámetros intrínsecos A y transformación H resultantes de calibrar una cámara del laboratorio, con patrón plano y cúbico.

5.7.3. Experiencias con el microscopio

La evaluación de la calibración del microscopio se hizo también con patrones planos de 4×5 esquinas interiores, con cuadrados de 6 mm o 4 mm de lado dependiendo de la configuración de la óptica (valor del aumento y de la distancia de trabajo). El criterio adoptado fue colocar el patrón paralelo al plano de la imagen y desplazarlo en la dirección del eje óptico, siempre dentro de la región en foco. De esta manera se tomaron varias imágenes sobre las que se corrió el algoritmo descrito.

En las Tablas 5.16 a 5.19 se muestran los valores del error RMS resultantes de aplicar el procedimiento de evaluación luego de calibrar el microscopio del Hospital de Tacuarembó.

Los datos que se presentan corresponden a dos configuraciones diferentes del sistema óptico:

Configuración 1 Aumento 2, Distancia de trabajo 300 mm. En esta configuración se trabajó con patrones de casillas de 6 mm.

Configuración 2 Aumento 3, Distancia de trabajo 300 mm. En esta configuración se trabajó con patrones de casillas de 4 mm.

A su vez, se muestran los resultados para dos calibraciones diferentes de cada configuración, es decir, utilizando dos conjuntos de parámetros obtenidos en dos ejecuciones diferentes del procedimiento de calibración. Los valores de todos los parámetros pueden consultarse en la Tabla 5.20.

Las posiciones del patrón para la evaluación fueron tomadas con una separación a lo largo del eje óptico de aproximadamente 4 mm, siempre en un entorno de los 300 mm correspondiente a la distancia de trabajo fijada. En las tablas se muestran los valores ordenados, comenzando por la posición más alejada hasta terminar en la más cercana a la cámara.

Las figuras 5.7 a 5.10 muestran las posiciones del patrón utilizadas para la evaluación, y sobre ellas se han superpuesto:

- las posiciones de las esquinas *detectadas* mediante la función `cvFindChessBoardCorners`, representadas por el símbolo \times .
- las posiciones de las esquinas *proyectadas*, representadas por el símbolo \otimes .

En todos los casos la calibración fue hecha con 15 imágenes del patrón plano con una resolución de 640×480 píxeles.

Configuración 1: Aumento 2, Distancia de trabajo 300 mm

Configuración 1: Aumento 2, Distancia de trabajo 300 mm	
Posición	Error RMS (píxeles)
1	4.52
2	4.44
3	3.10
4	1.86
5	0.92
6	3.80
7	2.36
8	1.52

Tabla 5.16: Error RMS en la reproyección de puntos, aumento 2, distancia de trabajo 300 mm, Conjunto de parámetros 1. Para estas imágenes, un pixel equivale aproximadamente a 0.089 mm.

Configuración 1: Aumento 2, Distancia de trabajo 300 mm	
Posición	Error RMS (píxeles)
1	10.60
2	6.64
3	5.20
4	1.30
5	6.47
6	2.60
7	2.85
8	3.42

Tabla 5.17: Error RMS en la reproyección de puntos, aumento 2, distancia de trabajo 300 mm, Conjunto de parámetros 2. Para estas imágenes, un pixel equivale aproximadamente a 0.085 mm.

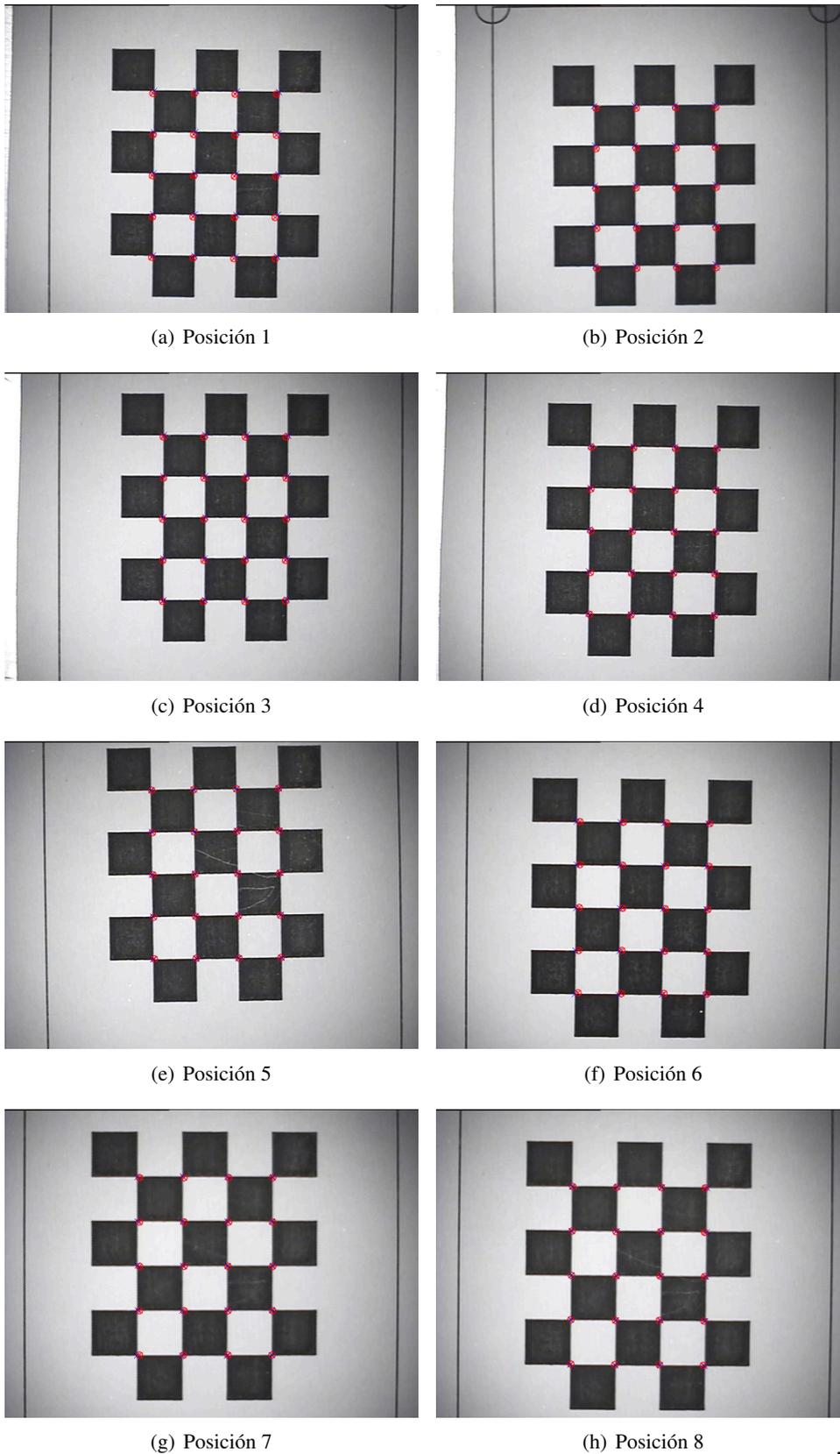


Figura 5.7: Error en la reproyección de puntos, aumento 2, distancia de trabajo 300 mm, Conjunto de parámetros 1. Con el símbolo \times se marcan las esquinas *detectadas*, mientras que el símbolo \otimes corresponde a las esquinas *proyectadas*.

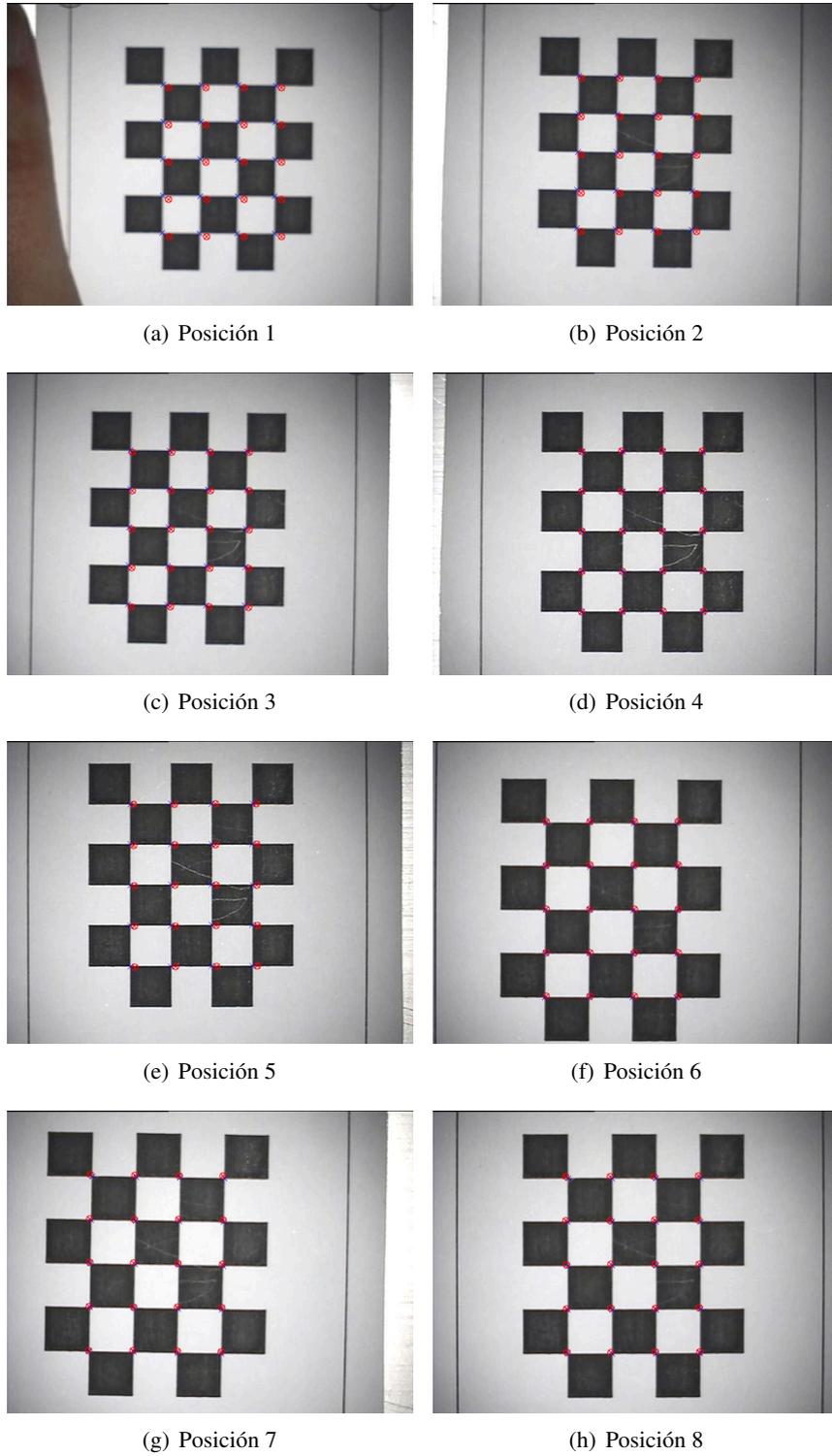


Figura 5.8: Error en la reproyección de puntos, aumento 2, distancia de trabajo 300 mm, Conjunto de parámetros 2. Con el símbolo \times se marcan las esquinas *detectadas*, mientras que el símbolo \otimes corresponde a las esquinas *proyectadas*.

Configuración 2: Aumento 3, Distancia de trabajo 300 mm

Configuración 2: Aumento 3, Distancia de trabajo 300 mm	
Posición	Error RMS (píxeles)
1	4.70
2	5.87
3	10.09
4	7.06
5	8.36
6	11.36

Tabla 5.18: Error RMS en la reproyección de puntos, aumento 3, distancia de trabajo 300 mm, Conjunto de parámetros 1. Para estas imágenes, un pixel equivale aproximadamente a 0.062 mm.

Configuración 2: Aumento 3, Distancia de trabajo 300 mm	
Posición	Error RMS (píxeles)
1	3.55
2	7.11
3	2.32
4	4.43
5	6.47
6	1.89
7	4.74
8	2.11

Tabla 5.19: Error RMS en la reproyección de puntos, aumento 3, distancia de trabajo 300 mm, Conjunto de parámetros 2. Para estas imágenes, un pixel equivale aproximadamente a 0.091 mm.

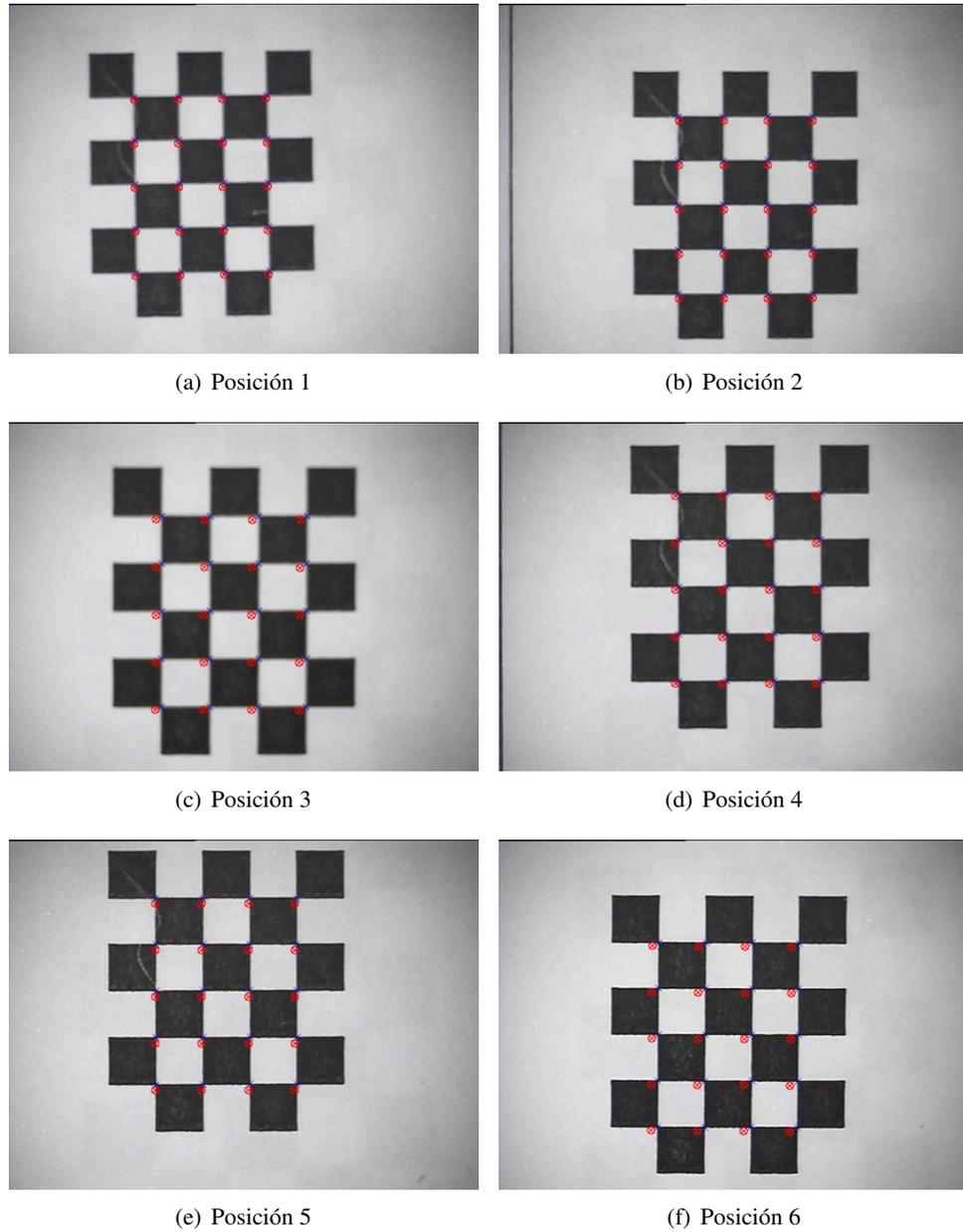


Figura 5.9: Error en la reproyección de puntos, aumento 3, distancia de trabajo 300 mm, Conjunto de parámetros 1. Con el símbolo \times se marcan las esquinas *detectadas*, mientras que el símbolo \circ corresponde a las esquinas *proyectadas*.

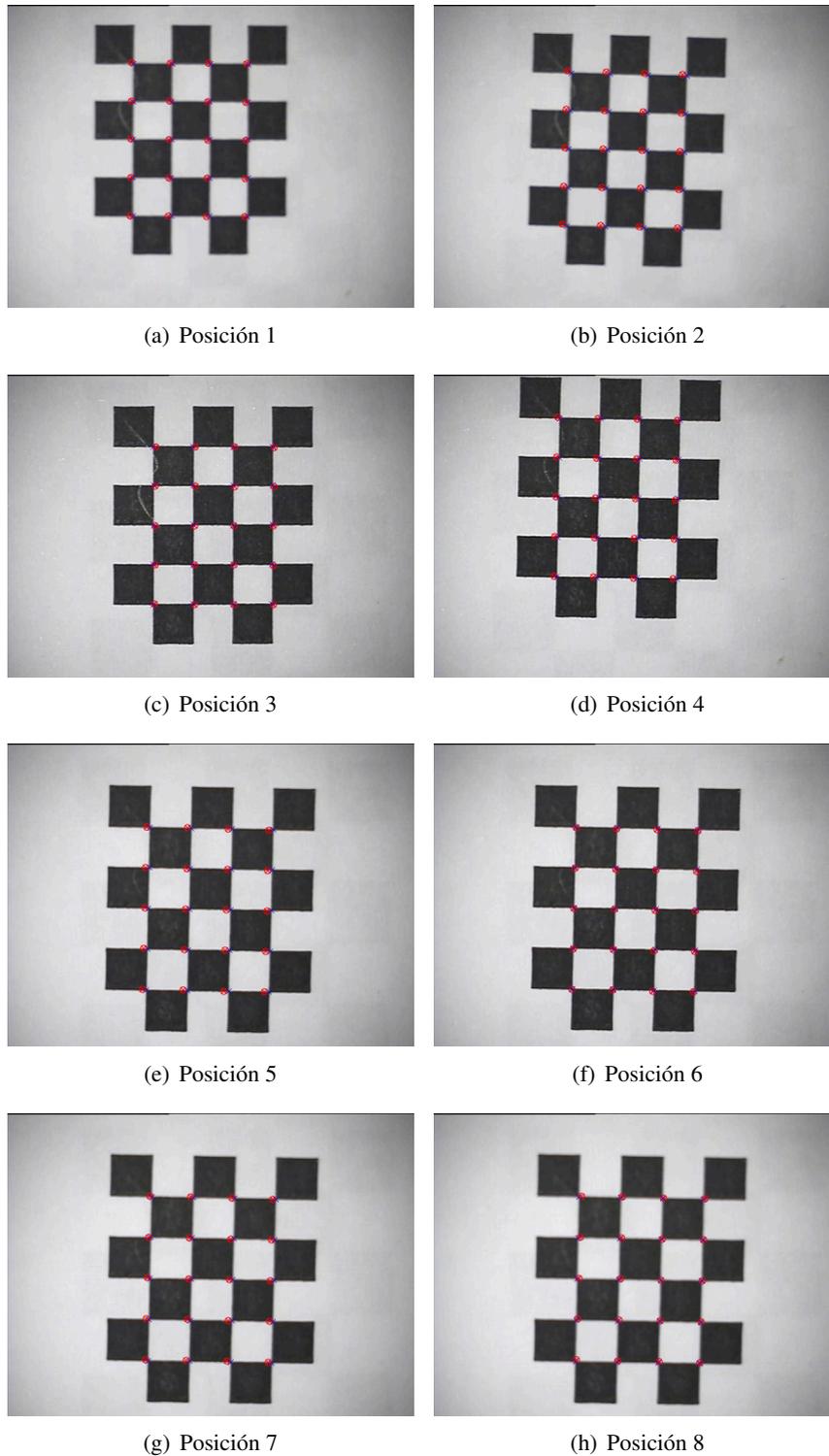


Figura 5.10: Error en la reproyección de puntos, aumento 2, distancia de trabajo 300 mm, Conjunto de parámetros 2. Con el símbolo \times se marcan las esquinas *detectadas*, mientras que el símbolo \otimes corresponde a las esquinas *proyectadas*.

Intrínsecos (A)			Transformación mundo-cámara (H)			
Configuración 1						
4212.61	0	319.5	0.997777	0.063145	0.021293	-72.790062
0	4241.73	239.5	0.022387	-0.016671	-0.99961	28.469131
0	0	1	-0.062766	0.997865	-0.018048	122.787048
4107.62	0	319.5	0.995903	0.045824	0.077961	-72.155014
0	4101.87	239.5	0.078307	-0.005787	-0.996913	1.438246
0	0	1	-0.045231	0.998933	-0.009351	128.392731
Configuración 2						
7096.37	0	319.5	0.998301	0.055314	0.018339	-73.404678
0	7138.91	239.5	0.019214	-0.015323	-0.999698	85.05822
0	0	1	-0.055016	0.998351	-0.01636	122.204315
7515.77	0	319.5	0.997909	0.058878	0.026662	-77.859024
0	7530.93	239.5	0.027804	-0.018669	-0.999439	110.313385
0	0	1	-0.058347	0.998091	-0.020267	124.698586

Tabla 5.20: Parámetros de calibración correspondientes a las Configuraciones 1 y 2.

5.7.4. Conclusiones

Si bien en las diferentes visitas a los Hospitales se calibró la óptica de los tres microscopios, las pruebas para la evaluación fueron hechas sobre el microscopio del Hospital de Tacuarembó. Los principales problemas que presentaron los microscopios del Hospital de Clínicas y del Hospital Maciel fueron de tipo mecánico relacionados con el acople entre la óptica y las cámaras. El desgaste de los adaptadores que une ambas partes impide que las cámaras queden firmemente adheridas al microscopio, lo que claramente tiene consecuencias negativas en esta aplicación. Por otro lado, los adaptadores reducen el área que se proyecta sobre la imagen: si se compara la vista que se tiene desde el microscopio con la que se observa en la imagen, se puede comprobar que en esta última se han perdido las partes más exteriores. Esto se da en los tres microscopios, pero es especialmente notorio en el del Hospital de Clínicas. La Figura 5.11 es una imagen de uno de los primeros patrones de calibración, tomada con el microscopio y la cámara de este hospital. Puede verse cómo la parte superior queda recortada, perdiéndose una parte del campo de visión del microscopio.

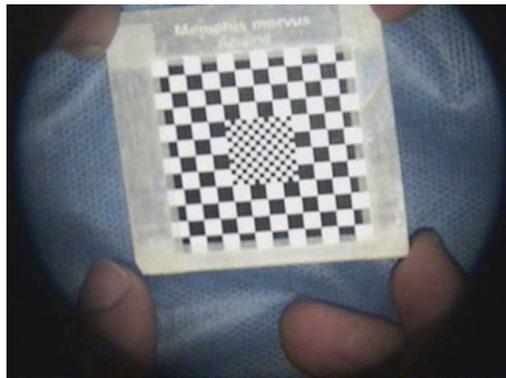


Figura 5.11: Imagen de un patrón de calibración tomada con el microscopio y cámara del Hospital de Clínicas.

De la Figura 5.11 puede hacerse otra observación importante. Fácilmente se comprueba que la imagen proveniente del microscopio no está centrada respecto a la imagen que se extrae de la cámara, ya que el centro del círculo determinado por la región iluminada no coincide con el centro de la imagen. En consecuencia, en estas condiciones la suposición de que el punto principal se encuentra en el centro de la imagen no es válida.

En cuanto a los resultados obtenidos tanto con el microscopio del Hospital de Tacuarembó como con las cámaras del laboratorio, es importante analizar qué factores influyen sobre el error RMS en la reproyección de puntos. Lo que se busca en principio es evaluar *únicamente* el método de calibración que se está usando, esto es, qué tan bueno es el ajuste de los parámetros y cuál es el error que se comete al modelar la óptica del microscopio como una cámara pinhole sin tener en cuenta la distorsión y fijando el punto principal en el centro de la imagen. Sin embargo, existen

otros factores que influyen sobre los resultados:

- Error de detección del localizador. El algoritmo de evaluación implementado requiere conocer la posición espacial de los puntos de control (esquinas del patrón) y un error en la ubicación de estos puntos influye sobre el resultado final. Dado que la posición de los puntos de control se conoce a través de la ubicación de los marcadores adheridos al patrón, el error cometido por el localizador al detectar los marcadores tendrá efecto sobre el error cometido en la reproyección. Puede encontrarse información sobre la precisión del localizador en la Subsección 3.2.1.
- Movimiento del patrón. Tal como se explicó al comienzo de esta sección, el método de evaluación compara la posición de las esquinas detectadas sobre el damero con la posición que se obtiene luego de proyectar las coordenadas de su ubicación espacial a través de las matrices resultantes de la calibración. Es necesario entonces sincronizar la detección de los marcadores del patrón con la adquisición de la imagen sobre la que luego se detectarán las esquinas, para que efectivamente se trate de puntos correspondientes. Sin embargo, entre los instantes en que se procesan una y otra tarea puede producirse un movimiento del patrón, lo cual tendrá influencia sobre la medida del error RMS. En la práctica se vio que esta influencia no es menor, especialmente dado que no se construyó un dispositivo que permitiera fijar el patrón en diferentes posiciones frente a la cámara, sino que el mismo era sostenido con la mano o en el mejor de los casos apoyado sobre la camilla dentro de la sala, lo cual tampoco asegura que el patrón permanezca inmóvil.
- Movimiento relativo entre el patrón o el microscopio y sus marcadores. El método de evaluación de calibración, y todos los algoritmos en general, asumen que tanto el patrón como el microscopio están rígidamente adheridos a sus marcadores asociados y esto es a grandes rasgos verdadero. Sin embargo, dada la construcción de los dispositivos, se pudo constatar en la práctica que luego de algunos minutos de uso pueden darse movimientos de los marcadores, lo cual claramente afecta el error de reproyección, ya sea porque la matriz H calculada deja de ser válida en el caso de que se muevan los marcadores del microscopio, o porque se está ubicando mal las esquinas en el espacio en caso de que se muevan los marcadores del patrón.

La combinación de estos factores se ve reflejada en los valores del error RMS presentados en las Tablas 5.11 a 5.14 correspondientes a la calibración de la cámara del laboratorio, y en las Tablas 5.16 a 5.19 para la calibración del microscopio. Si bien en la mayoría de los casos el error superó el pixel, en algunos se alcanzó una precisión subpixélica, lo que muestra que al correr los algoritmos en condiciones óptimas éstos brindan buenos resultados. Por otro lado, las imágenes de las Figuras 5.7 a 5.10 para el caso del microscopio, dan una medida cualitativa del desempeño de

los algoritmos y si se tiene en cuenta la precisión requerida por la aplicación, los resultados obtenidos pueden considerarse aceptables para un número mayor de casos.

Todos los resultados presentados relativos al microscopio corresponden a calibraciones de su óptica realizadas con el patrón tipo damero plano dado que las pruebas hechas con el patrón cúbico no dieron buenos resultados durante el trabajo en la sala de operaciones. En una gran cantidad de casos no se consiguió una buena detección de las esquinas, pero aún cuando éstas se detectaron correctamente, tanto la matriz de parámetros intrínsecos A como la transformación H entre el sistema de coordenadas del mundo y de la cámara adquirieron valores muy alejados de lo esperado; aún no se ha podido determinar el motivo por el cual esto sucede.

Como comentario final, cabe mencionar que en la práctica con el microscopio se comprobó que los mejores resultados se tienen cuando se trabaja con aumentos bajos. Para aumentos mayores a 3 o 4 no se logró una calibración aceptable. Si bien la bibliografía consultada sobre aplicaciones con microscopios adopta el modelo de cámara pinhole más coeficientes de distorsión, se plantea la duda de si este modelo sigue siendo adecuado al incrementar el aumento del microscopio o si sería conveniente estudiar otro que ajuste mejor a el comportamiento físico del sistema óptico.

Capítulo 6

Realidad Aumentada

6.1. Introducción

Realidad Aumentada (RA) es el proceso mediante el cual se añade información virtual a la real. Una foto con una anotación de texto, aunque simple, es un ejemplo. Procesos más sofisticados adaptan los contenidos en tiempo real de acuerdo a la geometría del entorno. Ésta es la principal diferencia con la realidad virtual, puesto que no sustituye la realidad física, sino que sob reimprime los datos informáticos al mundo real.

Si bien la RA es un fenómeno relativamente nuevo ya ha encontrado su lugar en diferentes áreas de aplicación, debido fundamentalmente a que permite al usuario interactuar con su alrededor, en un lenguaje digital.

A lo largo de este proyecto de fin de carrera, si bien se aplicó RA en el caso particular de un microscopio quirúrgico, se abordaron temas que son completamente generales y comunes a cualquier aplicación de RA. En este capítulo se presentan los aspectos más generales y se destina el próximo capítulo para mostrar la aplicación de RA en el caso particular de un microscopio quirúrgico.

6.1.1. Definición

Hay dos definiciones comúnmente aceptadas de Realidad Aumentada en la actualidad [67]. Una de ellas fue propuesta por Ronald Azuma en 1997. La definición de Azuma dice que la realidad aumentada:

- Combina elementos reales y virtuales.
- Es interactiva en tiempo real.
- Está registrada en 3D.

Además Paul Milgram y Fumio Kishino definieron la realidad de Milgram-Virtuality Continuum en 1994. La describen como un continuo que abarca desde el entorno real a un entorno virtual puro. Entre medio hay Realidad Aumentada (más cerca del entorno real) y Virtualidad Aumentada (está más cerca del entorno virtual).



Figura 6.1: Realidad de Milgram-Virtuality Continuum

6.1.2. Aplicaciones

Con el desarrollo de las nuevas tecnologías audiovisuales y de las tarjetas gráficas, la Realidad Aumentada ha emergido como una interesante posibilidad en diferentes áreas de aplicación. En arquitectura se ha utilizado para reestructurar virtualmente edificios. También ha sido utilizada en museos y exhibiciones para mostrar información sobre objetos o más habitualmente en simulaciones de vuelo, vídeo juegos, publicidad y diseño web.

Para conocer la ubicación del usuario se pueden utilizar diferentes técnicas. Una de las técnicas más comunes, debido a su bajo precio, es utilizar una cámara y algún patrón conocido. Se puede así determinar la ubicación relativa entre el usuario y la cámara. Existen otras posibilidades, no tan económicas, como pueden ser GPS, acelerómetros o sensores ópticos.

6.1.3. Software libre para RA

ARToolKit Librería GNU GPL que permite la creación de aplicaciones de realidad aumentada, desarrollada originalmente por Hirokazu Kato en 1999. Fue publicada por el HIT Lab de la Universidad de Washington. Actualmente se mantiene como un proyecto de código abierto alojado en SourceForge con licencias comerciales disponibles en ARToolWorks.

ATOMIC Authoring Tool Es un software Multi-plataforma para la creación de aplicaciones de realidad aumentada, el cual es un Front-end para la librería ARToolKit. Fue Desarrollado para no-programadores, y permite crear rápidamente, pequeñas y sencillas aplicaciones de Realidad Aumentada. Está licenciado bajo la Licencia GNU GPL.

La Figura 6.2 muestra un cubo colocado virtualmente sobre un patrón de la librería Artoolkit. Al detectar el patrón de dimensiones conocidas el software calcula la ubicación del mismo respecto a la cámara y dibuja en dicho lugar un objeto, en este caso un cubo.



Figura 6.2: Realidad Aumentada con Artoolkit

6.2. Creación del Mundo Virtual

Para generar computacionalmente una escena 3D que se confunda con la realidad es necesario transitar por las siguientes etapas:

1. Crear el mundo virtual.
2. Determinar qué parte del mundo se va a mostrar en pantalla.
3. Determinar cómo se verá cada pixel en la imagen.

6.2.1. Creación del mundo 3D

La creación de un mundo 3D es el agregado de los objetos que podrían formar parte de una escena. En principio, estos objetos son sintéticos y no guardan ninguna relación entre sí, simplemente forman parte del mundo.

Los objetos más simples son puntos, líneas y polígonos. A partir de ellos es posible generar elementos más complejos, como pueden ser estructuras geométricas o mallas. Se podría pensar en una escena donde aparezca una botella, por ejemplo. Se hace necesario generar un conjunto de puntos con la forma de una botella. Cuantos más puntos se utilicen para la construcción del objeto botella seguramente más fidedigna será su representación geométrica. Pero, ¿cuáles son las características que debe cumplir ese conjunto de puntos para generar la sensación de que es una botella?. Se necesita más que un conjunto de puntos distribuidos de una cierta manera para representar virtualmente un objeto de la realidad. Es necesario considerar cuestiones como la textura, la luminosidad, flexibilidad, transparencia, posible interacción con los demás objetos, etc. Estas características son asignadas a los distintos objetos por el programador y se ven en las próximas secciones.

6.2.2. Escena OpenGL

OpenGL (Open Graphics Library) es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D. La interfaz consiste en más de 250 funciones diferentes que pueden usarse para dibujar escenas tridimensionales complejas a partir de primitivas geométricas simples, tales como puntos, líneas y triángulos. Fue desarrollada originalmente por Silicon Graphics Inc. (SGI) en 1992 [72].

En un determinado momento, la pantalla muestra sólo una pequeña parte del mundo virtual que se ha creado. Lo que se muestra en pantalla es determinado por un conjunto de operaciones matemáticas que contemplan aspectos como la óptica de la cámara, la posición de los objetos y el tamaño de imagen. A continuación se describe el manejo que realiza de ellas la especificación OpenGL con el fin de determinar quién sale en la foto.

Se puede establecer una analogía entre cómo se forma la escena que se ve en pantalla y el tomar una foto con una cámara. Los pasos que se siguen se muestran en la Figura 6.3 y son los siguientes:

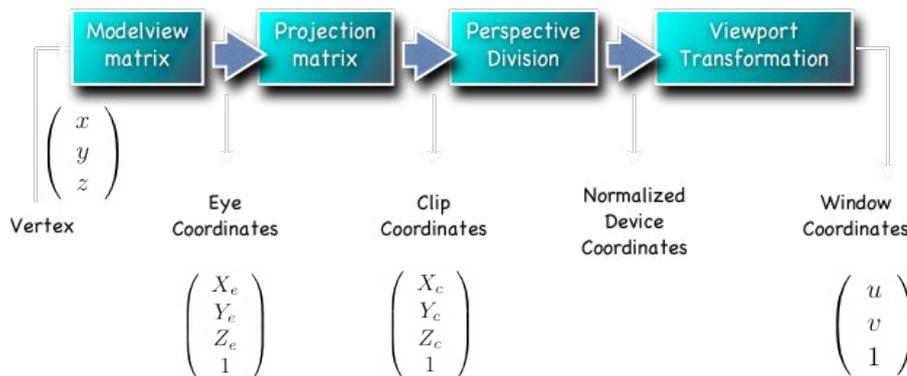


Figura 6.3: Cadena de transformaciones que se aplican para generar la escena. Imagen extraída de [48]

1. Se coloca la cámara y se la posiciona mirando en una dirección determinada (*Viewing transformation*).
2. Se colocan los objetos a ser fotografiados (lo que aparecerá en pantalla). En este paso es que se construye el mundo virtual, se posicionan todos los elementos que pueden llegar a aparecer “inyectados” al aumentar la realidad (*Modeling transformation*).
3. Se elige un lente para la cámara. Aquí es que se relacionan los parámetros intrínsecos de la cámara real con los de la cámara virtual. Para lograr una correlación entre lo visto en el mundo real y el mundo virtual es imprescindible que ambas cámaras proyecten los puntos del espacio de la misma manera (*Projection transformation*).

4. Una vez que se ha formado la imagen hay que determinar en qué sector de la pantalla se va a mostrar, qué espacio va a ocupar (*Viewport transformation*).

Estos pasos son llevados a cabo por la computadora mediante multiplicaciones de matrices.

Dado un punto en el espacio virtual, éste es colocado en el mundo virtual referido a un sistema de coordenadas y en una cierta posición respecto a la cámara virtual. Esta operación se codifica dentro de la Modelview matrix.

Luego se determina la forma en que este punto, más generalmente un objeto, se proyecta en la imagen. Existen dos tipos de proyección: perspectiva y ortogonal. En nuestro caso se utilizó la proyección perspectiva ya que se busca que los objetos se vean en pantalla de la misma forma en que se verían si fuesen virtuales.

Esta matriz de proyección determina un *frustum* (ver Figura 6.4), una sección de una pirámide cuyo vértice se encuentra en el centro de la cámara. Aquellos objetos que quedan comprendidos dentro del frustum serán visibles en la imagen, los que queden por fuera no. Los parámetros que definen el frustum son los siguientes:

- *fovy*: es el ángulo de vista en la dirección *y*.
- *aspect*: es la relación entre *w* y *h* en la imagen.
- *near*: mínimo valor de coordenada en *z* que se proyecta.
- *far*: máximo valor de coordenada en *z* que se proyecta.

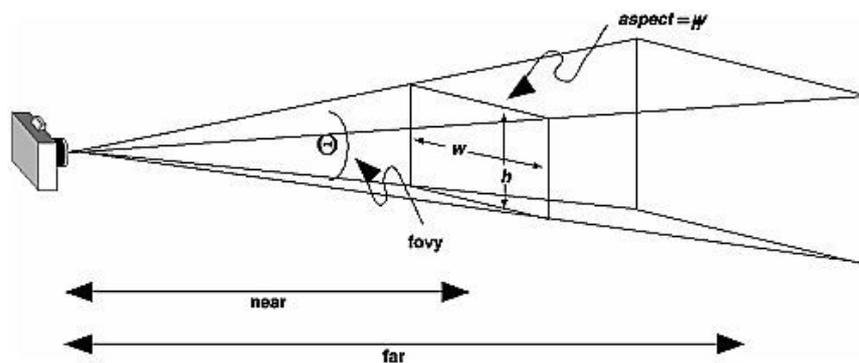


Figura 6.4: Parámetros que definen el área visible y la perspectiva utilizando OpenGL (frustum). Imagen extraída de [4].

Con el ángulo de vista y la relación de aspecto se define una pirámide de proyección. Esta pirámide es luego truncada, a partir de lo que se conoce como *clipping range* (rango de corte). Todos los puntos cuya coordenada *z* no se encuentre en el rango determinado por los parámetros *near* y *far* no son proyectados en la escena.

Para que los objetos virtuales se vean (geométricamente) como si fuesen reales es necesario construir la matriz de proyección a partir de los parámetros intrínsecos de la cámara obtenidos mediante la calibración. Este proceso se muestra en la Sección 6.3. Luego se realiza internamente la división en perspectiva y finalmente se determina la región de la pantalla en la que se desea dibujar la escena y el tamaño de la imagen.

6.2.3. Haciendo que parezca real

Esta sección se enfocará en ver cómo hacer para que los objetos ingresados en la escena virtual parezcan reales. Algunos de los aspectos que deben ser tomados en cuenta son:

Formas Al observar una escena real se pueden distinguir una gran variedad de formas: cuadrados, rectángulos, paralelogramos, círculos, triángulos, etc. Sin embargo, si se quieren representar de manera más real, con sus imperfecciones o si se quieren representar objetos más complejos es necesario combinar varias en un mismo objeto. La forma típica de generación de objetos virtuales es a partir de triángulos. La cantidad de triángulos necesarios para crear un objeto virtual depende de las particularidades geométricas del objeto y del nivel de detalle con que se lo quiera representar. Existe un compromiso entre la cantidad de formas a utilizar para la representación del objeto y el costo computacional asociado a su representación.

Textura Para extraer información sobre una superficie real se suele recurrir a dos de nuestros sentidos: la vista y tacto. Así por ejemplo, se la puede observar desde diferentes ángulos o tocarla para saber si es dura o blanda. En el caso de las superficies virtuales toda la información está contenida en lo que podemos ver. Esta información se puede disgregar en tres propiedades: color, textura y reflectancia.

Iluminación Éste es un de los aspectos más difíciles de manejar. Existen diversos factores para ello, a saber:

- La localización e intensidad de la fuente de luz puede variar con el tiempo.
- El camino seguido por los rayos es sensible a la posición de objetos no necesariamente pertenecientes al mundo virtual.
- En general existen múltiples fuentes de luz.

Al iluminar un objeto de manera no uniforme se aprecian variaciones en el brillo de su superficie. Estas variaciones son las que hacen que percibamos, por ejemplo, que una pelota es redonda. Estas variaciones en la intensidad de luz refuerzan la ilusión de que el objeto tiene profundidad. Otro efecto que se produce debido a la iluminación es el de las sombras. Un adecuado manejo de las mismas contribuye a que la escena parezca más real.

Z-Buffer El Z-Buffer asigna a cada objeto que pertenece a la escena un número que representa qué tan lejos se encuentra dicho objeto de la cámara. El nombre proviene del hecho de que se suele llamar z al eje perpendicular al plano de la imagen. Dados dos objetos pertenecientes a la escena virtual y que se proyectan al mismo punto en la imagen, es necesario determinar cuál de ellos se va a desplegar en pantalla. Para tomar dicha decisión se consulta al Z-Buffer: el objeto cuya coordenada z sea menor será el que se muestre en pantalla.

6.3. Alineación del mundo virtual con el real

Librerías como Artoolkit han facilitado el desarrollo de aplicaciones de Realidad Aumentada. No obstante, si se quiere profundizar en el desarrollo de una aplicación de Realidad Aumentada, un primer paso es desarrollar un Artoolkit propio, es decir agregar Realidad Aumentada sin necesidad de recurrir a Artoolkit.

Si se ha calibrado la cámara, esto es, si se conocen exactamente los parámetros intrínsecos y extrínsecos de la misma, es de suponer que no debería ser difícil configurar la cámara virtual para que proyecte los objetos espaciales de igual forma que la real, y de hecho no lo es. Sin embargo, existen algunos detalles a considerar en dicho proceso. Debido a que las librería gráficas (en particular OpenGL) no usan los mismos parámetros para armar la escena virtual, no se puede configurar dicha escena directamente a partir de los parámetros intrínsecos y extrínsecos. A continuación se explica la relación entre las cámaras reales y virtuales.

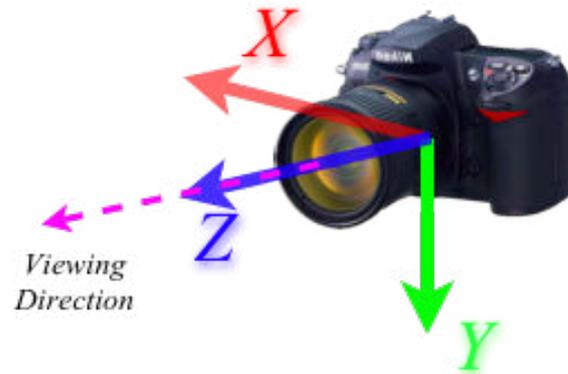
6.3.1. Modelos de la cámaras

En la Figura 6.5 se muestran los sistemas de coordenadas asociados a cada cámara. En particular se observa que la cámara mira hacia los z positivos y que el sistema de coordenadas asociado a ella es directo. A diferencia del modelo de cámara real, la cámara virtual mira hacia los z negativos. El sistema de coordenadas que se ve en la Figura 6.5(b) no es directo y el vector y apunta hacia arriba.

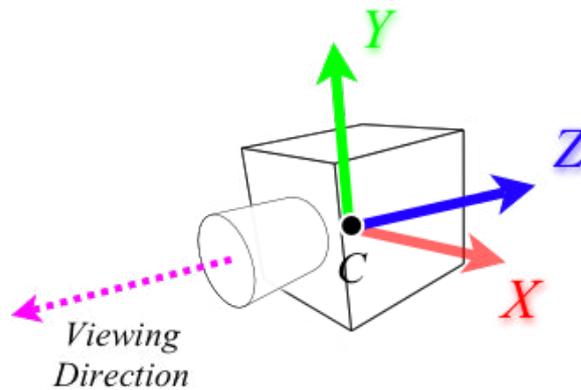
6.3.2. Correspondencia entre ambos modelos

Para establecer la correspondencia entre ambos modelos se deben seguir los siguientes pasos:

Mover la cámara La cámara virtual se pone en la misma posición que la cámara real, mirando hacia los z positivos y con la dirección y puntando hacia abajo. Tomando en cuenta que la matriz ModelView refiere los puntos respecto al sistema de referencia solidario a la cámara queda claro que, una vez que coinciden los ejes de ambos modelos de cámara, la matriz ModelView corresponde a la matriz de parámetros extrínsecos con la única diferencia de que es una matriz 4×4 cuya última fila es el vector $(0 \ 0 \ 0 \ 1)$.



(a) Modelo de la cámara real



(b) Modelo de la cámara OpenGL

Figura 6.5: Modelos de las cámaras. Imágenes extraídas de [48]

Determinar los parámetros del frustum Si se supone que el punto principal se encuentra en el centro de la imagen y se compara el modelo pinhole de la cámara con el frustum se deduce inmediatamente que:

$$\tan\left(\frac{fovy}{2}\right) = \frac{h}{2f_y} \quad (6.1)$$

Si se recuerda del modelo pinhole, f es la distancia focal en mm, f_x representa la distancia f en número de píxeles según la dirección x y f_y representa la distancia f en número de píxeles en la dirección y .

Se realiza el siguiente experimento: dada la proyección en la imagen de un plano paralelo a la cámara, sea d_x la cantidad de milímetros que representa un pixel en la dirección x y d_y la cantidad de milímetros que representa el mismo pixel en la dirección y . Se supondrá que para cierta resolución, por ejemplo 320×240 , $f_x = f_y$. Si ahora se cambia la resolución a 352×288 , la cámara va a conservar el mismo valor de f y va a ver el mismo espacio físico ya que el lente no se modificó. Sin embargo la cantidad de milímetros por pixel que se van a ver en el plano proyectado va a cambiar

debido al cambio de resolución.

$$f = d_x f_x \quad (6.2)$$

$$f = d_y f_y \quad (6.3)$$

Se define entonces la relación de aspecto de pixel como: $d = \frac{d_x}{d_y} = \frac{f_y}{f_x}$. En el modelo de proyección OpenGL se supone que $f_x = f_y$ y por eso se toma la relación de imagen como relación de aspecto espacial. En caso de que esto no sea así, se puede establecer el factor correctivo en la relación de aspecto, es decir, en vez de escribir $aspect = \frac{w}{h}$ se escribe la relación de aspecto espacial, es decir:

$$aspect = \frac{d_x w}{d_y h} = \frac{f_y w}{f_x h} \quad (6.4)$$

Corrección del punto principal Si el punto principal no se encuentra en el centro de la imagen se puede utilizar la matriz Viewport para corregirlo.

Si se consulta la especificación OpenGL se encuentra la función `glViewport(x0, y0, width, height)`.

$$x_w = (x_{nd} + 1) \frac{width}{2} + x_0 \quad (6.5)$$

$$y_w = (y_{nd} + 1) \frac{height}{2} + y_0 \quad (6.6)$$

A x_{nd} e y_{nd} se les llama coordenadas normalizadas y toman valores en el rango $[-1,1]$.

A partir de las ecuaciones anteriores y de la Figura 6.6 se deduce que una desviación del punto principal del centro de la imagen se corrige utilizando las Ecuaciones 6.7 y 6.8:

$$x_0 = p_x - \frac{width}{2} \quad (6.7)$$

$$y_0 = \frac{height}{2} - p_y \quad (6.8)$$

Nuestro Artoolkit La Figura 6.7 muestra un ejemplo de Realidad Aumentada utilizando la librería OpenCV para calcular la posición de un damero en el espacio y OpenGL para dibujar el objeto virtual. No se usa el patrón Artoolkit.

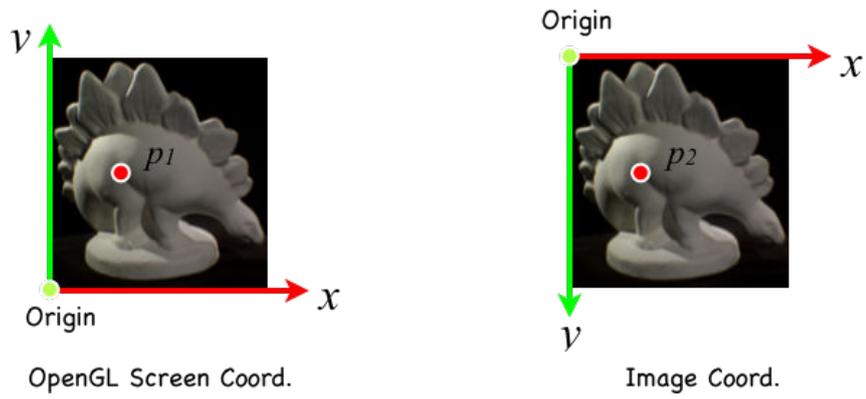


Figura 6.6: Ubicación de los ejes en la imagen

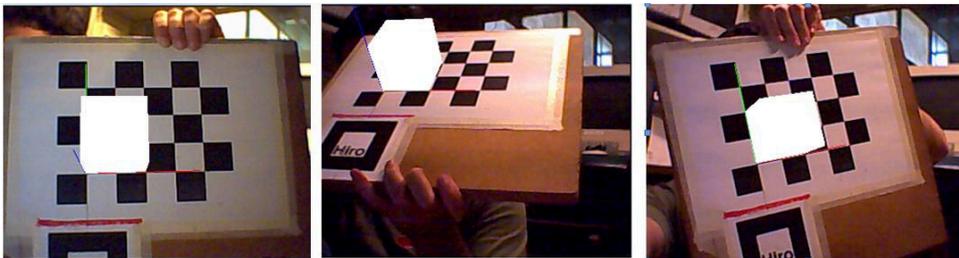


Figura 6.7: Realidad Aumentada utilizando OpenCV y OpenGL

Capítulo 7

Realidad Aumentada en el Microscopio

7.1. Introducción

En el Capítulo 6 se describió cómo crear una escena virtual y cómo colocar en dicha escena modelos gráficos de forma que éstos se perciban como si fuesen reales. Este procedimiento, en el que se mezclan objetos virtuales con objetos de la realidad, se llama Realidad Aumentada (RA). Se mostró el método que se utiliza para alinear los mundos reales y virtuales y se vio cómo lograr la alineación a partir de los parámetros de calibración vistos en el Capítulo 4. En el presente capítulo se mostrará la aplicación del procedimiento de calibración detallado en el Capítulo 5 para dotar de RA a la señal de video extraída de un microscopio quirúrgico.

En el caso de RA aplicado a la medicina es útil considerar a las imágenes médicas como mapas o modelos 3D de la anatomía del paciente. En vez de desplegar esos mapas o modelos en un ambiente distinto al que está observando el neurocirujano se lo superpone a la escena que él está viendo en un determinado momento. Si se tiene, por ejemplo, el modelo de un tumor se puede superponer ese modelo gráfico en la escena en el lugar que debería verse el tumor real, si éste fuera visible. De esta manera se dota al neurocirujano de una especie de “visión en rayos X”.

Un requerimiento fundamental para lograr RA en aplicaciones médicas es un registrado preciso entre la escena virtual y la escena real. Si la alineación entre ambas escenas no es la adecuada, el sistema no funcionará. Otro aspecto importante a considerar es que la información debe ser desplegada en tiempo real, por lo que los tiempos de procesamiento deben, necesariamente, ser tomados en consideración.

7.2. Estado del arte

Existen diversas publicaciones académicas y libros que describen sistemas de RA aplicados a la cirugía guiada por imágenes [49, 50, 51, 75, 58, 15, 21]. En cuanto a dotar de realidad aumentada al microscopio quirúrgico [49, 15, 21] la principal

referencia es la publicación *Design and Evaluation of a System for Microscope-Assisted Guided Interventions (MAGI)*, de Edwards [15]. Allí se presenta el desarrollo de un sistema de realidad aumentada que superpone la información agregada virtualmente directamente en cada uno de los binoculares del microscopio. Esto supera los alcances de este proyecto, el cual se limita a superponer la información virtual a la salida de video compuesto del microscopio. No obstante, el procedimiento que se siguió durante el proyecto es similar al realizado por MAGI con la salvedad antes mencionada.

Dentro de las firmas comerciales, la que se encuentra a la avanzada en cuanto a productos de cirugía guiada por imágenes es la firma BrainLab [68]. Además de contar con los más completos y modernos neuronavegadores, la empresa ofrece módulos de los mismos, los cuales son vendidos como complemento del equipo. Entre los módulos que ofrece, se encuentra el de integrar el microscopio quirúrgico al sistema de neuronavegación.

7.3. Descripción del sistema

El objetivo del sistema de realidad aumentada es superponer modelos tridimensionales de estructuras anatómicas en la señal de video extraída del microscopio.

Para lograr esto es necesario conocer la ubicación espacial de los distintos elementos que forman parte de la escena y para ello se utilizó el localizador Polaris Spectra. Este dispositivo es capaz de determinar con precisión la ubicación de ciertos marcadores reflectivos que, si se adhieren rígidamente a objetos que forman parte de la sala de operaciones, permiten conocer la ubicación de éstos, a menos de un movimiento rígido (transformación que ubica el objeto respecto a los marcadores). En particular se usó el localizador Polaris para conocer la ubicación del microscopio, del paciente y de una herramienta tipo puntero que se utiliza como señalador.

La ubicación de los elementos que pertenecen a la escena es referida al sistema de coordenadas del localizador Polaris, el cual se supone permanece estático durante la intervención. Es imprescindible que todos los elementos que pertenecen a la escena puedan referirse a un cierto marco de referencia fijo, aunque éste no necesariamente debe ser el localizador. Si se desea mover el localizador se pueden adherir marcadores a algún objeto de la sala que permanezca fijo durante la intervención y utilizarlo como referencia.

Para conocer la ubicación espacial del microscopio se adhirieron los marcadores que se muestran en la Figura 5.1. Se llamará M a la matriz de transformación que ubica los marcadores adheridos al microscopio respecto al sistema de referencia del localizador (ver Figura 7.1).

Para superponer un objeto de la escena real en el video es necesario saber cómo proyecta el microscopio los puntos espaciales. En el Capítulo 4 se vio que esta información se encuentra contenida en la matriz de parámetros intrínsecos A .

Sean $\mathbf{X}_c = (X_c, Y_c, Z_c)^T$ las coordenadas de un punto P respecto al sistema

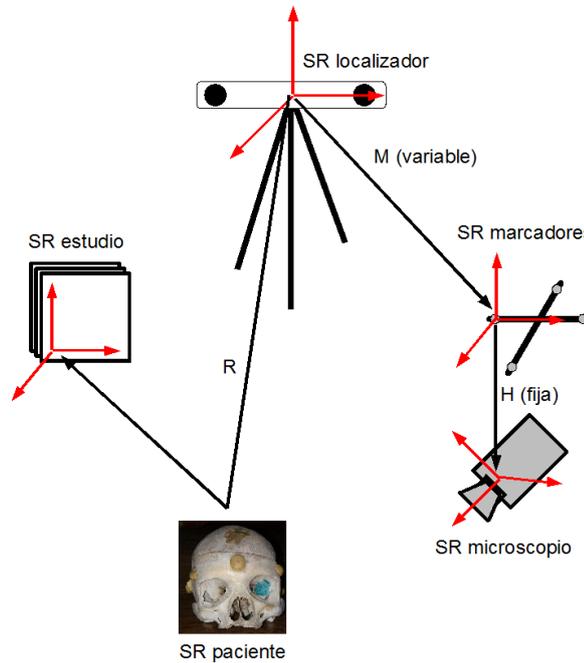


Figura 7.1: Esquema del sistema de realidad aumentada en el microscopio quirúrgico

de referencia de la cámara (coordenadas normalizadas), entonces por lo visto en el Capítulo 4 se tiene que:

$$\mathbf{x} = A\mathbf{X}_c$$

donde \mathbf{x} es la proyección del punto P en la imagen. Esta ecuación establece que si se conocen las coordenadas de un punto en el sistema de referencia de la cámara y la matriz de parámetros intrínsecos A entonces es inmediato saber dónde se proyecta dicho punto en la imagen.

La matriz M por su parte, lleva puntos del sistema de referencia (de aquí en más SR) de los marcadores adheridos al microscopio al SR del localizador Polaris (pensar en el origen). Por lo tanto, si se premultiplican las coordenadas \mathbf{X}_l referidas al localizador Polaris por la matriz M^{-1} se realiza la transformación inversa

$$\mathbf{X}_m = M^{-1}\mathbf{X}_l$$

donde X_m son las coordenadas del punto P respecto a los marcadores adheridos al microscopio.

Durante el proceso de calibración que se describió en el Capítulo 5 se obtuvo, además de la matriz de parámetros intrínsecos A , la matriz H que ubica la posición del centro óptico de la cámara del microscopio respecto a los marcadores adheridos al mismo. Teniendo en cuenta que H lleva puntos del SR de la cámara al SR de los marcadores adheridos al microscopio se tiene que:

$$\mathbf{X}_c = H^{-1}\mathbf{X}_m$$

Si se combinan las ecuaciones se tiene que

$$\mathbf{x}(t) = AH^{-1}M(t)^{-1}\mathbf{X}_l(t)$$

donde se ha destacado el hecho de que las matrices A y H no cambian durante el transcurso de la operación (para valores de aumento y distancia de trabajo fijos). En cambio sí puede modificarse la ubicación de los puntos que se quiere proyectar en la señal de video. El puntero por ejemplo es movido por el cirujano. También el microscopio se mueve durante la intervención.

Superposición de información preoperatoria

Hasta el momento se ha encontrado el conjunto de transformaciones que vinculan las coordenadas espaciales brindadas por el localizador con la imagen obtenida en el microscopio. Lo que falta es relacionar el estudio preoperatorio y la información extraída de éste con el paciente.

El estudio se encuentra referido a un sistema de coordenadas que no coincide con el sistema de referencia del paciente en el momento de la intervención. Si bien el estudio es realizado antes de la operación, la rigidez del cráneo permite suponer que la posición relativa de las distintas estructuras que aparecen en el estudio se mantiene, es decir, se puede alinear la información obtenida a partir de los estudios antes de la operación con el paciente mediante una simple transformación rígida R . A este mapeo entre el estudio del paciente y la ubicación del mismo en la sala de operaciones se le llama *registrado*.

La matriz R se estima a partir de tres o más correspondencias entre puntos del estudio y puntos del paciente. Si se llama \mathbf{X}_e a un punto referido al SR del estudio entonces éste se vincula con la posición en el SR del localizador mediante la transformación de registrado R de la siguiente manera

$$\mathbf{X}_l = R\mathbf{X}_e$$

Los modelos de estructuras anatómicas que se proyectan en la señal de video no son otra cosa que un conjunto de puntos 3D referidos al sistema del estudio, por lo que para proyectar los mismos alcanza con aplicar el conjunto de transformaciones que se detallaron anteriormente

$$\mathbf{x}(t) = AH^{-1}M(t)^{-1}R\mathbf{X}_e$$

El cálculo de estas transformaciones implica la realización de una serie de procedimientos. Tareas como el seguimiento del paciente, del microscopio y de otros instrumentos quirúrgicos son llevadas a cabo durante la operación, mientras que otras se realizan antes de la misma.

7.4. Tareas fuera de sala

Dentro de las tareas a realizar antes de la operación se encuentran: colocación de fiduciaros en el paciente, realización del estudio, planificación de la operación, calibración de instrumentos quirúrgicos.

A continuación se detallan los procedimientos llevados a cabo durante este proyecto.

7.4.1. Colocación de fiduciaros

Se colocan fiduciaros en el paciente. En este proyecto se trabajó con un fantoma, el cual ya contaba con marcadores adheridos.



Figura 7.2: Cráneo utilizado para realizar las pruebas

7.4.2. Realización del estudio

Se cuenta con un estudio de TC realizado al fantoma.

7.4.3. Segmentación del estudio

La segmentación de las estructuras de interés del estudio se realizaron utilizando el visualizador de estudios médicos 3D Slicer [45]. El 3D Slicer permite abrir archivos de imágenes de TC (entre otras modalidades) y mostrarlas en el plano axial, sagital y coronal. Además se pueden generar modelos de superficies tridimensionales a partir de la segmentación de los cortes bidimensionales.

Existen varios métodos para segmentar las estructuras, pudiendo clasificarse en primera instancia en semiautomáticos y manuales. En los primeros se construye el modelo 3D a partir de una semilla ingresada por el usuario o a partir de una franja de niveles de grises también determinada por el usuario.

Para segmentar manualmente una estructura se selecciona una de las vistas (axial, sagital o coronal) y se marca, para cada uno de los cortes en que se observa la estructura, el contorno de la misma (Figura 7.3(b)), luego el programa determina la

región interior del contorno como formando parte de la estructura. En este proyecto se utilizó el método manual ya que suele ser más preciso. A partir de todas las regiones encontradas se construye un modelo de superficie de la estructura segmentada, consistente en una malla de elementos triangulares.

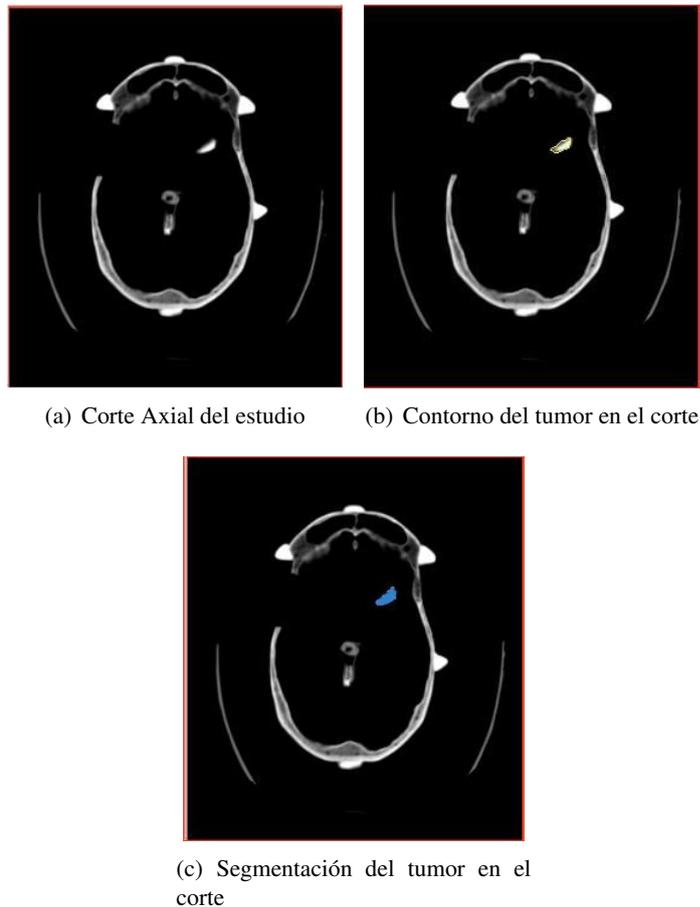


Figura 7.3: Creación de modelos de superficie en 3D Slicer [45]

7.4.4. Calibración del puntero

El localizador brinda la posición de los marcadores adheridos al puntero pero lo que en realidad interesa conocer es el punto sobre el cual se apoya el mismo. Esto es necesario para determinar, por ejemplo, la ubicación de ciertos puntos característicos del paciente (fiduciaros) que se utilizan para conocer su ubicación en la sala de operaciones.

Se considera que el puntero está calibrado cuando se conoce la posición de la punta respecto a sus marcadores. Notar que esto no implica conocer la orientación de la herramienta respecto a los marcadores, simplemente la posición de la punta.

El procedimiento de calibración del puntero consiste en extraer posiciones de sus marcadores mientras se pivotea al mismo sobre un punto fijo (ver Figura 7.4). La posición de la punta se calcula como el punto más invariante del rígido marcadores-puntero en el conjunto de movimientos. Para realizar la calibración se utilizó la implementación que viene con el software del localizador Polaris. También existe una implementación en la librería IGSTK. A continuación se muestran los detalles matemáticos.

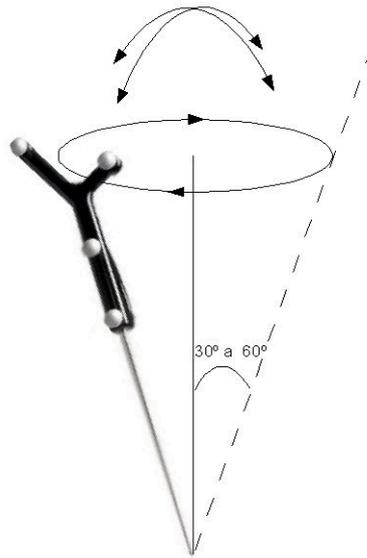


Figura 7.4: Calibración del puntero mediante pivoteo

Se quiere encontrar la ubicación de la punta $\mathbf{O} = (O_x, O_y, O_z)^T$ respecto a los marcadores. Sea P la transformación que ubica los marcadores en el espacio. El punto (x, y, z) que está siendo tocado por la punta en un determinado momento se puede escribir de la siguiente manera:

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = P \begin{pmatrix} O_x \\ O_y \\ O_z \\ 1 \end{pmatrix} \quad (7.1)$$

$$P = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} & T_x \\ r_{yx} & r_{yy} & r_{yz} & T_y \\ r_{zx} & r_{zy} & r_{zz} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7.2)$$

donde $r_{xx}, r_{xy}, \dots, r_{zz}, T_x, T_y$ y T_z son las rotaciones y traslaciones respectivamente que componen a la transformación rígida P . Cabe recordar que P va a ir cambiando

con la posición del puntero, es decir que la relación anterior se cumplirá para cada una de las posiciones del puntero. Lo que no va a cambiar es el punto de apoyo de la herramienta, es decir, (x, y, z) . Otra forma de escribir lo anterior es:

$$\begin{aligned} r_{xx}O_x + r_{xy}O_y + r_{xz}O_z + (-1) \cdot x + 0 \cdot y + 0 \cdot z &= -T_x \\ r_{yx}O_x + r_{yy}O_y + r_{yz}O_z + 0 \cdot x + (-1) \cdot y + 0 \cdot z &= -T_y \\ r_{zx}O_x + r_{zy}O_y + r_{zz}O_z + 0 \cdot x + 0 \cdot y + (-1) \cdot z &= -T_z \end{aligned} \quad (7.3)$$

Esto se puede reescribir de la siguiente manera:

$$\left[\begin{array}{c|ccc} P & -1 & 0 & 0 \\ & 0 & -1 & 0 \\ & 0 & 0 & -1 \end{array} \right] \begin{pmatrix} O_x \\ O_y \\ O_z \\ x \\ y \\ z \end{pmatrix} = - \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} \quad (7.4)$$

Si ahora se construyen un par de matrices M y N que sean de la forma anterior pero que tengan la información de todos los puntos, queda el siguiente sistema para resolver:

$$M \begin{pmatrix} O_x \\ O_y \\ O_z \\ x \\ y \\ z \end{pmatrix} = N \quad (7.5)$$

Este sistema puede resolverse por mínimos cuadrados (consultar Apéndice 5 de [29]), lo que conduce a la solución:

$$\begin{pmatrix} O_x \\ O_y \\ O_z \\ T_x \\ T_y \\ T_z \end{pmatrix} = (M^t M)^{-1} M^t N \quad (7.6)$$

Una vez calibrado el puntero se lo utiliza para determinar la posición de puntos en el espacio respecto al localizador Polaris.

7.4.5. Calibración del patrón

Para calibrar el microscopio es necesario establecer una correspondencia entre puntos espaciales y puntos en la imagen. Los puntos espaciales se podrían determinar

marcando con el puntero cada una de las esquinas del patrón, pero este método es poco práctico. Es posible calibrar el patrón de forma que la posición de todas sus esquinas se pueda determinar con sólo detectar la posición de los marcadores adheridos a él. Los detalles de la implementación de este procedimiento se muestran en el Apéndice B.

7.4.6. Calibración del microscopio

El proceso de calibración del microscopio se detalló en el Capítulo 5. Este procedimiento determina la matriz de parámetros intrínsecos A y la transformación que ubica el centro de la cámara respecto a los marcadores adheridos al microscopio H .

La Figura 7.5 muestra la calibración del microscopio utilizando el patrón damero.



Figura 7.5: Calibración del microscopio utilizando un damero

7.5. Tareas dentro de sala

Estas son las tareas que se realizan dentro de la sala de operaciones.

7.5.1. Registrado del Paciente

Se sigue el procedimiento que se describe a continuación:

- Se coloca el paciente en la sala de operaciones de forma tal que permanezca inmóvil a lo largo de la intervención.

- Se determina la posición de los fiduciaros en el paciente tocándolos con el puntero calibrado. Previamente se había indicado manualmente su ubicación sobre el estudio utilizando el visualizador 3D Slicer [45].
- A partir de la correspondencia entre los puntos referidos al localizador y los referidos al estudio se calcula la transformación de registrado (ver Figura 7.1). En la librería IGSTK existe una clase que implementa un registrado rígido a partir de al menos tres pares de puntos correspondientes. Se implementa la solución propuesta por K.P. Horn [31] que minimiza el error en la alineación de los puntos en el sentido de mínimos cuadrados.
- Se muestra el error RMS de registrado. Si es aceptable se guarda la transformación; de lo contrario se procede a extraer nuevamente la posición de los fiduciaros.

7.5.2. Aplicación principal

La aplicación principal se desarrolló utilizando la biblioteca IGSTK (ver Capítulo 10). En la misma existen clases encargadas de modelar el comportamiento de los distintos elementos que forman parte de la escena. En particular existen las siguientes clases:

- **Tracker**: Modela el comportamiento de los diferentes sistemas de localización comerciales que existen, siendo uno de ellos el localizador Polaris. Establece la interfaz de comunicación del dispositivo físico con la aplicación de software.
- **TrackerTool**: Modela el comportamiento de los marcadores que se adhieren a las herramientas. En particular almacenan la ubicación de los marcadores respecto al localizador. Pueden existir varios objetos del tipo `TrackerTool` asociados a un mismo `Tracker`. En este caso se utilizó una instancia para modelar el comportamiento de los marcadores adheridos al microscopio y otra para modelar al puntero; éstas se llaman `TrackerToolMicro` y `TrackerToolPuntero` respectivamente.

Estos objetos pueden tener asociada una matriz de calibración. El efecto que tiene asociar una calibración a un `TrackerTool` es que el objeto `TrackerTool` no almacena la posición de la configuración de los marcadores sino que almacena la transformación resultante de la composición de la transformación de los marcadores con la transformación de calibración. Así por ejemplo, si se asocia al `TrackerToolPuntero` como matriz de calibración la traslación O de la punta respecto a los marcadores adheridos a él, el puntero devuelve la posición de la punta y no de los marcadores.

En la Figura 7.6 se muestran dos formas posibles de relacionar al modelo del puntero con el `TrackerTool`. A la izquierda se los relaciona mediante la traslación O obtenida mediante pivoteo y que brinda la ubicación de la punta respecto a los marcadores. A la derecha se relaciona al modelo con el

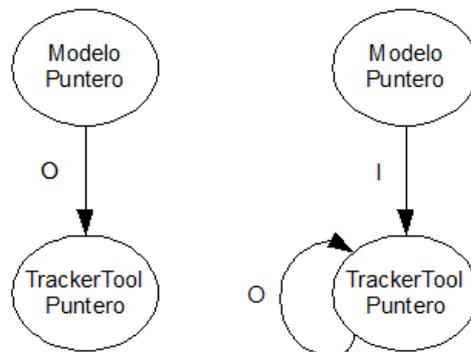


Figura 7.6: Posibles configuraciones para un modelo de la punta

TrackerTool a través de la matriz identidad I pero se asocia al TrackerTool la matriz de calibración O .

Si el modelo del puntero fuese el punto $(0, 0, 0)$, en ambos casos éste se dibujaría exactamente en la posición de la punta.

- **View:** Esta clase representa al mundo virtual, el cual tiene asociada una cámara virtual que es quien captura las distintas escenas. Para que los modelos gráficos generados se proyecten en la imagen en la misma posición que los reales (o de la misma forma que lo harían si fueran visibles) es necesario configurar dicha cámara siguiendo los pasos que se detallaron en el Capítulo 6.

Modelos gráficos Para generar una aplicación de Realidad Aumentada no alcanza sólo con tener un mundo virtual y una cámara sino que es necesario además contar con objetos y representaciones de los mismos. Para el caso de esta aplicación básica se crearon los siguientes modelos:

- **Modelo del Puntero:** Se lo generó utilizando la aplicación SketchUp [26].
- **Modelo de algunos de los fiduciaros adheridos al cráneo de prueba.**
- **Modelo de una estructura que simula ser una lesión**

Estructura de la escena Una vez que se ha decidido cuáles son los objetos que interesa mostrar y se han creado modelos de los mismos es necesario establecer la relación que existe entre ellos; a esto se le llama grafo de la escena. La Figura 7.7 muestra el grafo para esta aplicación en la que se desea superponer en la imagen del video los modelos del puntero y de un tumor.

Para conocer la posición espacial de los marcadores adheridos al microscopio y al puntero se asocian los objetos TrackerToolPuntero y TrackerToolMicro al Tracker. El TrackerToolPuntero tiene asociada la matriz de calibración O por

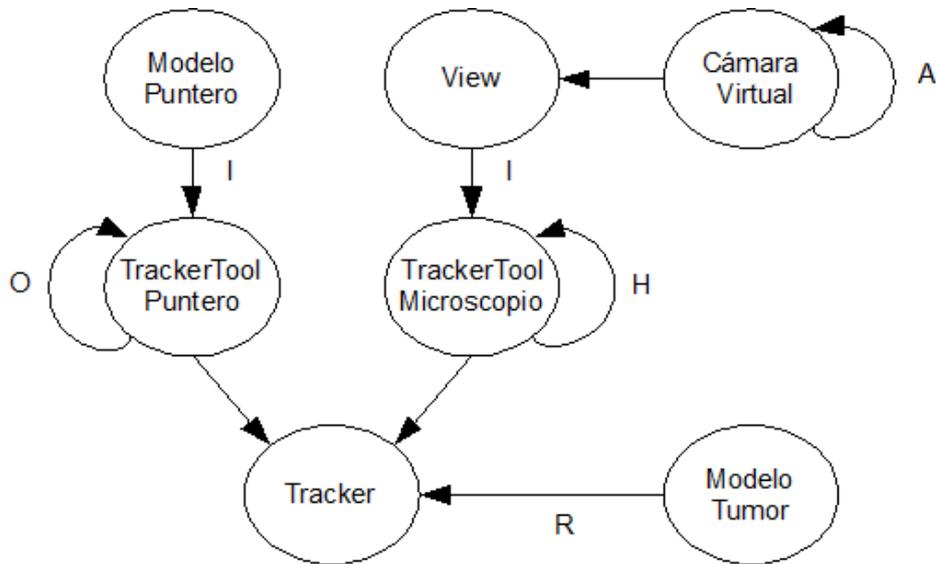


Figura 7.7: Grafo de la Escena

lo que almacena la ubicación de la punta y no de sus marcadores. A su vez está relacionado con el modelo del puntero mediante la identidad.

Mediante la calibración del puntero se extrajo sólo la posición de la punta respecto a los marcadores, pero no se conoce con certeza la orientación de la misma. A los efectos de crear el modelo se supuso que la punta de la herramienta está orientada según el eje z de los marcadores adheridos a ella.

Al `TrackerToolMicro` se le asocia como matriz de calibración a la posición de la cámara respecto a los marcadores (H).

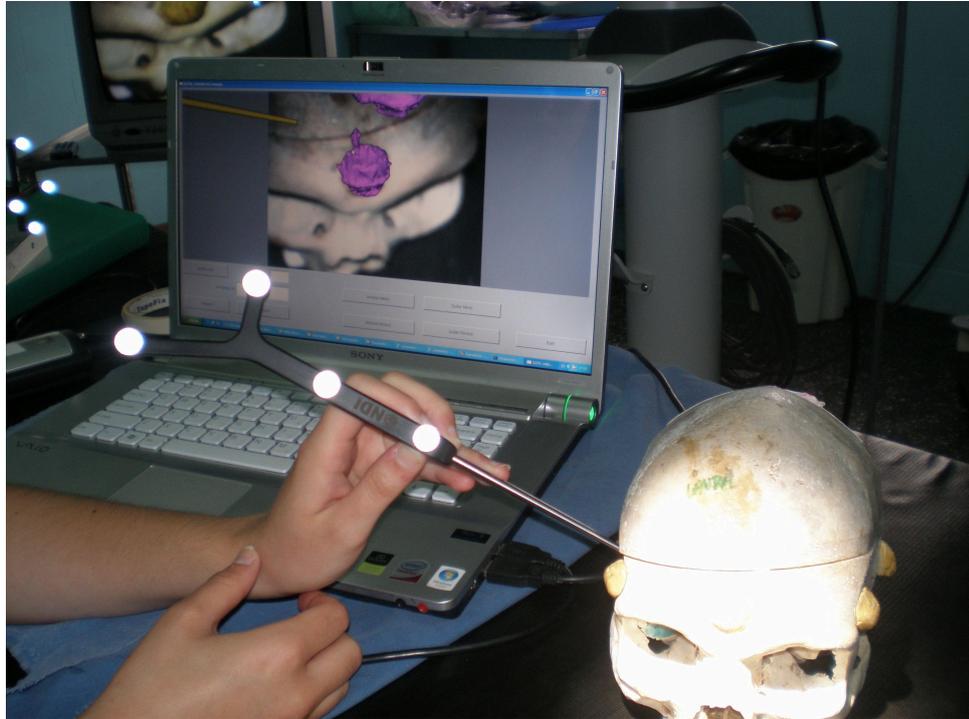
Por otro lado, se configura la cámara virtual. Se desea que la cámara virtual proyecte la imagen igual que lo hace la cámara del microscopio, por eso se configura la misma utilizando los parámetros intrínsecos (A) obtenidos durante la calibración.

Como se vio en el Capítulo 6 la cámara virtual de la escena se debe orientar de forma que quede mirando hacia los z positivos y con el eje y hacia abajo, que es la configuración del modelo de la cámara durante la calibración. Una vez que ha sido orientada de esa manera sólo resta relacionar el `View` con el `TrackerToolMicro` (que es quien almacena la posición de la cámara).

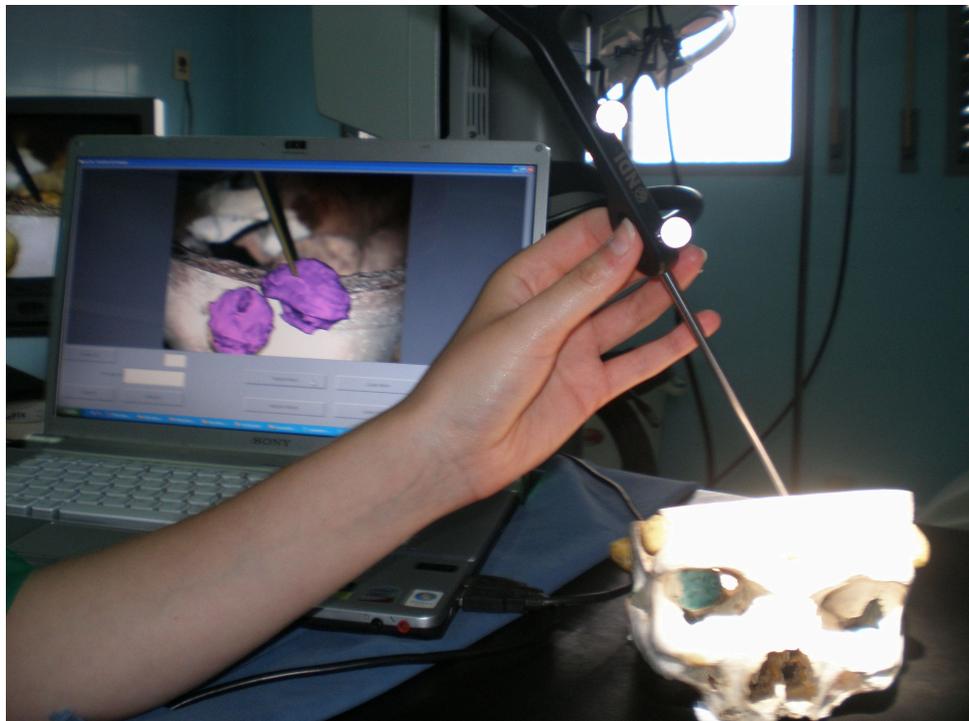
Para finalizar se procede a ubicar el modelo del estudio en la sala de operaciones virtual; para ello se utiliza la transformación de registrado.

Además de los elementos básicos que se muestran en el diagrama podrían aparecer otros; podrían existir otros objetos `TrackerTool` y otros modelos asociados a ellos, o incluso asociados al microscopio.

7.6. Resultados

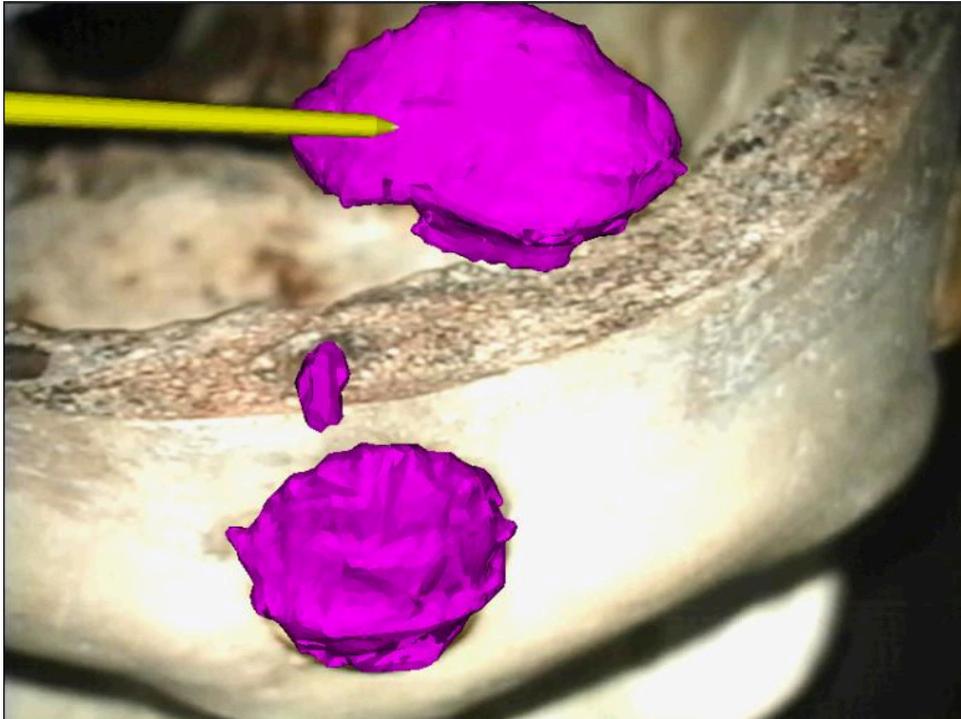


(a)

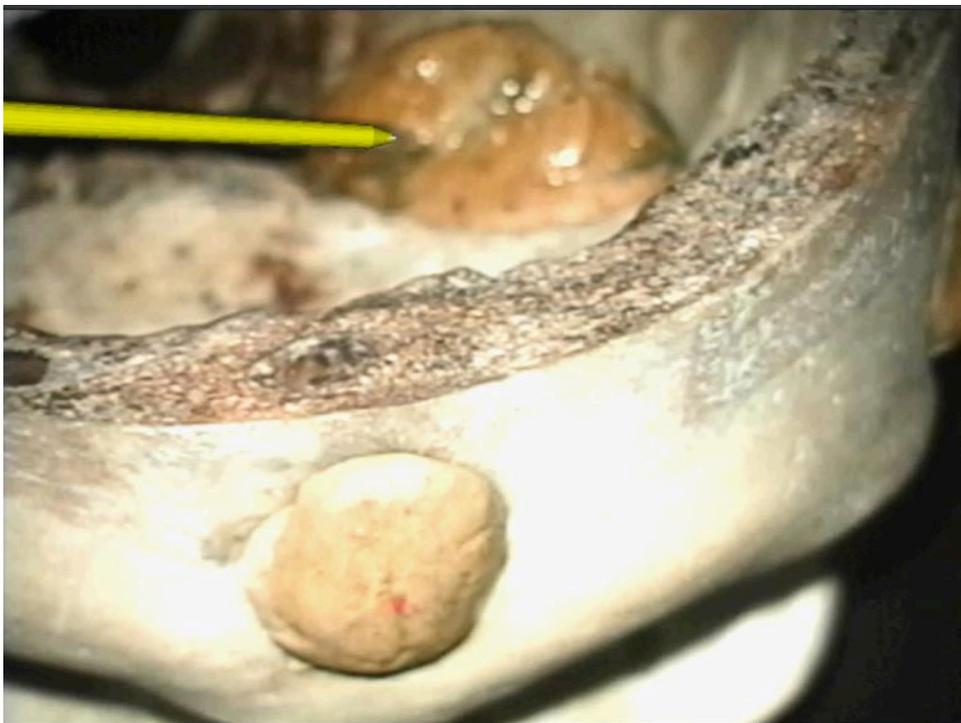


(b)

Figura 7.8: Se observan el cráneo de pruebas, el puntero y una computadora que muestra el video con realidad aumentada



(a)



(b)

Figura 7.9: Modelos añadidos virtualmente

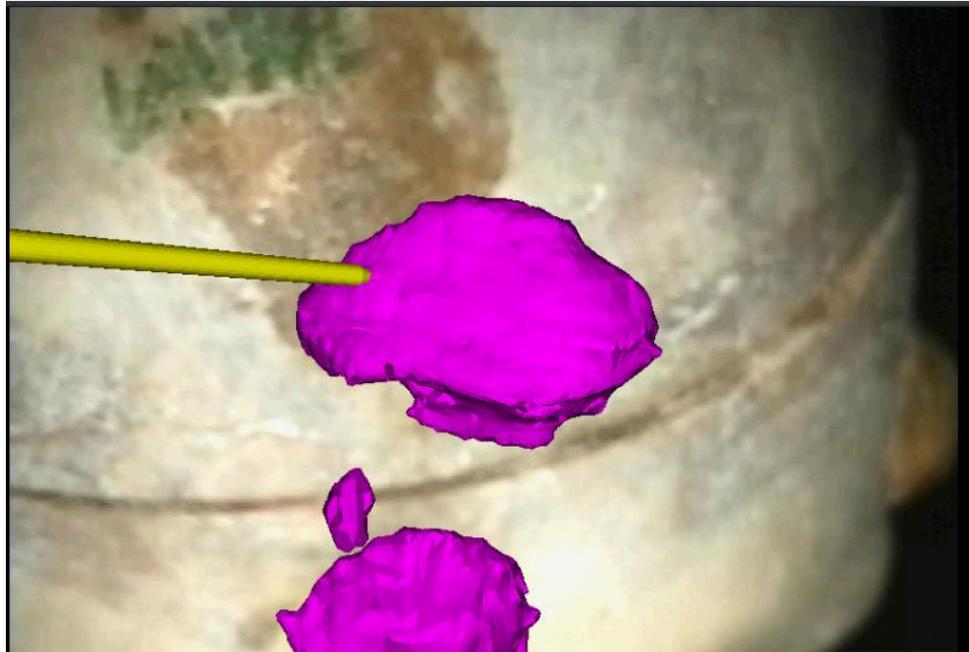


Figura 7.10: El puntero y la estructura que está tocando se encuentran dentro del cráneo cerrado

7.7. Reslicing

Cuando se agrega un modelo en la escena virtual se observa la proyección del mismo en la imagen. Si bien esto permite tener una idea de dónde se encuentra el objeto, sucede que debido a la proyección, se pierde noción de la distancia respecto al microscopio. Para enriquecer la información que se muestra al cirujano se implementó un método mediante el cual se muestra en pantalla la intersección entre el modelo gráfico y un cierto plano de corte.

Para que esta información sea de utilidad para el neurocirujano es importante la elección que se realice del plano de corte. Parece razonable suponer que el mismo pase por el punto que el cirujano está señalando en un momento ya que seguramente ese sea su punto o zona de interés. Para determinar la orientación del plano de corte surgen dos candidatos naturales: el puntero y el microscopio. Se resolvió utilizar como plano de corte aquel que pasa por el punto que está marcando el cirujano con la herramienta y es normal al eje óptico del microscopio.



(a)



(b)

Figura 7.11: Intersección entre el modelo 3D del tumor y el plano que pasa por la punta de la herramienta y es perpendicular al eje óptico para dos profundidades distintas

7.8. Fuentes de error

Las fuentes de error son múltiples y pueden ser divididas en tres categorías:

- **Calibración del microscopio:** en el Capítulo 5 se realizó el estudio del error asociado a la calibración. La práctica demostró que ésta es la principal fuente de error. Para estimarlo se re proyectaron los puntos de un damero que no había sido utilizado en el proceso de calibración.

Una fuente común de error a considerar, vinculada al proceso de calibración del microscopio, es el movimiento de los marcadores adheridos al mismo. Se constató que un pequeño movimiento de los marcadores hace que la matriz H que se obtuvo durante la calibración ya no represente la ubicación del centro óptico. En especial, la componente de rotación de la transformación es muy sensible, ya que una pequeña variación en ésta (un toquecito) produce un error de decenas de píxeles en la imagen.

- **Registrado:** Durante el registrado se realizó la alineación de un conjunto de puntos (del paciente) con otro (puntos del estudio). Como medida de error en el cálculo de la transformación de registrado se utilizó el FRE (Fiducial Registration Error). El FRE [66, 20, 44] es la diferencia cuadrática media entre los puntos fiduciaros marcados con el localizador y los puntos correspondientes en el estudio, luego de aplicada la transformación de registrado.

Desde el momento en que se le realizaron los estudios hasta ahora el cráneo se ha ido deteriorando; concretamente se han caído trozos de los fiduciaros que se muestran en el estudio, lo cual dificulta la selección de los puntos espaciales y la segmentación de las estructuras.

El error en la etapa de registrado rondó entre 1 y 2 mm.

- **Seguimiento:** Si se dejan los objetos estáticos en la escena y se observa la superposición de los distintos modelos en la imagen se aprecia un cierto temblor de los mismos. La amplitud del movimiento es directamente proporcional al aumento del lente y a la distancia de los marcadores adheridos a los patrones. En general ronda entre 1 y 3 píxeles.

Notar que, debido a la forma en que se evalúa el procedimiento de calibración, cuando se estima el error de calibración también se está estimando el error debido al seguimiento de los marcadores adheridos al microscopio y de los marcadores adheridos al patrón.

7.9. Métodos de validación de los resultados

El correcto alineamiento entre el mundo virtual y el real en los sistemas de RA, y en particular en sistemas médicos es un aspecto crítico ya que un error de unos pocos

milímetros puede inducir al cirujano a cometer un error importante. No obstante, no es sencillo determinar con precisión el error que se está cometiendo, más allá de la simple inspección visual. El método visual de chequeo de la calibración se realiza en tiempo real de la siguiente manera:

- Se coloca el patrón damero de forma que los marcadores adheridos a él sean visibles por el localizador y el damero sea visible por el microscopio.
- Se muestra sobre el damero un cubo de arista igual al doble del tamaño de sus casillas.
- Se mueve el patrón y se observa si el cubo dibujado en pantalla acompaña el movimiento y si su tamaño es el correcto.

7.10. Conclusiones

Se desarrolló e implementó un sistema de Realidad Aumentada utilizando el microscopio quirúrgico que se encuentra en el Hospital de Tacuarembó. Se probó y evaluó positivamente el sistema para valores de aumento menores a 3. Los errores obtenidos, luego de toda la cadena de transformaciones involucradas estuvieron en un entorno de 4 píxeles trabajando con una resolución de las imágenes de 640×480 . Nos parece que mejorando ciertos aspectos se puede disminuir considerablemente el error. En particular, sería deseable contar con estudios realizados a un cráneo nuevo ya que el actual se ha deteriorado; esto mejoraría considerablemente la etapa de registrado.

El proceso de calibración, cuando exitoso, demostró funcionar correctamente. Sin embargo, no se logró determinar un procedimiento que funcione para el cien por ciento de los intentos. Sí se determinaron empíricamente una serie de factores que influyen en el éxito o no de la misma; éstos son:

- **Movimiento del patrón:** es importante que el patrón permanezca quieto en el momento de extraer las imágenes para que el cuadro obtenido coincida exactamente con la posición determinada por el sensor Polaris.
- **Distribución de los puntos:** deben cubrir la mayor parte posible del campo de visión del microscopio.

7.11. Trabajo para hacer en el futuro

Hubo algunos aspectos que no fueron estudiados en profundidad y que tal vez deban ser atacados en el futuro. Éstos son:

- **Marcadores adheridos al microscopio** Para seguir al microscopio se utilizó una sola herramienta con cuatro marcadores. Si bien el localizador Polaris es muy preciso, entendemos que una mayor cantidad de marcadores, distribuidos de forma tal que su baricentro coincida (idealmente) con el centro óptico del

microscopio, podría mejorar los resultados de la calibración. En particular, creemos que la transformación que ubica al centro óptico respecto a los marcadores, se haría más robusta frente a rotaciones del microscopio. Al colocar más marcadores también se podría mejorar la inmunidad al ruido.

- **Actualización de la calibración** La práctica ha demostrado que luego de un tiempo el sistema comienza a variar y los parámetros obtenidos durante la calibración dejan de ser buenos. Una razón para ello podría ser un pequeño movimiento de los marcadores adheridos al microscopio. Esto no debería sorprender ya que una pequeña variación en la posición de los mismos podría introducir un error significativo en la proyección de los puntos espaciales.

Este fenómeno hace dudar sobre si la calibración es un proceso que se puede realizar totalmente fuera de la sala. En principio debería serlo; de hecho en las pruebas realizadas con cámaras domésticas los parámetros obtenidos funcionan, incluso con el correr de los días.

En [60] se propone un método de actualización de la de calibración que asume que los parámetros intrínsecos de la cámara no cambian durante la operación y que los errores de proyección se deben a variaciones en los parámetros extrínsecos. Se propone un método de actualización de los parámetros extrínsecos que básicamente consiste en establecer una correspondencia entre puntos espaciales y puntos en la imagen durante la operación y a partir de ellos, manteniendo fijo los parámetros intrínsecos, encontrar la posición del centro óptico que minimice el error de reproyección.

- **Variaciones en el aumento y la distancia de trabajo** Entendemos que aún no se ha resuelto completamente el sistema para un aumento y una distancia de trabajo determinada por lo que parecería ser un tanto apresurado intentar resolverlo para todo el rango de estas variables. En general lo que se hace es calibrar el sistema para diferentes valores de aumento y distancia de trabajo y luego realizar una interpolación para aquellos valores en que no se cuenta con información [21, 74]. La diferencia entre los métodos radica en la forma de realizar la interpolación.

Cabe mencionar que en ninguno de los microscopios con los que se hicieron pruebas existía la posibilidad de extraer la información de aumento y distancia de trabajo en un momento dado, más allá de la lectura de un display.

Capítulo 8

Reconstrucción Estéreo

8.1. Introducción

La obtención de información 3D precisa a partir de lo visto por dos o más cámaras ha sido un tema de investigación y desarrollo en las últimas décadas. El creciente interés que esto ha despertado se debe fundamentalmente a dos razones. Por un lado a la aplicación de estas técnicas en áreas como la arquitectura (para el diseño, conservación y restablecimiento de obras), el reconocimiento de caras y la animación virtual. Un segundo factor que sin duda ha influido decisivamente y ha acelerado e incrementado su interés es el desarrollo de las computadoras y las tarjetas gráficas. Si bien la teoría detrás de la reconstrucción 3D no es nueva, las áreas de aplicación solían verse limitadas por el alto costo computacional necesario para correr los algoritmos originados por la teoría.

8.2. Visión estéreo

Los ojos, y el posterior tratamiento que le da el cerebro a las señales que ellos captan, son los responsables de mostrarnos el espacio de la forma en que lo apreciamos, con nociones de profundidad, distancia, formas y tamaños. Esto es así dado que el ser humano es capaz de reconstruir una escena en el espacio a partir de dos imágenes (capturadas por los ojos). En la actualidad, experiencias como ver cine en tres dimensiones ponen de manifiesto este fenómeno. En cada uno de los oculares de los lentes se proyectan imágenes prácticamente iguales, pero con una diferencia en la posición relativa de los objetos que son percibidos. Esta diferencia, llamada *disparidad*, es la que permite al cerebro determinar la profundidad de los objetos.

8.2.1. No invertibilidad del proceso de formación de la imagen

El proceso de formación de la imagen se puede pensar como una transformación que lleva puntos del espacio 3D en puntos en la imagen 2D. Esto implica necesariamente una pérdida de información. La pérdida se debe a que, dado un punto

en el espacio y la recta definida por éste y el centro óptico de la cámara, todos los puntos pertenecientes a esta recta se proyectan sobre el mismo punto del plano de la imagen. Se pierde así la profundidad del punto. En este sentido, el proceso de formación de la imagen es no invertible: dado un punto en la imagen, no es posible determinar a qué punto del espacio corresponde, sino que sólo puede determinarse una recta a la que pertenece.

Se puede, sin embargo, recobrar la información perdida a partir de al menos dos imágenes de la misma escena tomadas desde diferentes posiciones. En el caso de este proyecto, uno de los objetivos es reconstruir la superficie del cerebro a partir de lo visto por un par de imágenes tomadas con un microscopio de uso quirúrgico.

Para comprender los fundamentos que aparecen detrás de la teoría y los algoritmos de tratamiento de imágenes estereo, es necesario introducir primero los conceptos referidos a la geometría epipolar. En esencia, esta geometría combina dos modelos pinhole (uno para cada cámara) y establece la relación entre la proyección de un punto del espacio en una imagen y la proyección del mismo punto en la otra imagen. A continuación se describen conceptos que se utilizarán en el transcurso del presente capítulo y del siguiente.

8.3. Descripción de la geometría de un par estereo

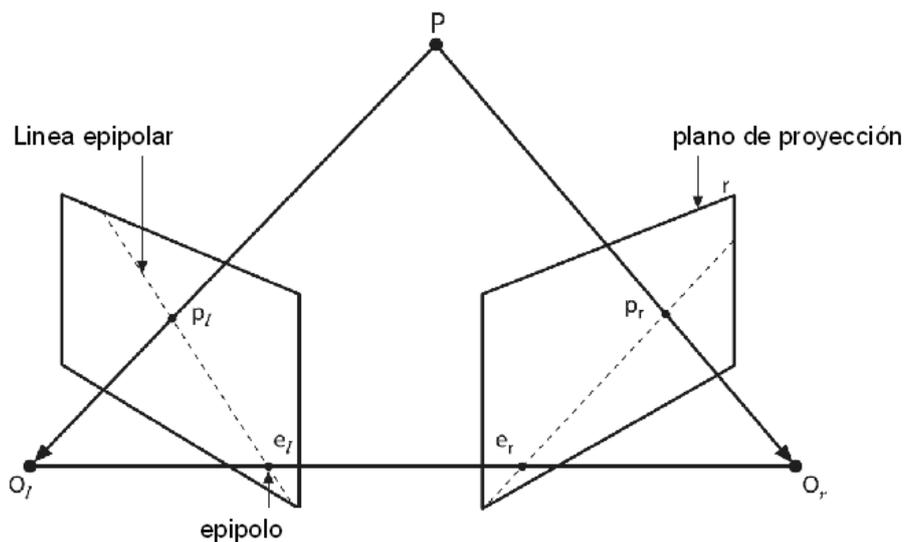


Figura 8.1: Conceptos de la geometría epipolar

En la Figura 8.1 se esquematizan los principales conceptos que aparecen en un par estereo. Éstos son:

- O_l y O_r : centros de proyección de las cámaras izquierda y derecha respectivamente.

- Π_l y Π_r : planos de las imágenes de las cámaras izquierda y derecha respectivamente.
- P : un punto del espacio.
- p_l y p_r : las proyecciones del punto P sobre los planos de las imágenes izquierda y derecha respectivamente.
- e_l y e_r : epipolos de las cámaras izquierda y derecha respectivamente.

Dadas dos cámaras, cada una de ellas con un centro de proyección y un plano de imagen, un punto P del espacio se proyecta sobre los planos de las imágenes de las cámaras en los puntos p_l y p_r . Los puntos que interesa definir con claridad son los *epipolos* e_l y e_r . El epipolo e_l de la cámara izquierda es la proyección del centro de la cámara derecha, O_r , sobre el plano de la imagen izquierda, Π_l . De forma análoga, el epipolo e_r es la proyección de O_l sobre Π_r .

Se define además el *plano epipolar* como el plano que pasa por el punto P y los centros de proyección de las cámaras O_l y O_r . De su definición se desprende que los epipolos e_l y e_r y las proyecciones del punto P sobre las cámaras (p_l y p_r) pertenecen a este plano. La intersección del plano epipolar con los planos de las imágenes de las cámaras son las rectas $p_l e_l$ y $p_r e_r$ para el plano izquierdo y derecho respectivamente. A estas rectas se les llama *líneas epipolares* y son de vital importancia en el proceso de reconstrucción estéreo. Este hecho se hará evidente en las siguientes secciones. Resumiendo, se han definido hasta ahora:

- Los epipolos.
- El plano epipolar.
- Las líneas epipolares.

Observando las definiciones anteriores, es claro que la proyección sobre los planos Π_l y Π_r de cualquier punto en el espacio que pertenezca al plano epipolar pertenecerá a las líneas epipolares determinadas por ese plano epipolar. Utilizando el razonamiento anterior se deduce que todos los puntos del espacio que se proyectan sobre una línea epipolar tendrán su correspondiente en la otra imagen sobre la otra línea epipolar. Este hecho tiene gran importancia, ya que como se verá en los siguientes apartados, para poder realizar una reconstrucción estéreo es necesario poner en correspondencia las proyecciones de un punto en ambas imágenes; esta información facilita la búsqueda de dichas correspondencias.

A continuación se resumen algunas propiedades importantes de la geometría epipolar:

- Todo punto 3D visible por ambas cámaras está contenido en un plano epipolar que intersecta a cada imagen en una línea epipolar.

- Dada la proyección de un punto del espacio en una imagen, su correspondiente proyección pertenece a la línea epipolar de la otra imagen. Ésta es la denominada *restricción epipolar*.
- La restricción epipolar facilita la búsqueda de puntos correspondientes, resultando en una búsqueda de una sola dimensión.
- Dados dos puntos A y B , si sus proyecciones en una imagen pertenecen a una línea epipolar y aparecen en un cierto orden horizontal, sus proyecciones correspondientes en la otra imagen pertenecerán a la línea epipolar y aparecerán en el mismo orden.

8.4. Matriz esencial y matriz fundamental

Ahora se introducen dos objetos matemáticos muy importantes para lo que sigue. Éstos son la *matriz esencial* E y la *matriz fundamental* F [9, 29, 42]. La matriz esencial contiene información sobre la posición relativa entre las cámaras, es decir, guarda la información de la traslación T y la rotación R que ubican a la cámara derecha respecto de la izquierda, como se puede ver en la Figura 8.2. T y R son los parámetros extrínsecos del par estéreo. Por otro lado, la matriz fundamental F contiene la misma información que la matriz esencial agregando además información sobre los parámetros intrínsecos dados por el modelo pinhole de las cámaras.

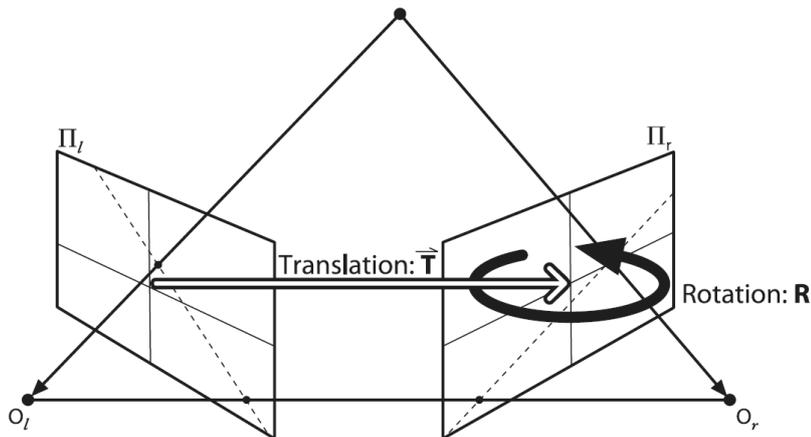


Figura 8.2: Traslación y rotación que describen la posición relativa entre las cámaras. Imagen extraída de [23]

Matriz esencial

Si se toma un punto P en el espacio se puede considerar su posición respecto al sistema de coordenadas de cada una de las cámaras, P_l y P_r , que no son otra cosa que

la posición del punto P respecto al centro de proyección de cada una de las cámaras, O_l y O_r respectivamente. A P_l y P_r se le denominan coordenadas normalizadas. Utilizando esta notación, se puede ver que:

$$P_r = R(P_l - T) \quad (8.1)$$

Se considera ahora el plano epipolar, el cual de forma general se puede representar como en la Ecuación 8.2, donde x es un punto genérico perteneciente al plano epipolar, a es un punto fijo del plano y n es el vector normal al plano:

$$(x - a) \cdot n = 0 \quad (8.2)$$

Recordando que el punto P_l pertenece al plano epipolar y notando que el punto T también está contenido en este plano, lo anterior se puede reescribir como:

$$(P_l - T)^T \cdot (T \times P_l) = 0 \quad (8.3)$$

La Ecuación 8.1 es equivalente a

$$(P_l - T) = R^{-1}P_r \quad (8.4)$$

Recordando que $R^T = R^{-1}$ por ser una matriz de rotación, y sustituyendo lo anterior en 8.3 se tiene que:

$$(R^T P_r)^T (T \times P_l) = 0 \quad (8.5)$$

Siempre es posible ver un producto vectorial entre dos vectores como el producto entre un vector y una matriz correctamente definida. De esta manera, el producto vectorial $T \times P_l$ puede expresarse como

$$T \times P_l = SP_l \quad (8.6)$$

donde

$$S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$$

Sustituyendo en la Ecuación 8.5:

$$P_r^T R S P_l = 0 \quad (8.7)$$

El producto RS es lo que se define como la matriz esencial E :

$$P_r^T E P_l = 0 \quad (8.8)$$

La ecuación anterior puede reescribirse para introducir las coordenadas de la proyección del punto P sobre las imágenes, p_l y p_r . Utilizando el modelo pinhole de las cámaras se tienen las siguientes relaciones:

$$\begin{aligned} p_l &= f_l P_l / Z_l \\ p_r &= f_r P_r / Z_r \end{aligned}$$

por lo que a menos de un factor multiplicativo ($Z_l Z_t / f_l F_r$) la Ecuación 8.8 queda:

$$p_r^T E p_l = 0 \quad (8.9)$$

Como se desprende de la Ecuación 8.9, la matriz esencial es quien relaciona las proyecciones de un punto sobre los planos de las imágenes de ambas cámaras. En general no se conocen las coordenadas de los puntos p_l y p_r , pero sí la posición en píxeles de la proyección del punto P sobre las imágenes.

Matriz Fundamental

La matriz E contiene toda la información sobre la geometría de las dos cámaras, pero no involucra a los parámetros intrínsecos. Lo que interesa para realizar triangulación estéreo es, más que la información geométrica, la relación entre coordenadas en píxeles en ambas cámaras. Para encontrar esta relación es necesario introducir información sobre los parámetros intrínsecos, pudiendo así relacionar las coordenadas q en píxeles con la posición p en el espacio. Para la cámara izquierda, por ejemplo, se tiene que

$$q_l = A_l P_l$$

donde A_l es la matriz de parámetros intrínsecos de la cámara izquierda. Sustituyendo en 8.8

$$q_r^T (A_r^{-1})^T E A_l^{-1} q_l = 0 \quad (8.10)$$

Definiendo ahora la matriz fundamental F como

$$F = (A_r^{-1})^T E A_l^{-1} \quad (8.11)$$

la Ecuación 8.10 se transforma en:

$$q_r^T F q_l = 0 \quad (8.12)$$

Como se dijo anteriormente, la matriz fundamental F es muy similar a la matriz esencial E , con la diferencia que F trabaja con píxeles de la imagen mientras que E trabaja con coordenadas espaciales.

Esta matriz establece la relación entre las coordenadas en píxeles de puntos correspondientes en ambas imágenes, como se puede observar en la Ecuación 8.12. Otra propiedad que cumple la matriz fundamental es que define tanto los epipolos como las líneas epipolares. Esto se observa respectivamente en las Ecuaciones 8.14 y 8.16, donde se recuerda que $p_l e_l$ y $p_r e_r$ son las líneas epipolares de la cámara izquierda y derecha respectivamente.

$$F e_l = [0 \ 0 \ 0]^T \quad (8.13)$$

$$F^T e_r = [0 \ 0 \ 0]^T \quad (8.14)$$

$$p_l e_l = F p_r \quad (8.15)$$

$$p_r e_r = F^T p_l \quad (8.16)$$

8.5. Reconstrucción estereo y disparidad

Se supone que se cuenta con un par de cámaras que cumplen con el modelo pinhole dispuestas como se muestra en la Figura 8.3, donde los planos de las imágenes de ambas cámaras son coplanares. Además se supone que las cámaras tienen igual largo focal f y que sus centros están relacionados por un vector de traslación T de dirección paralela a la dirección x de las imágenes y módulo B . En estas condiciones, la proyección de un punto P tendrá igual coordenada y en ambas cámaras. Sin embargo, existe una diferencia en la coordenada x ; ésta es llamada *disparidad* d . Como puede deducirse de la Figura 8.3, la disparidad está vinculada con la profundidad Z del punto P , y su relación está dada por:

$$Z = \frac{fB}{d}$$

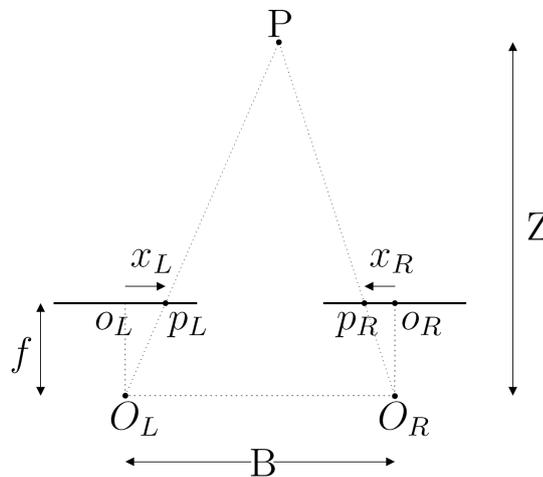


Figura 8.3: Geometría de un par estereo para la definición de la disparidad. Imagen extraída de [40]

La ecuación anterior muestra que en estas condiciones es muy sencillo recuperar la profundidad del punto P si se conoce la proyección del mismo en ambas imágenes. Al procedimiento de encontrar la profundidad de un punto espacial a partir de su proyección en dos imágenes se le conoce como *triangulación*.

Sin embargo, esta configuración es muy difícil de lograr en la práctica. En el caso general, partiendo de una configuración diferente, será necesario aplicar una

serie de transformaciones matemáticas hasta llegar a las imágenes que se hubiese obtenido de haber contado con las cámaras en la posición ideal. Hecho esto se podrán aplicar los algoritmos existentes para calcular la profundidad de un punto a partir de su disparidad en un par estéreo.

8.6. Explicación detallada del proceso de reconstrucción

El proceso de recuperación de la estructura de una escena se puede separar en tres pasos: encontrar la proyección de un punto característico de la escena en las dos imágenes (puntos correspondientes, Figura 8.4), medir la diferencia relativa entre estos (disparidad) y por último utilizar esta información junto con a la posición relativa entre las cámaras para triangular. Cuando se desea implementar este procedimiento, resulta en que la búsqueda de puntos correspondientes entre ambas imágenes es computacionalmente costosa y complicada. Para intentar solucionar este problema, se suele agregar una etapa previa a la búsqueda de correspondientes mediante la cual, utilizando información sobre la geometría del par estéreo, se rectifican las imágenes. Las imágenes están rectificadas si las cámaras se encuentran como en la Figura 8.3, donde los sistemas de coordenadas de ambas cámaras difieren únicamente en una traslación y las distancias focales son iguales. El siguiente es un esquema de cómo se resuelve la reconstrucción estéreo y a continuación se detalla cada uno de los pasos.

- Calibración del par estéreo.
- Rectificación de las imágenes.
- Búsqueda de correspondientes.
- Triangulación.

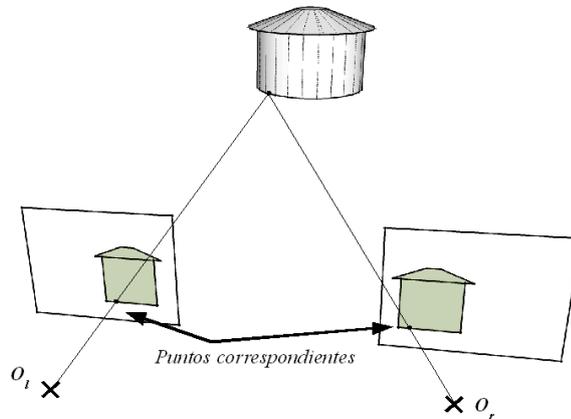


Figura 8.4: En la figura se ve un punto en el espacio y sus proyecciones sobre las imágenes; estas proyecciones son puntos correspondientes.

8.7. Calibración estéreo

La calibración estéreo es el proceso mediante el cual se encuentra la relación geométrica en el espacio entre las dos cámaras, dada por una matriz de rotación R y un vector de traslación T . El proceso de calibración estéreo es muy similar al de la calibración de cada una de las cámaras, salvo que ahora se cuenta con varios pares de imágenes de la misma escena (se supone que la escena no cambia entre la adquisición de las dos imágenes). Además, para todos los pares de imágenes, las cámaras deben mantener constante su posición relativa. Se supone además que previamente se ha calibrado cada cámara por separado, por lo que son conocidos los parámetros intrínsecos de cada una de ellas. En estas condiciones, el proceso de calibración consta de los siguientes pasos:

- Se detecta en ambas imágenes las esquinas del patrón de calibración (coordenadas en píxeles).
- Se asignan posiciones en el sistema de referencia espacial (coordenadas 3D) a los puntos del patrón.
- Se extraen los parámetros extrínsecos de cada cámara, es decir, dos matrices de rotación R_{r_i} y R_{l_i} y dos vectores de traslación, T_{r_i} y T_{l_i} , que ubican a ambas cámaras respecto al sistema de referencia espacial.
- Ahora que se tiene a ambas cámaras referidas al mismo sistema de coordenadas, se extrae la posición relativa entre ellas, R_i y T_i . Las ecuaciones para calcularlas son

$$\begin{aligned} R_i &= R_{r_i}(R_{l_i})^T \\ T_i &= T_{r_i} - R_i T_{l_i} \end{aligned}$$

- Se repiten los pasos anteriores para todos los pares de imágenes que se tiene y se calcula una estimación inicial para R y T como la mediana de todos los vectores $om(R_i)$ y T_i , donde $om(R_i)$ es la representación de la matriz R_i según la fórmula de Rodrigues [73].
- Con la estimación inicial calculada se busca el conjunto de parámetros extrínsecos R y T que minimiza el error de reproyección. Se utiliza el algoritmo de Levenberg-Marquardt [70].

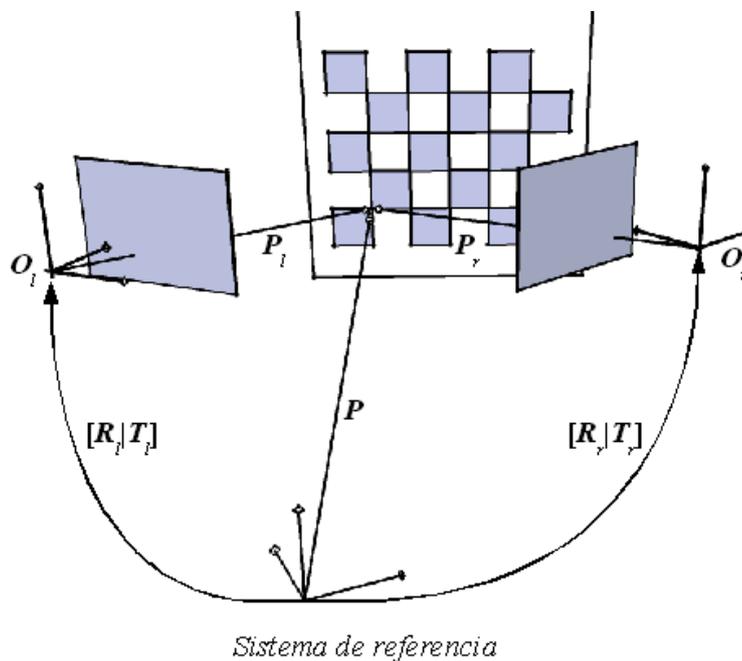


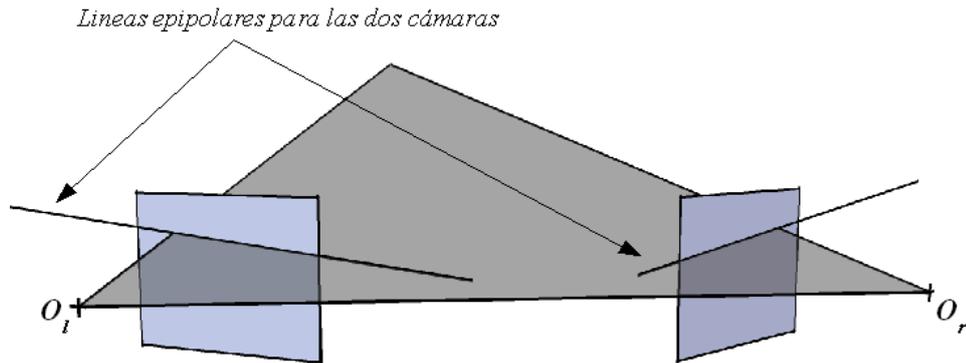
Figura 8.5: Movimientos para escribir las coordenadas de un punto P referido al mundo en coordenadas normalizadas de las cámaras

Como comentario final, se destaca que la rotación R que brinda la calibración estereo es tal que permite alinear los ejes de los sistemas de referencia de las cámaras, o lo que es lo mismo, hacer que ambos planos imágenes queden paralelos, pero esto no implica que las filas de las imágenes estén alineadas. Sin embargo, es posible lograrlo mediante el proceso de rectificación que se explica a continuación.

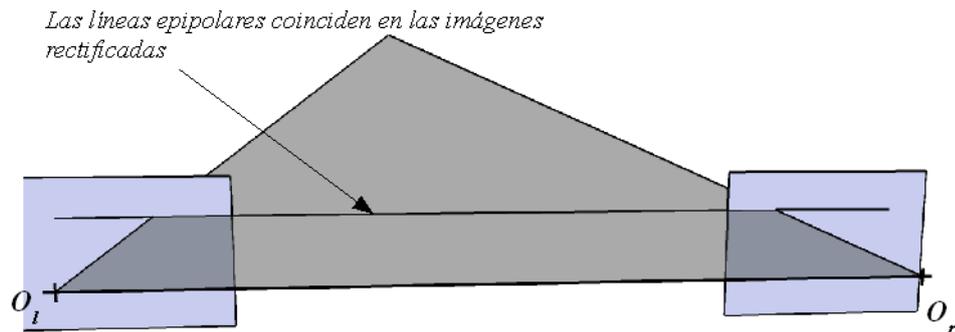
8.7.1. Rectificación

Como se mencionó anteriormente, se puede recuperar información de la profundidad de una escena a partir de la correspondencia de puntos en las imágenes. Para aumentar la eficiencia computacional de esta búsqueda, es deseable rectificar las imágenes, es decir, llevarlas a una configuración como la que se describió en

la Sección 8.5 (ver Figura 8.3). Si dos imágenes están rectificadas, sus epipolos se encuentran en el infinito, en la dirección de la recta $O_l O_r$, o lo que es lo mismo, sus líneas epipolares se intersectan en el infinito. En la Figura 8.6 se muestra cómo quedan las cámaras cuando se realiza la rectificación.



(a) Líneas epipolares para las imágenes originales



(b) Líneas epipolares en las imágenes rectificadas

Figura 8.6: Cambios en las líneas epipolares al rectificar

Hay que destacar que existen varias formas de rectificar las imágenes dado que existen en principio muchas formas de colocar los planos de imagen para que cumplan con las condiciones de rectificación. Como las imágenes son obtenidas con el par estéreo real, cuando se realiza la rectificación las nuevas imágenes tendrán menos información de la escena; se aprecia en la Figura 8.7 cómo quedan las imágenes rectificadas. La cantidad de información que se pierde depende de cómo se implemente la rectificación.

A continuación se pasa a detallar el algoritmo de rectificación de las imágenes que implementa la librería OpenCV ¹. Primero se debe mencionar que este algoritmo rectifica de forma de maximizar el área común vista por las imágenes rectificadas. La

¹OpenCV implementa en C++ el algoritmo de calibración que Bouguet desarrolló en Matlab. La



Figura 8.7: En las zonas negras se aprecia la pérdida de información al rectificar

rotación R que rota al plano de imagen derecho en el izquierdo es dividida en dos; al resultado se le llamará r_l y r_r que son rotaciones para la cámara izquierda y derecha respectivamente. Cada cámara es rotada en la mitad de la rotación original R ; es por esto que como resultado se obtiene que los rayos principales de ambas cámaras quedan según la dirección determinada por la bisectriz de los rayos originales. Se debe notar que las rotaciones anteriores dejan a los planos de las cámaras en planos paralelos, pero no aseguran la alineación de las imágenes.

Para que las filas de las imágenes queden alineadas se debe incluir una nueva matriz de rotación R_{rect} . Para rectificar, los ejes de ambas cámaras deben estar alineados (lo cual en este paso ya se cumple, como se aprecia en la Figura 8.8(b)), pero para asegurar que las filas de las imágenes están alineadas debe ocurrir que los ejes x de las cámaras sean colineales con la recta determinada por los centros de proyección de ambas cámaras $O_l O_r$ (Figura 8.8(c)), es decir, deben coincidir con la dirección de T .

Ahora que se sabe cuál es el resultado que se necesita para lograr la rectificación estereo, sólo se debe definir la base en que se pretende que queden alineados los ejes de las cámaras y de esa manera quedará definida R_{rect} . Para conseguir esto se define el primer versor de la base deseada e_1 como

$$e_1 = \frac{T}{\|T\|} \quad (8.17)$$

Dado que las imágenes ya se encuentran en un mismo plano, se debe asegurar que e_3 sea colineal con los rayos principales de las cámaras para no alterar esta condición (se recuerda que en este punto estos rayos ya son colineales y están en la dirección

implementación en C++ contenía errores en la función *cvStereoRectify*. Las correcciones se realizaron comparando esta función con la implementación de Matlab

determinada por la suma de los rayos principales originales de las cámaras). Por otro lado, el vector e_3 debe ser el producto vectorial entre e_1 y e_2 , de donde se deduce que e_2 está contenido en el plano de la imagen. Además, e_2 es ortonormal a e_1 , por lo que la elección para e_2 debe ser:

$$e_2 = \frac{T}{\|T\|} \times (0, 0, 1) = \frac{[-T_y \quad T_x \quad 0]^T}{\sqrt{T_x^2 + T_y^2}}$$

Finalmente,

$$e_3 = e_1 \times e_2$$

resultando en la dirección del eje óptico como se buscaba.

Resumiendo, la matriz de rectificación queda

$$R_{rect} = \begin{bmatrix} e_1^T \\ e_2^T \\ e_3^T \end{bmatrix} \quad (8.18)$$

Esta matriz rota ambas cámaras respecto del centro de proyección y hace que las líneas epipolares queden horizontales y los epipolos en el infinito. La alineación de las filas de las dos imágenes queda impuesta en las siguientes rotaciones para ambas cámaras:

$$R_{rect}^l = R_{rect} r_l \quad (8.19)$$

$$R_{rect}^r = R_{rect} r_r \quad (8.20)$$

R_{rect}^l y R_{rect}^r son las rotaciones con que se rectifican las imágenes de la cámara izquierda y derecha respectivamente. El siguiente paso es ver que los parámetros intrínsecos de las cámaras también van a sufrir modificaciones. Las nuevas matrices de parámetros intrínsecos serán A_{rect_l} y A_{rect_r} . Para profundizar sobre los detalles de esta transformación, se sugiere consultar el Apéndice C de [61].

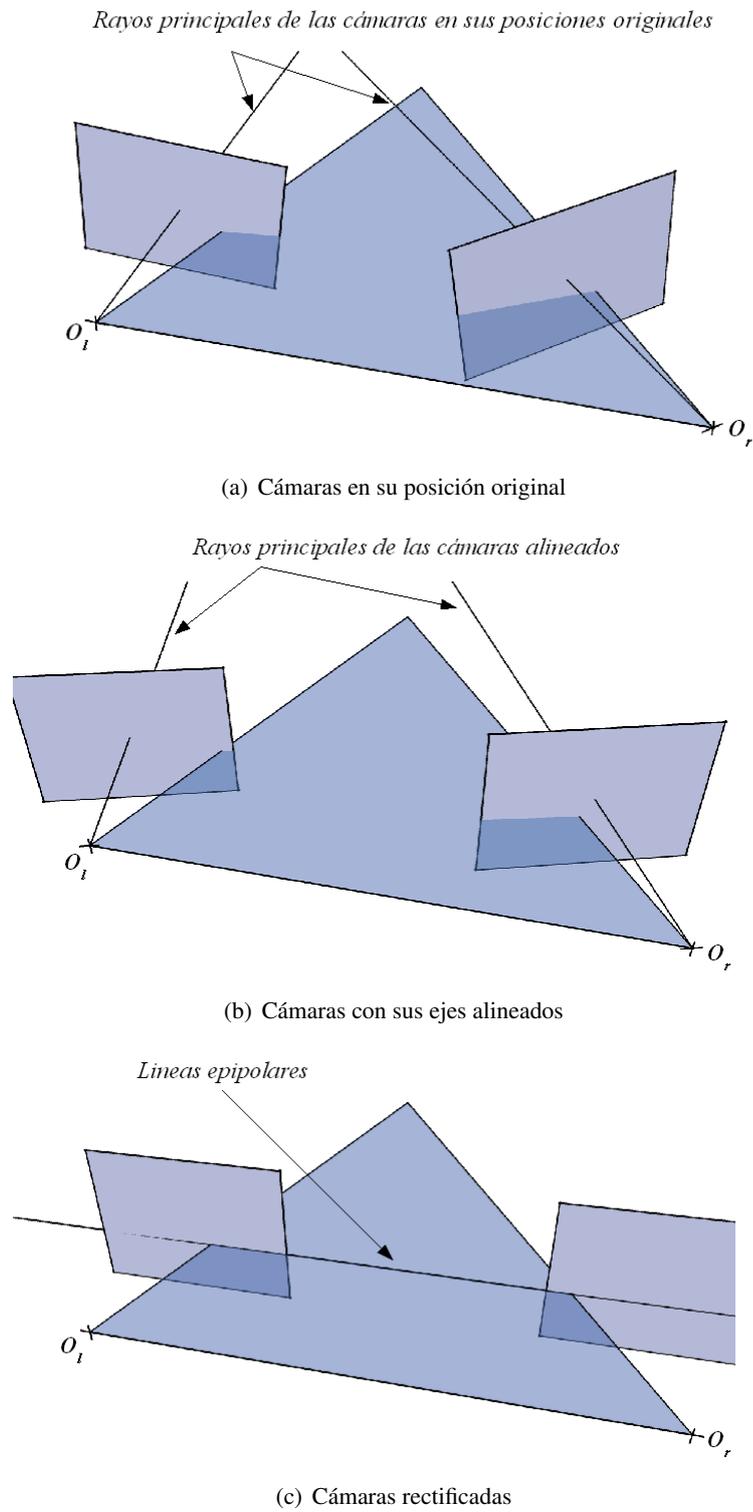
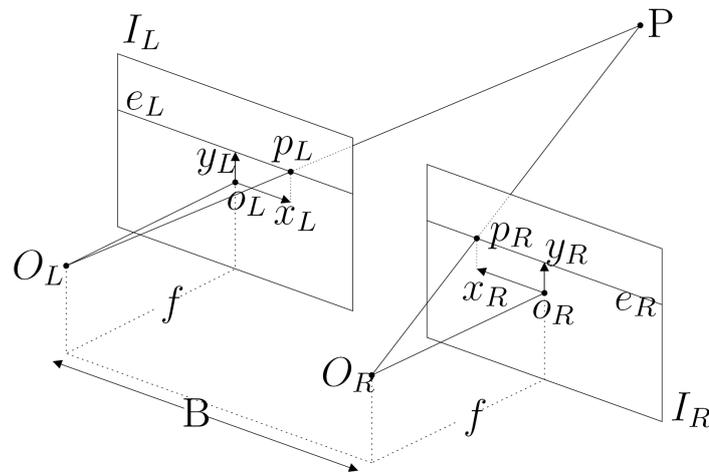


Figura 8.8: Proceso de rectificación paso a paso

8.8. Correspondientes

Para realizar la reconstrucción estereo de toda la escena mostrada por las imágenes, es preciso conocer las correspondencias de todos los píxeles. Para computar estas correspondencias se introduce el concepto de disparidad y mapas de disparidad.



(a) Configuración de cámaras rectificadas en un par estereo

Figura 8.9: Geometría de un par estereo para la definición de la disparidad. Imagen extraída de [40].

Como se puede ver en la Figura 8.9(a), las imágenes I_L e I_R ya están rectificadas. Es por esto que las líneas epipolares e_L y e_R coinciden para ambas imágenes. Dada esta configuración, si un punto P del espacio tiene proyección en la imagen I_L sobre la línea epipolar e_L su correspondiente en la imagen I_R pertenecerá a la línea epipolar e_R . Esto reduce a un grado de libertad la búsqueda de puntos correspondientes. La disparidad se define como la diferencia en la coordenada horizontal entre las proyecciones p_L y p_R , o sea $d = x_L - x_R$ (siempre teniendo en cuenta que las imágenes están rectificadas, por lo que los ejes horizontales de ambas cámaras coinciden). Así, las coordenadas de las proyecciones en las imágenes del punto P se relacionan de la siguiente manera:

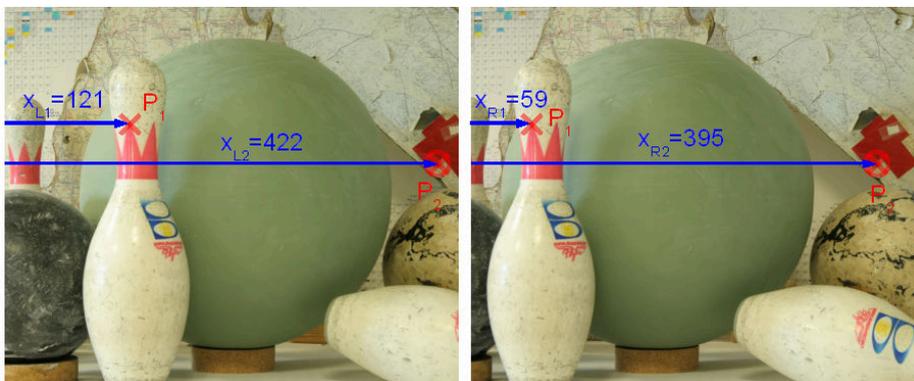
$$\begin{cases} x_L = x_R + d \\ y_L = y_R \end{cases}$$

Utilizando la semejanza de los triángulos $\widehat{PO_L O_R}$, $\widehat{p_R O_R O_L}$, $\widehat{p_L O_L O_R}$ mostrados en la Figura 8.9(a) se llega a la relación:

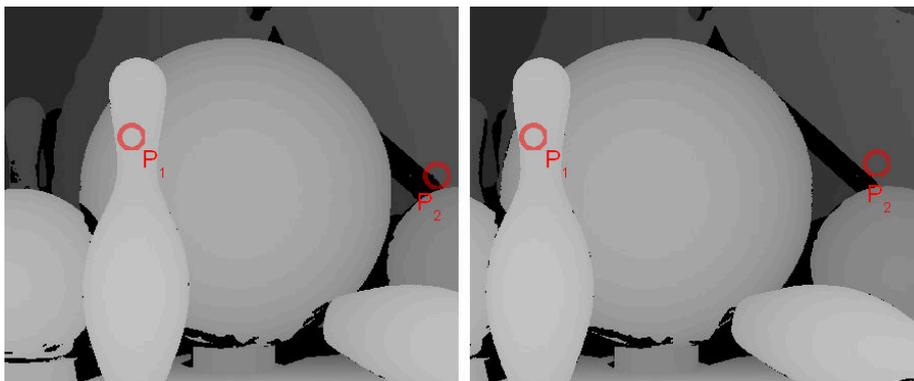
$$d = \frac{f}{Z} B \quad (8.21)$$

Notar que aquí B es el módulo de T , vector de traslación entre los centros de las cámaras.

De la Ecuación 8.21 se deduce que existe una relación inversamente proporcional entre la disparidad y la profundidad. Esto se puede percibir por ejemplo si se observa un objeto cercano primero con un ojo y luego con el otro: se advierte un movimiento del objeto. Si se realiza la misma experiencia con un objeto que está más alejado, el movimiento que se percibe es menor.



(a) Imágenes donde se marcan dos puntos P_1 y P_2 con sus respectivas coordenadas horizontales en ambas imágenes



(b) Mapas de disparidad donde se marcan los puntos P_1 y P_2

Figura 8.10: Mapa de disparidad y relación entre disparidad y profundidad

En la Figura 8.10 se aprecian los mapas de disparidad correspondientes a cada una de las imágenes del par estéreo. Dos puntos de la escena están marcados en las cuatro imágenes, P_1 y P_2 , siendo P_1 más cercano a la cámara que P_2 . Las disparidades obtenidas son:

$$\begin{cases} d = x_R - x_L \\ d_{P_1} = 62 \\ d_{P_2} = 27 \end{cases} \quad (8.22)$$

En los mapas de disparidad se observa que los puntos más alejados se representan con tonos más oscuros. Así se puede ver por ejemplo que el punto P_1 es más cercano a las cámara que P_2 .

Estos mapas son densos ya que brindan información de disparidad para todos los píxeles de la imagen. Sin embargo pueden existir puntos que sean visibles en una imagen y no en la otra (oclusiones) o puntos en los que no se haya podido establecer la correspondencia. Éstos se muestran en negro.

8.9. Algunos algoritmos para el cálculo de la disparidad

El fin de un algoritmo de cálculo de disparidad no es otro que conseguir un mapa denso de disparidades, esto es, saber cuál es el valor de disparidad de cada pixel en la imagen. Estos algoritmos deben determinar la correspondencia de todos los píxeles de la imagen utilizando para ello características visuales como ser: intensidad, color, geometría, movimiento u otras. A la hora de establecer las correspondencias, los algoritmos utilizan ciertas restricciones para facilitar la búsqueda de las mismas. Estas restricciones son utilizadas como hipótesis y provienen de propiedades físicas y geométricas que cumplen los objetos y superficies presentes en la escena. Las cuatro restricciones más importantes son [40]:

- *Restricción epipolar.* Viene dada por la geometría epipolar del par estéreo, e implica que el correspondiente de un punto en una imagen debe estar en la recta epipolar del punto en la otra imagen, limitando la búsqueda de correspondencias a una sola dimensión.
- *Restricción de orden.* Establece que si la proyección del objeto Q está a la izquierda de la proyección del objeto P en la imagen izquierda, entonces la proyección de Q estará a la izquierda de la proyección de P en la imagen derecha. Como ejemplo de esto, se puede intentar mirar ambos dedos índices uno detrás del otro y se puede apreciar un efecto “fantasma” de la imagen.
- *Restricción de unicidad.* Implica que cada punto de una imagen no puede tener más de un correspondiente en la otra imagen. Esta restricción contempla que pueda no existir ningún correspondiente, como puede ser en el caso en que esté oculto en la otra imagen.
- *Restricción de semejanza.* Implica que las características de los puntos en una imagen (intensidad o color, etc.) no deben cambiar mucho entre ambas imágenes.

- *Oclusiones*. Éstas no son restricciones a la hora de realizar la correspondencia entre puntos, pero sí son un problema dado que como ya se mencionó, éstos son puntos que aparecen en una sola imagen por lo que no tienen correspondiente en la otra. Las oclusiones siempre implican una discontinuidad en la profundidad de la escena y son fuente de errores en muchos algoritmos de reconstrucción estéreo.

Los algoritmos que se utilizan para el cálculo de disparidad se pueden dividir en dos categorías según qué características busquen en la imagen; estas categorías son locales y globales [40]. Los primeros son algoritmos que buscan correspondencias en un cierto vecindario, mientras que los globales son aquellos que minimizan alguna función de costo basados en información de toda la imagen.

Dentro del grupo de los algoritmos locales se puede realizar una separación dependiendo de los criterios que utilizan para encontrar las correspondencias.

- *Block Matching*

Cuando se busca la correspondencia de un punto de una imagen en la otra, si las mismas están rectificadas, el punto correspondiente tendrá la misma coordenada vertical que el punto original. A la fila determinada por esa coordenada vertical se le denomina *scanline*. Si se desea encontrar la correspondencia de un punto utilizando Block Matching, se define un vecindario en torno a ese punto y se compara su similitud con vecindarios de la otra imagen, centrados en puntos pertenecientes a la *scanline*. La región que presenta mayor similitud con la original (según un cierto criterio) es la que se selecciona como región correspondiente y el centro de la misma será el *punto correspondiente*. Dependiendo del criterio que se utilice para verificar la similitud entre regiones se pueden encontrar distintas funciones de costos a minimizar; algunas de ellas son:

- *Correlación normalizada (NCC)*

$$\frac{\sum_{u,v} (I_L(u, v) - \bar{I}_L)(I_R(u + d, v) - \bar{I}_R)}{\sqrt{\sum_{u,v} (I_L(u, v) - \bar{I}_L)^2 (I_R(u + d, v) - \bar{I}_R)^2}} \quad (8.23)$$

- *Suma de diferencias al cuadrado (SSD)*

$$\sum_{u,v} (I_L(u, v) - I_R(u + d, v))^2 \quad (8.24)$$

- *Suma de valores absolutos (SAD)*

$$\sum_{u,v} |I_L(u, v) - I_R(u + d, v)| \quad (8.25)$$

- *Optical Flow*

Plantea una ecuación diferencial que relaciona la disparidad d con el movimiento, asumiendo que la intensidad de los puntos no varía.

$$\nabla_x I d + I_t = 0$$

donde $\nabla_x I$ es el gradiente horizontal de la imagen e I_t es la derivada temporal que en este caso se considera nula dado que la escena debe permanecer estática entre las dos imágenes. El desplazamiento vertical del pixel se asume que es nulo ($\nabla_y I = 0$) dada la configuración del par estéreo.

- *Apareamiento de características*²

Se basa en buscar los puntos correspondientes en regiones de la imagen donde existen características relevantes (vértices, bordes, etc). Estos puntos no son muchos, y los mapas de disparidad que se pueden calcular no son densos; se obtienen puntos ubicados en el espacio pero no un mapa de disparidad con la profundidad estimada para cada punto de la imagen. Se destaca que de todas maneras no es necesario obtener un mapa de disparidad denso para realizar la reconstrucción de la escena, sino que alcanza con tener una cantidad suficiente de puntos para reconstruir una malla coherente con la realidad.

A diferencia de los métodos locales, los métodos globales imponen restricciones en toda la imagen (no en vecindarios relativos a cada pixel) en la minimización de alguna expresión de costo o energía que modela el fenómeno estéreo, reduciendo los errores en las regiones donde existen oclusiones o discontinuidades en la disparidad. Se utilizan comúnmente dos tipos diferentes de búsqueda:

- *Programación Dinámica*³

Es un método que reduce la complejidad de cálculo en problemas de optimización descomponiendo el mismo en sub-problemas menores. Las restricciones globales que se imponen con este método son, generalmente, la restricción epipolar y la restricción de orden. Para esto se construye una representación de las posibles correspondencias para cada punto construyendo una imagen que se denomina *imagen del espacio de disparidad* (DSI) donde se busca un camino que recorra este espacio y minimice un cierto costo. La gran desventaja de este método es que al interesarle sólo la globalidad de la imagen en una scanline (se puede pensar que sólo realiza una minimización en una línea epipolar) y no utiliza información de las scanlines adyacentes. Esto le da al mapa de disparidad un efecto de rayado.

²Extraído de [40], Capítulo 3

³Para una mejor comprensión de este tema, consultar el Capítulo 4 de la Tesis de Maestría de Federico Lecumbery [40]

- *Corte de Grafos*⁴

El corte de grafos se basa en armar un grafo a partir de los datos de las imágenes y buscar un corte mínimo. Dependiendo de cómo se arme el grafo, el resultado obtenido es la minimización de una cierta expresión de energía. Este procedimiento se puede considerar análogo al de hallar el mejor camino en una imagen bidimensional utilizando Programación Dinámica, pero extendido a tridimensional con coherencias en las dos dimensiones. El resultado es una superficie que minimiza un costo energético sobre un grafo plano. Estos métodos requieren un costo computacional mucho mayor que el caso de Programación Dinámica.

8.10. Triangulación

Recordando cómo se proyecta un punto en coordenadas normalizadas en la imagen, y asumiendo que luego de la rectificación los nuevos largos focales serán iguales, se tiene que

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (8.26)$$

Resumiendo las ecuaciones antes vistas, a partir de los parámetros intrínsecos de las cámaras y de los parámetros del par estereo se puede recuperar la profundidad de cualquier punto de la imagen, como puede verse en la siguiente ecuación:

$$\begin{cases} x = \frac{f}{Z}X + c_x \\ y = \frac{f}{Z}Y + c_y \\ d = \frac{f}{Z}B \end{cases} \quad (8.27)$$

Se define la matriz Q como en la Ecuación 8.28, donde B es el módulo del vector traslación entre las cámaras cuando éstas están rectificadas.

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & 1/B & 0 \end{bmatrix} \quad (8.28)$$

Con esta definición de Q , donde el último término de la diagonal es cero en el caso de que los rayos principales de las cámaras se intersecten en el infinito [24], el problema de triangular queda de la siguiente manera:

⁴Extracción de la Tesis de Maestría de Federico Lecumberry; para más información, consultar el Capítulo 4 de esta Tesis [40]

$$Q \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \quad (8.29)$$

De esta forma un punto en el espacio (3D) tendrá coordenadas $(X/W, Y/W, Z/W)$. Esta matriz es de fundamental importancia dado que contiene toda la información necesaria para realizar la triangulación en el caso de que las imágenes estén rectificadas y permite resolver el problema de forma lineal.

8.10.1. Disparidad y resolución de profundidad

Dada la mínima diferencia de disparidad detectada por los algoritmos, se puede determinar la resolución que se alcanza en la profundidad [53, 9], es decir, el mínimo salto en la dirección perpendicular al par estéreo (rectificado) que es posible distinguir.

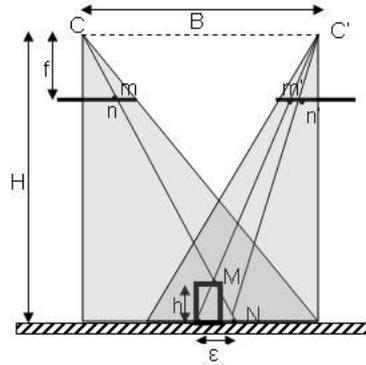


Figura 8.11: Relación entre profundidad y disparidad, cortesía de Javier Preciozzi.

La Figura 8.11 muestra un esquema de un par estéreo ya rectificado. Dado un punto M se pueden obtener las proyecciones m y m' de éste en cada una de las cámaras. Se considera además un plano auxiliar π colocado a una distancia H del centro de las cámaras. A la distancia entre el punto M y el plano π se le llama h y ϵ a la distancia entre las intersecciones de los haces de proyección de M con el plano π . Utilizando semejanza de triángulos se obtiene que:

$$\frac{B}{Z} = \frac{\epsilon}{h} \quad (8.30)$$

$$\implies h = \frac{Z}{B} \epsilon = \frac{H - h}{B} \epsilon \quad (8.31)$$

donde $Z = H - h$ es la profundidad del punto M . Si se llama d a la disparidad calculada en la imagen: $d = x_{m'} - x_m$, se puede relacionar utilizando semejanza de triángulos, la disparidad en la imagen con la disparidad en el plano auxiliar π :

$$\frac{f}{d} = \frac{H}{\epsilon} \implies \epsilon = \frac{H}{f}d \quad (8.32)$$

Combinando las Ecuaciones 8.31 y 8.32 se obtiene la expresión que relaciona la disparidad en la imagen con la profundidad:

$$d = \frac{hBf}{(H-h)H} \quad (8.33)$$

Para saltos pequeños de profundidad se puede suponer que h es mucho más pequeño que H , por lo que se tiene que:

$$\frac{h}{(H-h)} \simeq \frac{h}{H} \implies d = \frac{Bf}{H^2}h \quad (8.34)$$

De la Ecuación 8.34 se desprende que para lograr una buena resolución en profundidad es deseable que el largo focal f y la distancia entre las cámaras B sean grandes. Se puede incrementar el valor de f aumentando la resolución de las cámaras y/o el aumento. La precisión que se puede alcanzar al reconstruir un objeto disminuye cuadráticamente con la distancia de éste a la cámara.

Para una de las pruebas realizadas se obtuvo una resolución de 3,05 mm; los datos utilizados se ven a continuación:

- Traslación entre cámaras, $T = 15,39mm$
- Distancia focal, $f = 4312,7$
- Profundidad al objeto, $H = 450$

8.11. Algoritmos utilizados para búsqueda de correspondientes

En esta sección se muestra el funcionamiento de dos algoritmos para búsqueda de puntos correspondientes: Block Matching y Graph Cut. En el caso de Block Matching se realiza una descripción de su funcionamiento; también se muestran los resultados al aplicar los algoritmos a imágenes de prueba. Las imágenes de prueba utilizadas fueron extraídas de [12]. Estas imágenes cuentan con tres grandes virtudes: la primera es que ya están rectificadas, la segunda es que cumplen perfectamente con la restricción de similitud y la tercera fundamental es que se tienen los mapas de disparidad de ambas imágenes (“Ground Truth”).

Block Matching

La librería OpenCV implementa un algoritmo similar al desarrollado por Kurt Konolige [39]. Se trata de un algoritmo de tipo Block Matching que utiliza como medida de similitud de los vecindarios la Suma Absoluta de Diferencias (SAD).

Aspectos generales de este algoritmo son por ejemplo su rapidez, dado que busca correspondencias en un solo sentido y una sola vez. Por otro lado, al calcular el SAD para cada ventana y dado que la ventana no puede ser demasiado pequeña, como se verá en lo que sigue, este algoritmo presenta problemas en los cambios bruscos de profundidad; además no toma ninguna medida para evitar las oclusiones, hecho que también justifica su rapidez. Este algoritmo se puede escindir en tres pasos.

- Prefiltrado.

Se utiliza para normalizar el brillo de la imagen y resaltar la textura. El pixel central de la ventana de filtrado es sustituido por la expresión que se muestra en la Ecuación 8.35 donde \bar{I} es el promedio de intensidades de la ventana e I_{cap} es un parámetro del filtrado.

$$\min [max (I_{cap} - \bar{I}), -I_{cap}], I_{cap}] \quad (8.35)$$

- Búsqueda de correspondientes en las líneas epipolares.

Dado un punto en la imagen izquierda su correspondiente va a tener misma coordenada en y y va a tener igual o menor coordenada en x ($x_L > x_R$) por lo que la disparidad $d = x_L - x_R$ es positiva. Hay dos parámetros que controlan la búsqueda de estos correspondientes: la disparidad mínima y el número máximo de disparidades. La elección de ambos depende fuertemente de las imágenes que se tenga, o en todo caso de las profundidades en la escena que se desee reconstruir. Para calcular la disparidad mínima se debe observar la disparidad en píxeles que existe para un punto del espacio lejano a las cámaras. El número máximo de disparidades indica de cuántos píxeles estará conformado el intervalo de búsqueda de correspondientes, por lo que se determina calculando la disparidad de un punto cercano a las cámaras y restándole luego la disparidad mínima. Se observa que al fijar estos parámetros, indirectamente se está fijando la profundidad de la escena que se está cubriendo. Por otro, lado el volumen que se cubre de la escena aumenta ya sea mediante una disminución en la distancia entre las cámaras, una disminución en el largo focal o un aumento del rango de búsqueda.

- Postfiltrado.

Se realiza para eliminar falsas correspondencias. El máximo de similitud entre dos vecindarios, máximo de la función de correspondencia, se da cuando el vecindario está centrado en el punto correspondiente. A medida que el centro del vecindario se aleja de este punto, es de esperar que el valor de la función de correspondencia disminuya. Podría suceder por ejemplo que por ser una imagen de poca textura o porque existen vecindarios similares, el valor de máxima correspondencia no sea muy superior al valor de dicha función en otros puntos. Esto podría ser un motivo de falsa correspondencia. Para evitar estas situaciones se establecen dos umbrales: tener un mínimo valor de textura, y exigir una mínima diferencia entre los valores máximo y mínimo de la

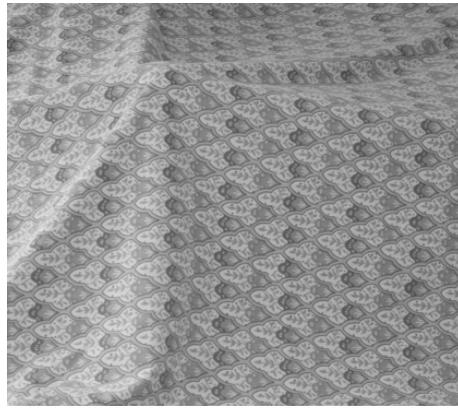
función de correspondencia. El propósito de este postfiltrado es determinar que el punto correspondiente que se elige del intervalo determinado por la mínima y máxima disparidad, se destaca lo suficiente del resto de los posibles puntos.

$$TasaDeUnicidad > \frac{(MaxValCorresp - MinValCorresp)}{MinValCorresp} \quad (8.36)$$

Prefiltrado Block Matching

Durante gran parte del tiempo en que se desarrollaron los conceptos de reconstrucción estéreo, en este proyecto se contaba con un dispositivo formado por un par de cámaras analógicas distintas. A la hora de buscar las correspondencias, es conveniente que las imágenes tengan texturas destacadas y brillo normalizado. Por este motivo fue necesario investigar el funcionamiento del prefiltrado, dado que de esa manera se puede atenuar la falta de similitud en las imágenes.

Los parámetros de este filtro son dos: el tamaño de la ventana de prefiltrado y el nivel de filtrado (parámetro I_{cap} que se menciona en la Ecuación 8.35). En la Figura 8.12 se muestra el resultado de filtrar las imágenes para distintos valores de los parámetros. Se observa que mientras más grande sea la ventana de filtrado, mayor es el efecto que ésta causa sobre los bordes de los detalles. Por otro lado, cuando se aumenta I_{cap} se consigue mayor detalle, pero hay que tener en cuenta que en el caso de que las imágenes sean distintas (como en el caso de las cámaras analógicas) también se estarán reasaltando sus diferencias.



(a) Imagen original

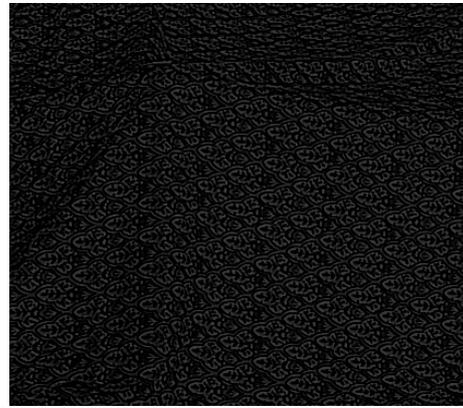
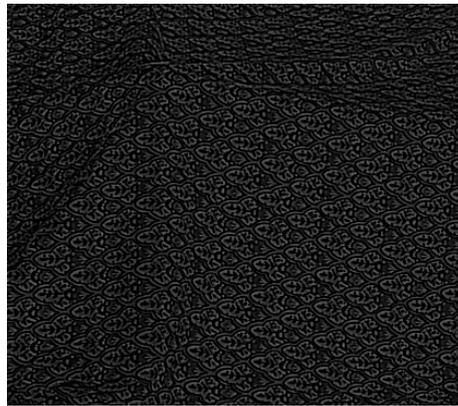
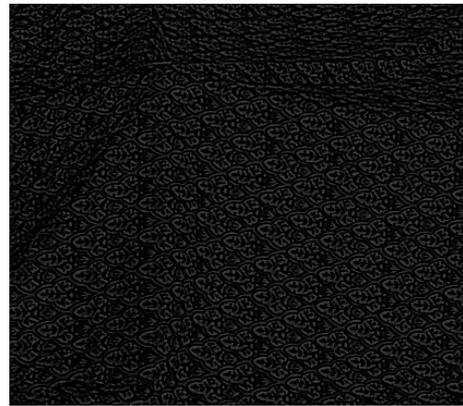
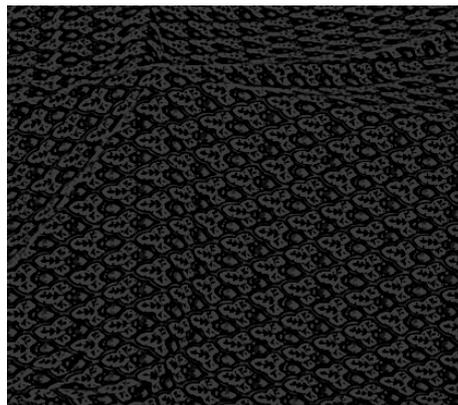
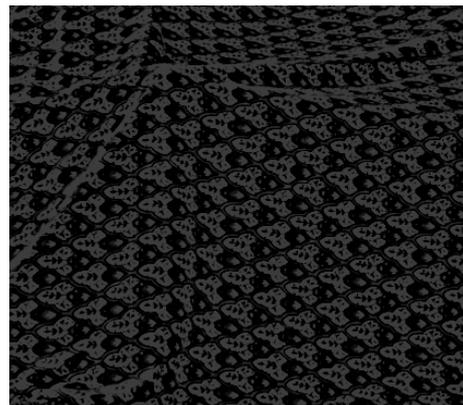
(b) Imagen filtrada; ventana de 5x5, $I_{cap} = 15$ (c) Imagen filtrada; ventana de 5x5, $I_{cap} = 30$ (d) Imagen filtrada; ventana de 5x5, $I_{cap} = 45$ (e) Imagen filtrada; ventana de 11x11, $I_{cap} = 30$ (f) Imagen filtrada; ventana de 21x21, $I_{cap} = 30$

Figura 8.12: Resultados del prefiltrado que realiza el algoritmo Block Matching variando sus parámetros

Resultados del algoritmo Block Matching

En esta sección se analizan los resultados de aplicar el algoritmo Block Matching a imágenes extraídas de [12]. En particular, se pondrá atención a cómo responde el algoritmo frente a cambios en el tamaño del vecindario.

Las Figuras 8.13 y 8.14 muestran dos conjuntos de imágenes sobre los que se ha aplicado el algoritmo para diferentes tamaños de vecindario. 8.13(a) y 8.14(a) son las imágenes originales (una de las imágenes del par); 8.13(b) y 8.14(b) son los mapas de disparidad verdaderos. Una primera inspección de los mapas de disparidad verdaderos permite ver que en 8.14(b) existen zonas negras, a diferencia de 8.13(b), donde no las hay. Esto es consecuencia de la presencia de oclusiones de la escena, lo que genera discontinuidades en el mapa de disparidad. 8.13(c) y 8.14(c) son los mapas de disparidad resultantes al considerar un vecindario de 15×15 píxeles, mientras que para los mapas de 8.13(d) y 8.14(d) se ha utilizado un vecindario de 5×5 píxeles.

De las imágenes de la Figura 8.12 se desprende que un vecindario pequeño, de 5×5 píxeles, funciona mejor en las zonas más texturadas: si la superficie no presenta características de tamaño comparable al de la ventana, se tendrá una región en la que el valor de la función de correspondencia tomará un valor uniforme, por lo que el postfiltrado no permitirá determinar la disparidad. Por el contrario, cuanto más grande sea el vecindario menos se dependerá de la textura de las imágenes para calcular las correspondencias. Como contrapartida, al tener un vecindario más grande se tiene una mayor zona de transición en las regiones de discontinuidad, resultando en que en las mismas aumente la zona en que no se puede determinar la disparidad. Además, existen más posibilidades de encontrar “falsas correspondencias”, lo que da una sensación de suavizado del mapa de disparidad.

En las imágenes de la Figura 8.13 se aprecia una imagen original, el mapa de disparidad Ground Truth, y dos mapas de disparidad que se obtuvieron para dos tamaños de ventana distintos. Si se toma una ventana muy pequeña, para obtener un correcto mapa de disparidad la imagen debe ser muy texturada, de lo contrario existirán zonas en las cuales el postfiltrado no permitirá estimar la disparidad. Por otro lado mientras más grande sea la ventana de búsqueda, menos se depende de la textura de las imágenes para calcular las correspondencias. Como contrapartida, al tener una ventana más grande se tiene una mayor zona de transición en las regiones de discontinuidad, resultando en que en las mismas aumente la zona en que no se puede determinar la disparidad. Además, como el vecindario es más grande, existen más posibilidades de encontrar “falsas correspondencias” que resultan en una sensación de suavizado del mapa de disparidad. Estos efectos se pueden apreciar en las imágenes 8.14(c) y 8.14(d), en la primera se observa que por ejemplo el pie derecho del bebe se confunde con el cilindro en el que está sentado; por otro lado tanto en la imagen 8.14(d) y 8.13(d) hay muchos puntos en los que no se pudo estimar la disparidad dado que la ventana era muy pequeña.

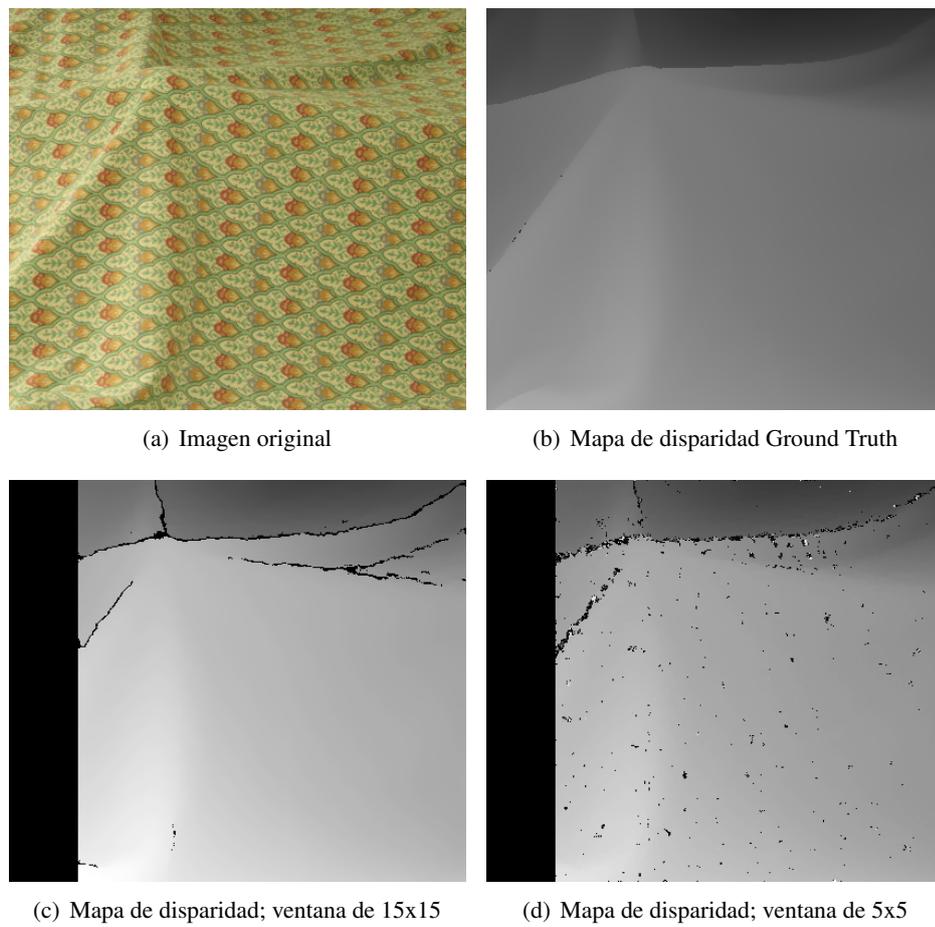


Figura 8.13: Resultados del algoritmo Block Matching con distintos tamaños de ventanas.

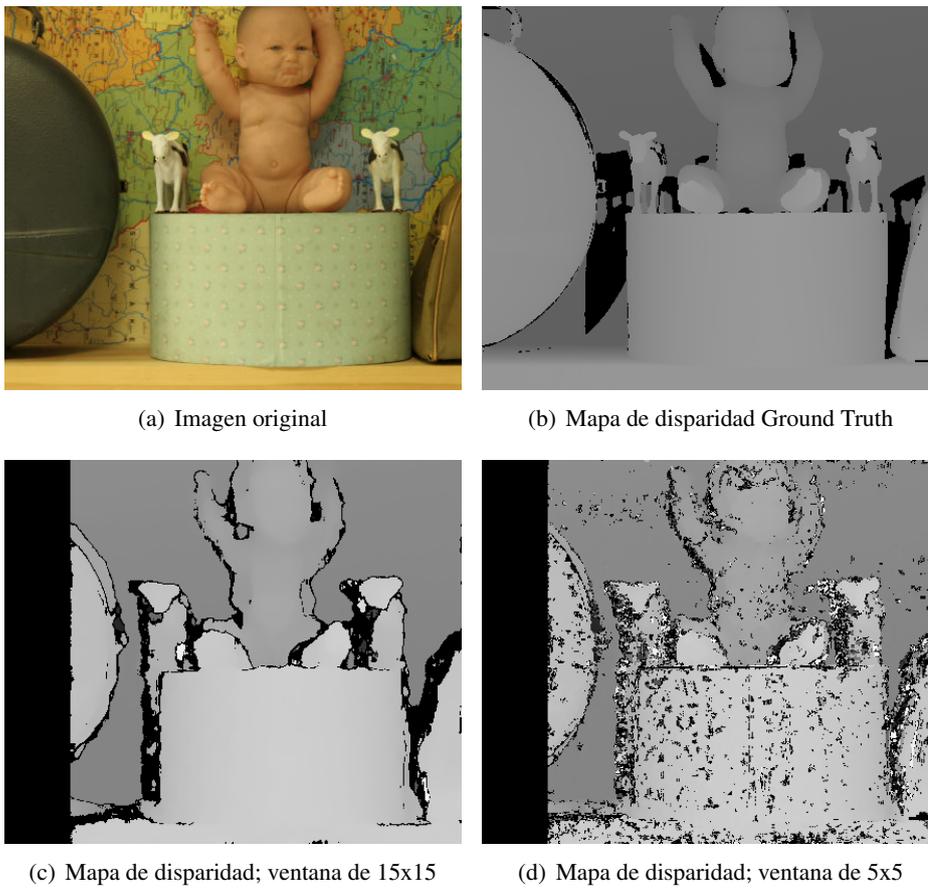


Figura 8.14: Resultados del algoritmo Block Matching con distintos tamaños de ventanas.

Graph Cut - comparación con Block Matching

OpenCV implementa el algoritmo de búsqueda de correspondientes de Kolmogorov [38], que es un algoritmo que se basa en corte de grafos. Este tipo de algoritmos como ya se mencionó, minimizan una cierta expresión de energía. La gran diferencia que aparece entre este tipo de algoritmos y Block Matching es que este último sólo busca correspondencia en la misma scanline mientras que los primeros buscan correspondencias en todas las scanlines simultáneamente, es decir que se busca la correspondencia de una superficie de mínimo costo. Ajustar una superficie minimizando alguna energía, presenta una coherencia local en todas las direcciones, en particular la horizontal (intra-scanline) y la vertical (inter-scanline), por la forma en que se construye la solución, y no se fuerza esta coherencia mediante restricciones en la minimización⁵. Otra ventaja de los algoritmos que funcionan por corte de grafos es que implementan una forma (agregando un término de energía) de evitar las oclusiones, mientras que Block Matching no. Las diferencias antes mencionadas hacen que el costo computacional entre ellos sea muy distinto siendo extremadamente elevado para Graph Cut, que en imágenes de 437×370 píxeles requiere en el orden del minuto, mientras que Block Matching requiere en el orden del segundo (estos tiempos son tomados con la misma PC).

Un punto que es importante destacar, es que Graph Cut no implementa disparidad sub-píxel mientras que Block Matching sí, esto hace que las superficies reconstruidas a partir del primer algoritmo presenten saltos discretos en la profundidad. La amplitud de los saltos queda determinada por la resolución del sistema y la distancia a la cámara de la superficie. Esta disparidad discreta dificulta el proceso de reconstrucción de las mallas ya que el algoritmo de reconstrucción tiende a unir puntos de igual profundidad. Por otro lado, los mapas de disparidad conseguidos con Block Matching son más suaves dado que este implementa un cálculo de la disparidad subpíxel $\times 16$, esto resulta en superficies reconstruidas más suaves.

Las diferencias antes mencionadas se aprecian en las imágenes de la Figura 8.15 que son los mapas de disparidad obtenidos con Graph Cut para las mismas imágenes de pruebas utilizadas en las Figuras 8.13 y 8.14. Se observa que este algoritmo no presenta grandes problemas en los cambios abruptos de profundidad y que intenta (con mejores resultados) resolver el problemas de las oclusiones (Figura 8.15(b) vs. Figura 8.14(c)). Por otro lado, si se compara la Figura 8.13(c) con la Figura 8.15(a) se aprecia que la primera presenta un mapa de disparidad visualmente continuo, mientras que la segunda muestra saltos discreto mayores.

⁵Extracción de la Tesis de Maestría de Federico Lecumberry, para más información revisar el capítulo 3 de [40]

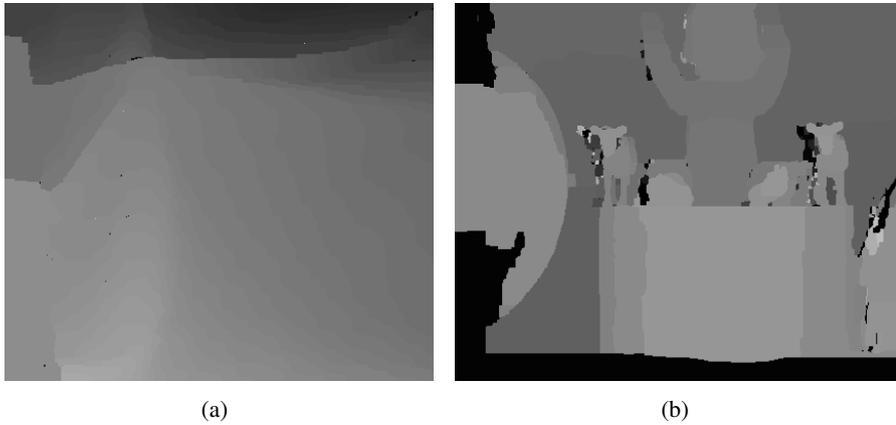
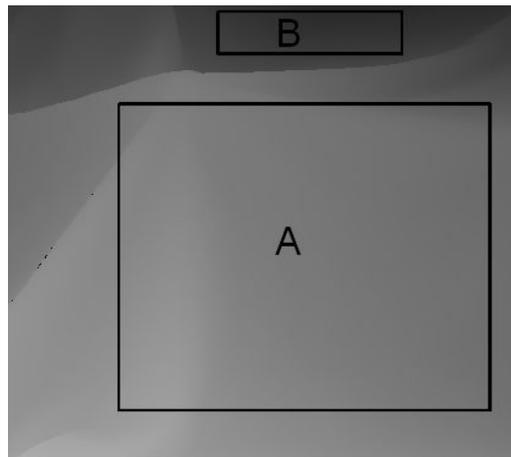


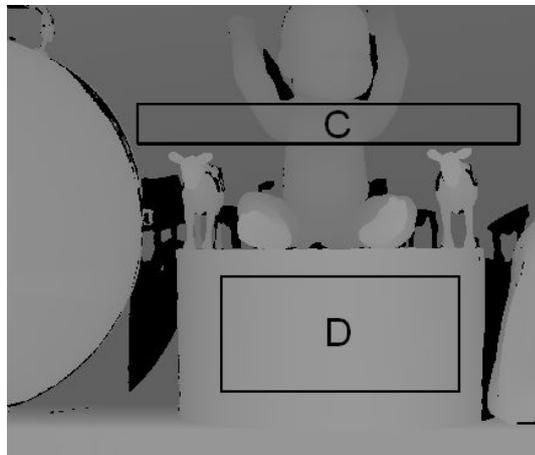
Figura 8.15: Mapas de disparidad obtenidos con Graph Cut

Análisis cuantitativo de los algoritmos

A continuación se muestran resultados cuantitativos de aplicarle los algoritmos de cálculo de disparidad a las imágenes de prueba. Se analiza la cantidad de puntos para los que no se pudo calcular la disparidad y también el error RMS entre el mapa de disparidad calculado con los algoritmos y el Ground Truth. Para las imágenes de ejemplo cuyo Ground Truth se muestra en las imágenes de la Figura 8.16 se corrieron los algoritmos de Block Matching y Graph Cut, el primero utilizando dos tamaños de ventana de SAD (5×5 y 15×15). En las imágenes se muestran las cuatro regiones en las que se compararon los algoritmos y en la Figura 8.17 se muestran las regiones extraídas de los mapas de disparidad.

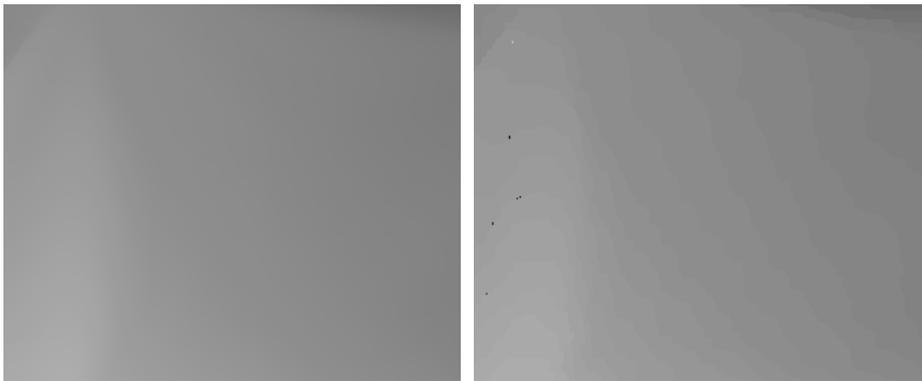


(a)



(b)

Figura 8.16: Ground Truth y regiones de las imágenes sobre las que se analizan los errores



(a) Región A del Ground Truth de la imagen 8.16(a). (b) Región A del mapa de disparidad utilizando Graph Cut.



(c) Región A del mapa de disparidad utilizando Block Matching con ventana de 5x5. (d) Región A del mapa de disparidad utilizando Block Matching con ventana de 15x15.

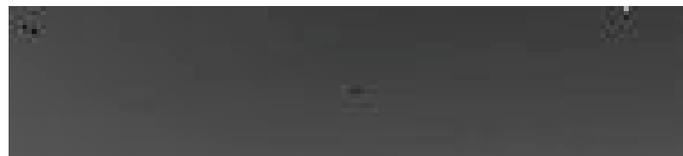
Figura 8.17: Regiones de los mapas de disparidad para la región A mostrada en la imagen 8.16(a)



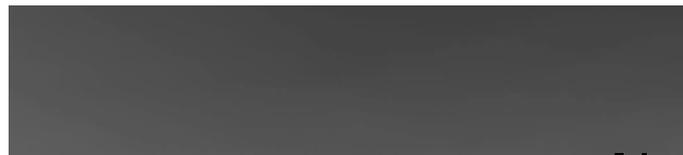
(a) Región B del Ground Truth de la imagen 8.16(a).



(b) Región B del mapa de disparidad utilizando Graph Cut.



(c) Región B del mapa de disparidad utilizando Block Matching con ventana de 5x5.



(d) Región B del mapa de disparidad utilizando Block Matching con ventana de 15x15.

Figura 8.18: Regiones de los mapas de disparidad para la región 1 mostrada en la imagen 8.16(a)

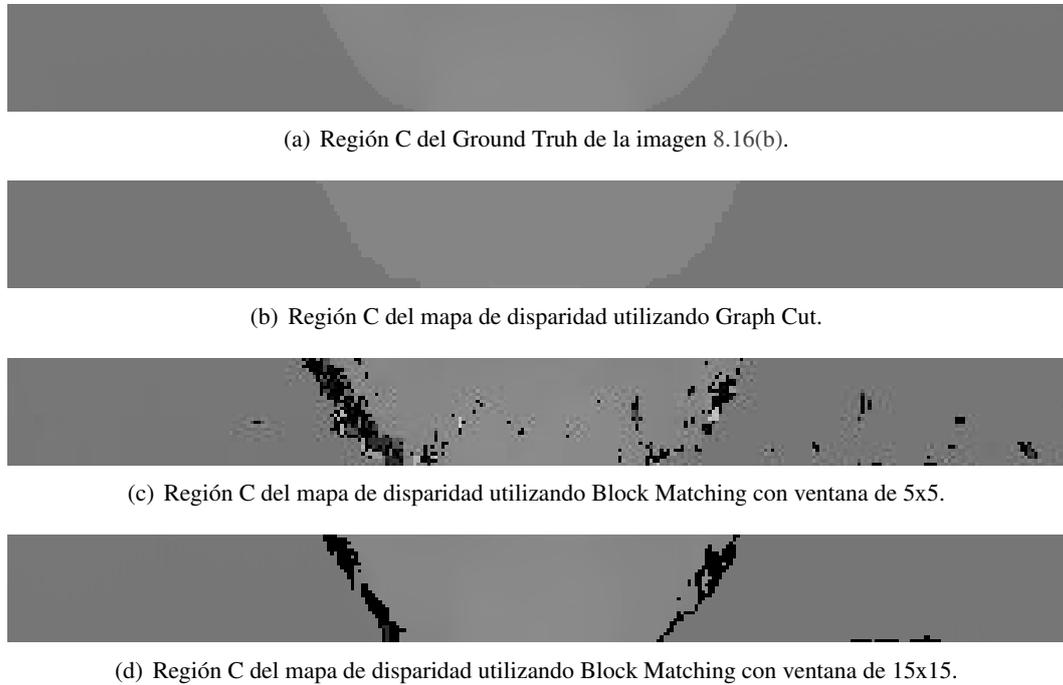


Figura 8.19: Mapas de disparidad de la región C mostrada en la imagen 8.16(b)

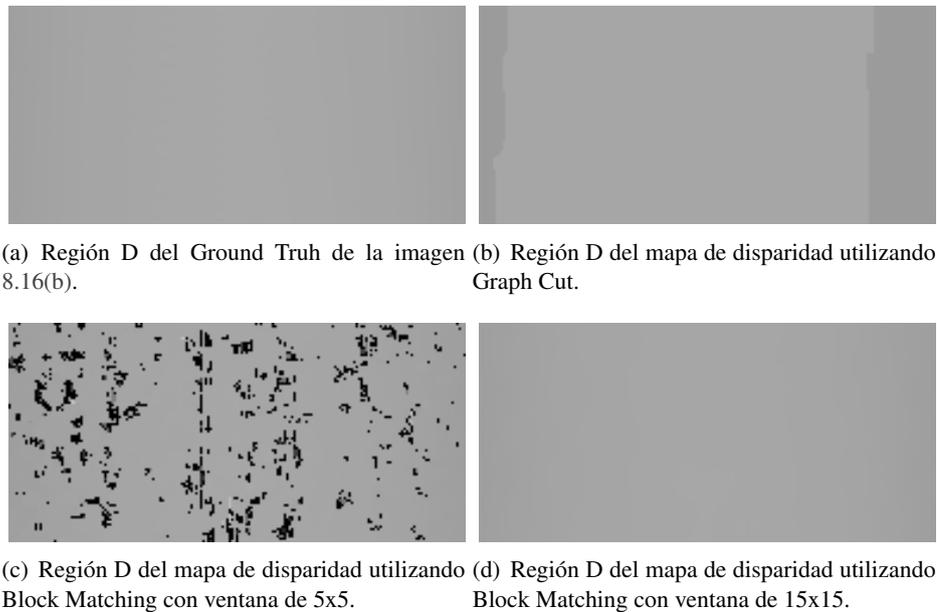


Figura 8.20: Mapas de disparidad de la región C mostrada en la imagen 8.16(b)

En las Tablas 8.1 y 8.2 se observan los resultados de comparar los mapas de disparidad en las distintas zonas. Como ya se comentó, utilizar una ventana de búsqueda más pequeña con Block Matching, puede traer problemas a la hora de encontrar puntos correspondientes dado que se tiene menos información de la textura. Con menos información de textura puede pasar que en el intervalo de búsqueda se encuentren varios puntos candidatos a correspondientes, en ese caso el algoritmo Block Matching decide no asignar punto correspondiente. Esto coincide con los resultados dado que, el error RMS entre los mapas de disparidad y el Ground Truth siempre fue mayor en el caso de utilizar ventanas de búsqueda de 5x5 que cuando se utilizaron ventanas de 15x15. Este resultado también coincide con la cantidad de puntos en los que no se puede calcular la disparidad.

En cuanto al algoritmo Graph Cut se observa, como era de esperarse, que son muy pocos los puntos en los cuales no puede calcular la disparidad, comparando con los resultados del algoritmo Block Matching. Por otro lado, un resultado que no es menor es que si bien se encontró la disparidad de la gran mayoría de los puntos de las regiones, el error RMS no dio tan buenos resultados. En particular, para las regiones *B* y *D* Block Matching con ventanas de 15x15 da mejores resultados. Si se observa en las imágenes los mapas de disparidad obtenidos en estas regiones utilizando Graph Cut, se verá que hay saltos de disparidad discretos más grandes que los que se dan con Block Matching (visualmente continuo). Aquí está la gran ventaja de utilizar el algoritmo Block Matching respecto de Graph Cut, el primero da como resultado un mapa de disparidad con resolución subpíxelica x16, mientras que Graph Cut no. Este efecto de saltos discretos grandes en los mapas de disparidad hace que al reconstruir, la escena quede con discontinuidades en la profundidad; de esta manera los algoritmos que generan las mallas tienden a unir planos a profundidades constantes en vez de la superficie.

	Región A (249x299)	Región B (32x147)	Región C (30x306)	Región D (91x190)
Block Matching 5x5	24.75	4.81	64.33	64.66
Block Matching 15x15	15.28	4.81	52.20	16.72
Graph Cut	12.84	12.81	32.44	42.26

Tabla 8.1: Error RMS.

	Región A (249x299)	Región B (32x147)	Región C (30x306)	Región D (91x190)
Block Matching 5x5	488	3	431	1325
Block Matching 15x15	229	3	294	0
Graph Cut	7	0	0	0

Tabla 8.2: Número de puntos sin cálculo de disparidad.

'10

Capítulo 9

Implementación Estéreo

9.1. Introducción

Este capítulo detalla los distintos sistemas de adquisición estéreo con los que se realizaron las pruebas. Se muestran los distintos dispositivos utilizados en el laboratorio y en el Hospital de Tacuarembó, así como también los resultados obtenidos. Por otro lado se observan resultados obtenidos con distintos métodos utilizados en la construcción de mallas, teniendo como punto de partida los puntos en el espacio.

9.2. Descripción de dispositivos estéreo

En el transcurso de este proyecto se utilizaron diferentes dispositivos para obtener imágenes de un par estéreo.

- Par estéreo con cámaras web.
- Par estéreo con cámaras analógicas *distintas*.
- Par estéreo del Microscopio del Hospital de Tacuarembó.
- Una sola cámara + Posicionador.

9.2.1. Par estéreo con cámaras web

Con esta implementación se dieron los primeros pasos en los temas de reconstrucción estéreo. El dispositivo se compone de dos cámaras web convencionales, que pueden adquirir imágenes de hasta 352×288 píxeles de resolución. Las mismas están colocadas en un soporte que permite la rotación individual y que a su vez mantiene una traslación fija entre ellas. El dispositivo se puede apreciar en la Figura 9.1 mientras que en la Figura 9.2 se observan algunas de las imágenes adquiridas.



Figura 9.1: Par estéreo con cámaras web.



Figura 9.2: Ejemplos de imágenes capturadas con el par estereo de cámaras web.

Con los primeros avances en los temas de reconstrucción estereo se pudieron introducir los conceptos de calibración estereo y rectificación, obteniendo resultados razonables para los equipos mencionados (cámaras web de calidad estándar). En cuanto a conseguir mapas de disparidad que permitieran realizar triangulación, este dispositivo fallaba por varios motivos. El primero y fundamental, la mala calidad de

las imágenes que hace que se pierdan texturas y dificulta a los algoritmos de búsqueda de puntos correspondientes. Por otro lado, las imágenes eran muy pequeñas y con muy baja resolución (esto provoca una distancia focal f pequeña) por lo que no se obtuvo la resolución necesaria. Un ejemplo del proceso de rectificación y del cálculo del mapa de disparidad se observa en la Figura 9.3. En el mapa de disparidad se ven distintas profundidades, indicadas por regiones de distintas tonalidades (se pueden apreciar al menos cuatro profundidades bien diferenciadas).



Figura 9.3: Rectificación de un par estéreo de cámaras web y mapa de disparidad.

9.2.2. Par estéreo con cámaras analógicas *distintas*

En procura de mejorar los mapas de disparidad, dado que no prosperó el par estéreo de cámaras web, se pasó a implementar un par estéreo con dos cámaras analógicas de mejor calidad; esta implementación se ve en la Figura 9.4. Las cámaras analógicas utilizadas ya fueron especificadas en la Subsección 3.3.1. El tipo de imágenes que se adquirieron con este par estéreo se puede apreciar en la Figura 9.5.



Figura 9.4: Par estéreo con cámaras analógicas distintas.

Viendo las imágenes de la Figura 9.5 se aprecia que las mismas tienen características distintas en cuanto a brillo y calidad. Esto se debe a que los CCD de las cámaras son distintos y a que también las tarjetas capturadoras son distintas (en la Sección 9.3 se hará fuerte hincapié en los inconvenientes que esto trae). Los resultados de rectificación también fueron satisfactorios, al igual que con las cámaras web, y se mejoraron los mapas de disparidad. De todas maneras el sistema no pudo conseguir la resolución requerida, dado que los saltos en profundidad siguieron siendo demasiado grandes.

Un ejemplo de los resultados obtenidos con este sistema de adquisición se puede apreciar en la Figura 9.6. Aquí se observan claramente los planos de la caja y la chapa, pero no se aprecia la inclinación que éstas presentan, por ejemplo en la cara inferior de la caja. Esto hace que el sistema siga siendo inadecuado para la reconstrucción de las superficies deseadas. Este problema de resolución se puede apreciar claramente en la Figura 9.7 donde por ejemplo no se alcanza a distinguir la mano, sino que se mezcla todo en un mismo plano.



Figura 9.5: Ejemplos de imágenes capturadas con el par estero de cámaras analógicas distintas.



Figura 9.6: Resultados utilizando cámaras analógicas: imágenes rectificadas - mapa de disparidad.

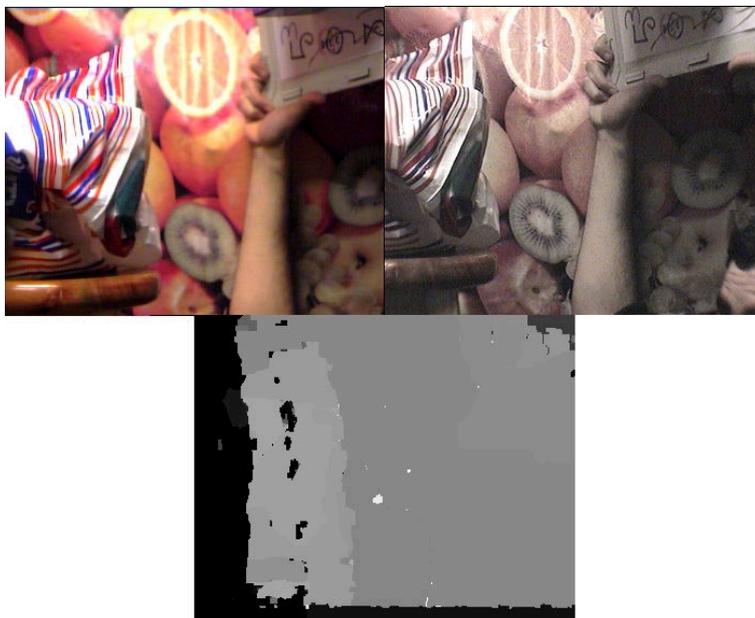


Figura 9.7: Resultados utilizando cámaras analógicas: imágenes rectificadas - mapa de disparidad. Baja resolución del mapa de disparidad.

9.2.3. Par estéreo del Microscopio del Hospital de Tacuarembó

En la mayoría de las intervenciones quirúrgicas en las que se utiliza el microscopio, ambos oculares auxiliares son utilizados. Uno de ellos es utilizado por el ayudante del cirujano, mientras que en el otro se coloca una cámara conectada a un televisor. Utilizando este televisor el personal presente en el block puede ver lo mismo que observa el cirujano y de esta forma estar al tanto del estado de la intervención.

Puntualmente, se pudo contar con un par de cámaras para colocar en los oculares auxiliares del microscopio del Hospital de Tacuarembó, las mismas se describieron junto con sus adaptadores en la Sección 3.2.2. Estas cámaras capturan las imágenes provenientes de ambos oculares del microscopio; de esta manera se obtiene un par estéreo. En la Figura 9.8 se puede ver el par de cámaras.

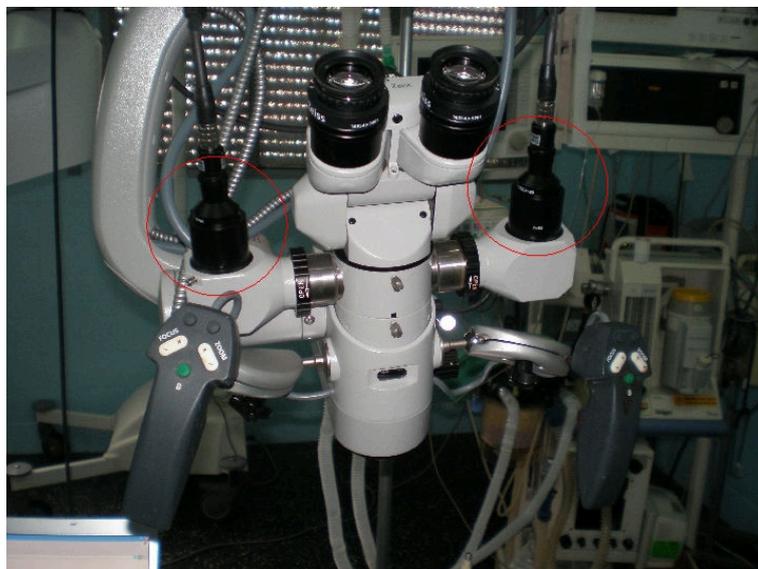


Figura 9.8: Cámaras presentes en el microscopio del Hospital de Tacuarembó.

Al no contar en el laboratorio con un par estéreo de cámaras que permitiera realizar de forma completa el proceso de reconstrucción, se decidió utilizar una sola cámara y el localizador Polaris para realizar dicho proceso. Dado que la posibilidad de utilizar el par estéreo del microscopio surgió en el tramo final del proyecto, cuando ya se había decidido utilizar una sola cámara, no se utilizó este dispositivo. Una ventaja extra, además de poder realizar pruebas en el laboratorio, es que realizar estéreo con una sola cámara no interfiere con el funcionamiento normal de las intervenciones quirúrgicas, dado que permite que el ayudante pueda utilizar el ocular libre. Actualmente es muy importante que la reconstrucción estéreo pueda realizarse utilizando una sola cámara, dado que de lo contrario, los ayudantes del cirujano deberían cambiar su forma de trabajo.

La utilización del par de cámaras del microscopio queda como trabajo a futuro, para lo que será de vital importancia contar con dos tarjetas adquisidoras idénticas y

más tiempo para realizar pruebas con el microscopio.

9.2.4. Una sola cámara + localizador Polaris

Dados los problemas que se describieron anteriormente, se decidió implementar un dispositivo que los solucionara y con el que además fuera posible realizar pruebas en el laboratorio. El mismo está conformado por una sola cámara y el localizador Polaris. Para realizar la reconstrucción se deben tomar dos imágenes con la cámara y, utilizando el localizador, adquirir la posición de la cámara al tomar ambas imágenes. Con estas posiciones, calcular la posición relativa entre las dos posiciones de la cámara y utilizando esa información rectificar las imágenes. Luego de rectificadas, seguirá el proceso normal para realizar la reconstrucción.

Al utilizar una sola cámara se tiene la desventaja de que la escena debe permanecer estática mientras se adquieren las dos imágenes; ésta es una limitación del sistema. Por otro lado si se cuenta con escenas estáticas, todo lo que queda son ventajas. La primera y fundamental ventaja (principal motivación) es que al utilizar la misma cámara y la misma tarjeta adquisidora para capturar las dos imágenes, éstas tendrán las mismas características, facilitando así en gran medida la búsqueda de puntos correspondientes. Otra ventaja que se obtiene, es que el movimiento entre las dos imágenes (entre las “dos cámaras”) es fijado por el usuario al mover la cámara para tomar la segunda imagen. Con este dispositivo se obtuvieron muy buenos resultados; el análisis del funcionamiento y los resultados serán discutidos en la Sección 9.4.

9.3. Problemas con cámaras distintas

Dadas las dificultades que se presentaron al realizar la reconstrucción estéreo y la falta de un profundo conocimiento sobre los algoritmos de búsqueda de correspondientes, se decidió en primera instancia, analizar cuál era el efecto de contar con imágenes que no cumplieran con la restricción de similitud ¹(Sección 8.9). Para esto se analizaron los resultados de los algoritmos de búsqueda de correspondientes con imágenes de pruebas. Estas imágenes provienen de la base de Middlebury [12] y como ya se mencionó cuentan con tres características: la primera es que están perfectamente rectificadas, la segunda es que cumplen perfectamente con la restricción de similitud y la tercera es que el paquete de imágenes cuenta con lo que llaman “Ground truth”, que son los mapas de disparidad de ambas imágenes. En la Figura 9.9 se ve un ejemplo de las imágenes originales.

Para analizar cómo influye el hecho de adquirir las imágenes con cámaras distintas, se decidió evaluar el desempeño de los algoritmos Block Matching y Graph Cut al variar de a una por vez las siguientes propiedades:

¹En ese momento del proyecto, el mejor dispositivo de captura estéreo con que se contaba eran las dos cámaras analógicas distintas, por eso se realizó el esfuerzo de intentar resolver los problemas que aparecían por tener imágenes distintas

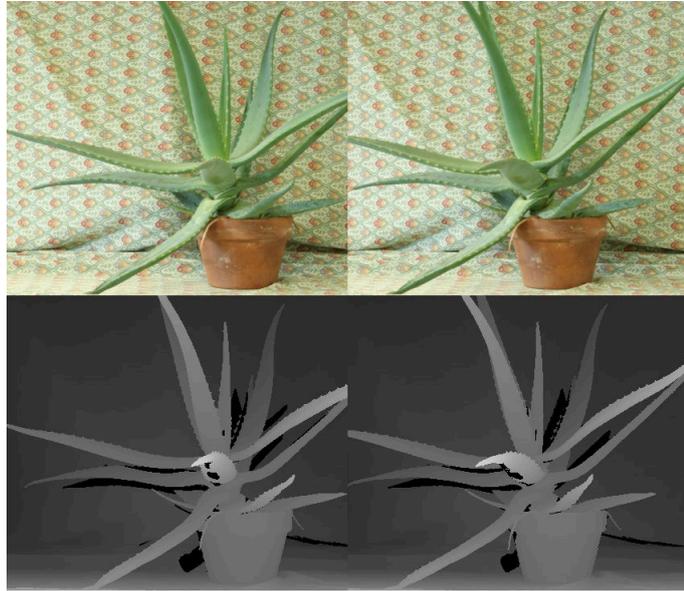


Figura 9.9: Imágenes que se utilizaron para realizar las pruebas a los algoritmos de búsqueda de correspondientes.

- Contraste en las imágenes.

Se cambió el contraste de una de las imágenes para verificar qué efectos causa en los algoritmos de búsqueda de correspondientes. Esto simula el hecho de que las imágenes captadas con cámaras y tarjetas adquisidoras distintas tienen características de contraste diferente. Las imágenes de prueba fueron las que se muestran en la Figura 9.10, donde se muestra la imagen izquierda original y la imagen derecha con el contraste modificado.

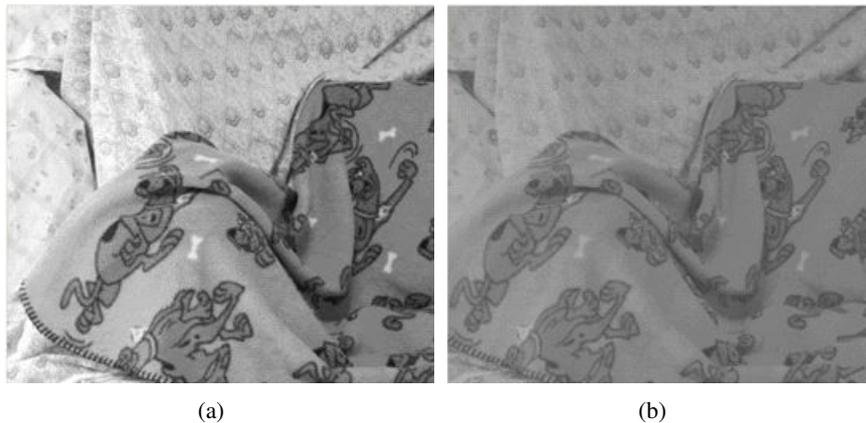
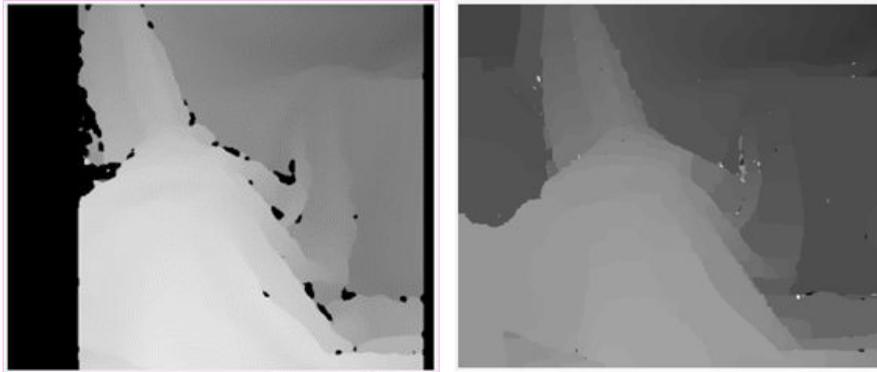


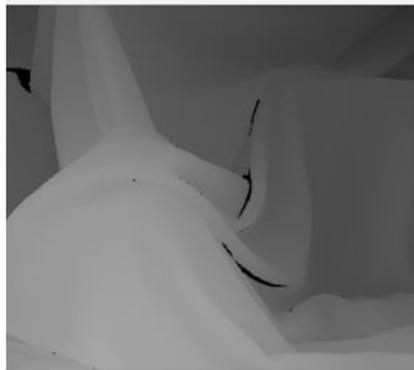
Figura 9.10: Cambio en el contraste de la imagen derecha.

Para estas imágenes los mapas de disparidad que resultan se pueden ver en



(a) Mapa de disparidad para las imágenes sin modificar con Block Matching

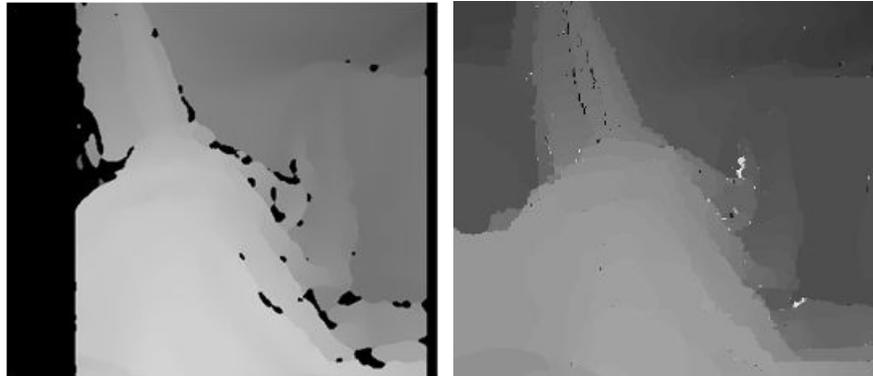
(b) Mapa de disparidad para las imágenes sin modificar con Graph Cut



(c) Mapa de disparidad real

Figura 9.11: Mapas de disparidad para las imágenes sin modificar.

la Figura 9.12. Como se aprecia al comparar el mapa de disparidad original (Figura 9.11) con los que se obtienen con las imágenes con distintos niveles de contraste, éstos comienzan a perder puntos correspondientes. En Block Matching se puede ver que aparecen más regiones negras en las zonas donde hay cambios de profundidad (por ejemplo en los pliegues de la ropa), mientras que en Graph Cut el mapa de disparidad comienza a ser más granulado.



(a) Mapa de disparidad utilizando Block Matching (b) Mapa de disparidad utilizando Graph Cut

Figura 9.12: Mapas de disparidad resultantes al variar el contraste de una imagen.

- Ruido en la imagen.

La siguiente prueba que se realizó fue agregarle ruido gaussiano a una de las imágenes, y de la misma manera que como cuando se cambió el contraste, observar los mapas de disparidad resultantes y compararlos con los originales. Se realizaron dos pruebas con distintos niveles de ruido que se pueden observar en las Figuras 9.13 y 9.14. Los resultados, como se observa en las figuras, son que cuanto más se rompe la restricción de similitud entre las imágenes, el algoritmo de Block Matching comienza a fallar cada vez más. Por otro lado, el algoritmo de Graph Cut es mucho más robusto a esos cambios en las imágenes, dando resultados buenos para ambos niveles de ruido.

- Error en la rectificación.

En este punto se analizó cómo incide en los algoritmos de búsqueda de puntos correspondientes el hecho de que las imágenes no estén correctamente rectificadas. Para esto se partió de las imágenes de prueba (ya rectificadas) y se desplazó a una de las imágenes en sentido vertical respecto de la otra. Esto es hacer que las líneas epipolares queden corridas por el desplazamiento aplicado. Para esta prueba bastó con realizar un desplazamiento entre las imágenes de hasta dos píxeles para poder apreciar los efectos. En la Figura 9.15(a) se aprecia el desplazamiento entre las imágenes (de forma exagerada), y los resultados para desplazamientos de uno y dos píxeles.

Como conclusión, mediante estos experimentos se pudo evaluar el funcionamiento de los algoritmos de búsqueda de puntos correspondientes. De los tres factores estudiados es notorio que errores en la rectificación resultan en mapas de disparidad pésimos, y por esto se buscó un dispositivo que lograra una correcta calibración estereo (que es la base para la rectificación). Por otro lado quedan otros dos problemas: imágenes distintas y resolución en la profundidad. Para evitar todos

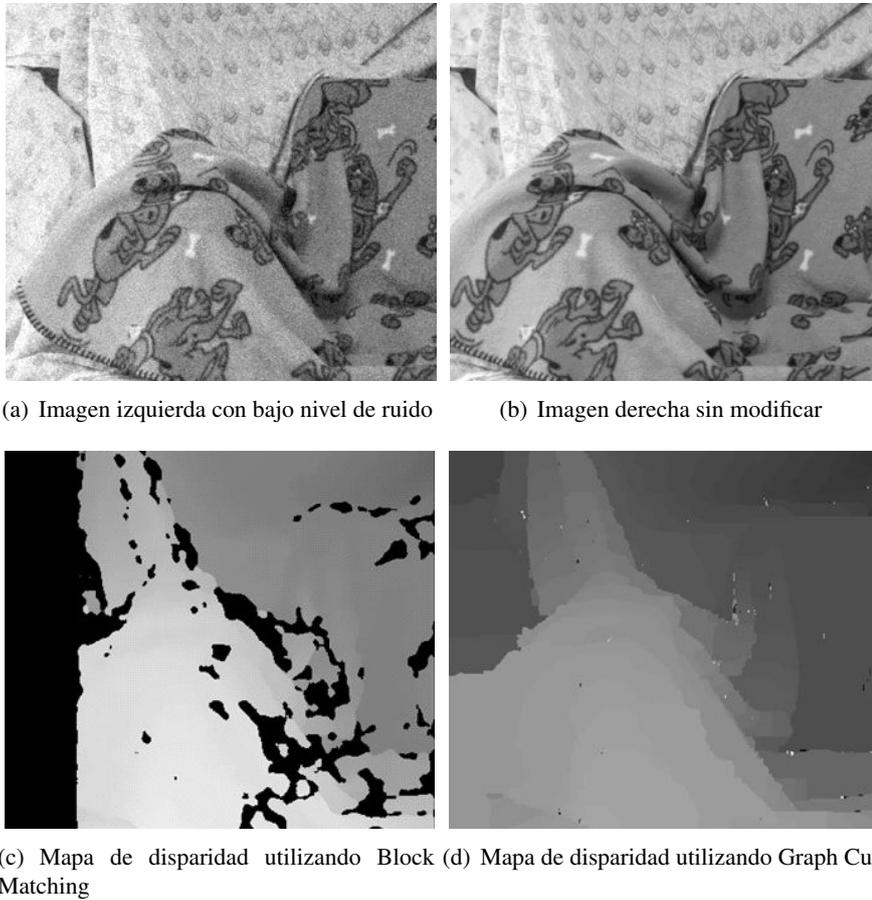


Figura 9.13: Mapas de disparidad utilizando como base las imágenes con ruido gaussiano.

estos inconvenientes se decidió utilizar un dispositivo formado por una sola cámara y el localizador Polaris.

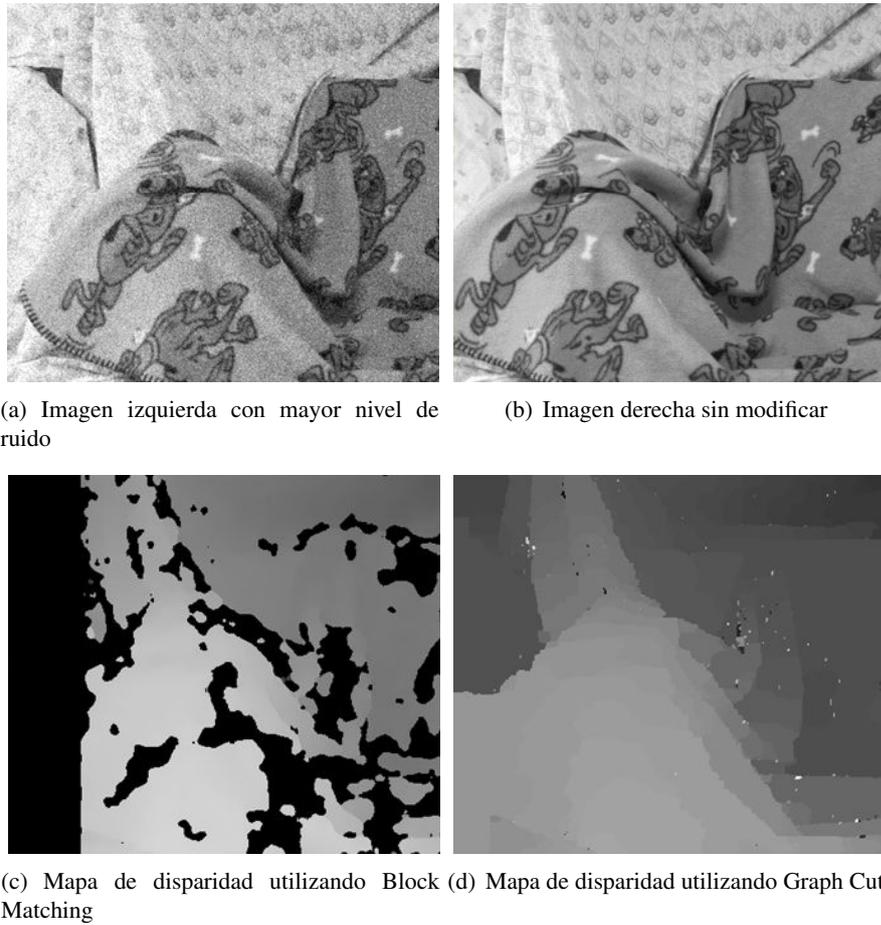
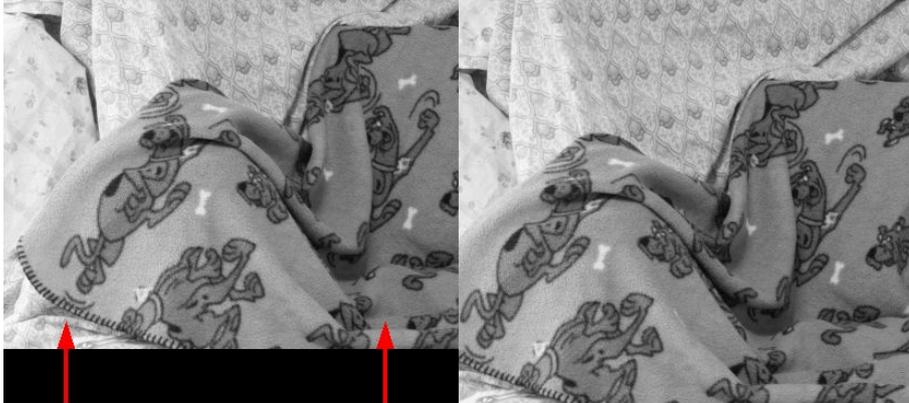
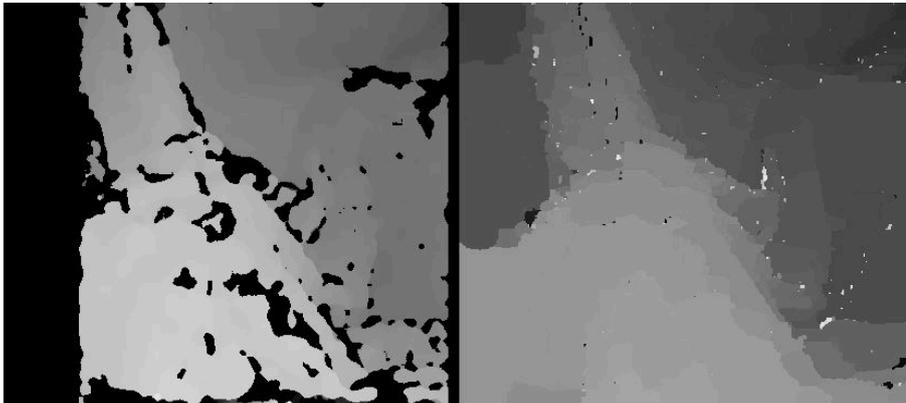


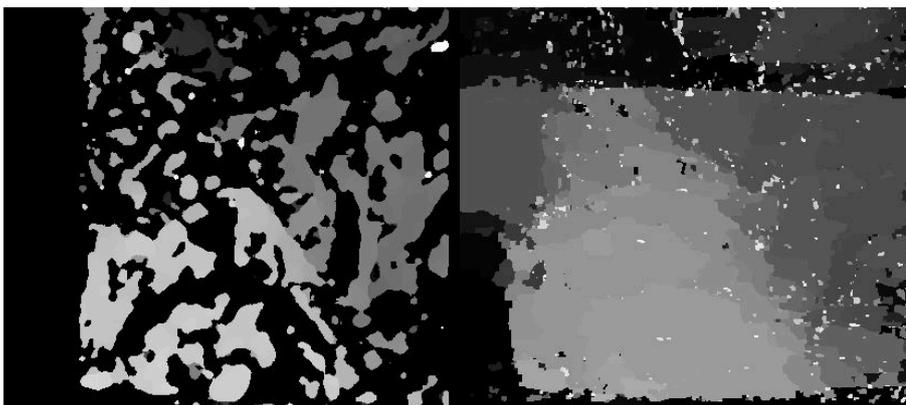
Figura 9.14: Mapas de disparidad utilizando como base las imágenes con ruido gaussiano.



(a) Ejemplo exagerado de corrimiento entre las imágenes para simular una mala rectificación



(b) Mapa de disparidad (izquierda BM y derecha GC) con un desplazamiento entre las imágenes de **un** pixel



(c) Mapa de disparidad izquierda BM y derecha GC) con un desplazamiento entre las imágenes de **dos** pixeles

Figura 9.15: Análisis del efecto de una mala rectificación en los mapas de disparidad.

9.4. Reconstrucción con una cámara + localizador Polaris

Debido a las dificultades que presenta la búsqueda de puntos correspondientes utilizando imágenes con características disímiles, se resolvió implementar un dispositivo que cuenta con una sola cámara. Este dispositivo tiene la gran ventaja de que al tomar ambas imágenes utilizando la misma cámara y tarjeta adquisidora, se cumple con la restricción de similitud. Otra ventaja es que se puede utilizar una cámara de fotos digital (en lugar de una cámara de video) y dadas las grandes resoluciones que estas manejan en la actualidad, es posible alcanzar la resolución de profundidad deseada.

La pregunta que surge es: ¿cómo es posible tener un par estéreo con una sola cámara? La respuesta es que sólo se puede realizar la reconstrucción si se conoce exactamente el movimiento de la cámara entre los instantes de captura de las imágenes. Es en este punto que entra en juego la participación del localizador Polaris, dado que si se adhieren a la cámara un conjunto de marcadores y se obtiene con el localizador las dos posiciones de la cámara, se puede encontrar el movimiento de la cámara entre ambos instantes.

En este momento parece importante precisar que el objetivo de la calibración estéreo es determinar la ubicación relativa entre las dos posiciones de la cámara, es decir la posición de la “cámara derecha” respecto de la “cámara izquierda”. Dado que la ubicación del centro de la cámara respecto a los marcadores adheridos a ella no varía, se podría pensar que dicha posición relativa se podría determinar sólo a partir de la ubicación de los marcadores. En la Figura 9.16 se observa que esto no es así dado que para dos posiciones distintas de la cámara, la traslación entre los marcadores $T_{\text{marcadores}}$ es diferente de la traslación de los centros de las cámaras T_{camaras} . Otro aspecto importante a destacar es que no basta con contar con la posición relativa entre las “dos cámaras”, sino que además ésta debe estar expresada en el sistema de coordenadas de una de las cámaras.

Procedimiento para realizar reconstrucción estéreo con una sola cámara

- Calibración del sistema Cámara - Marcadores. Esto incluye los parámetros intrínsecos de la cámara y el movimiento que lleva desde el sistema de coordenadas de los marcadores hasta el sistema de coordenadas de la cámara (H).
- Adquisición de la imagen izquierda y de la posición de los marcadores en ese instante.
- Adquisición de la imagen derecha y de la posición de los marcadores en ese instante.
- Cálculo de la posición relativa entre “las cámaras”. En la Figura 9.17 se aprecian claramente las transformaciones que entran en juego en este punto, donde H es la transformación entre los marcadores adheridos a la cámara y la

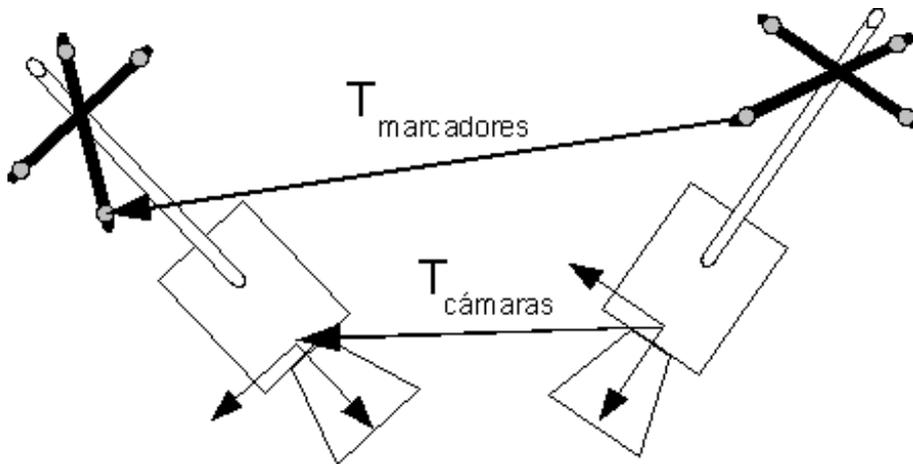


Figura 9.16: Se observa que sólo con la posición relativa entre los marcadores no se puede conocer la posición relativa entre las cámaras

misma; y M_{izq} y M_{der} son los movimientos rígidos que ubican al sistema de referencia de los marcadores respecto del localizador. Observando lo anterior se deduce que la matriz de calibración estereo queda

$$[R | t] = (M_{izq}H)^{-1}M_{der}H \quad (9.1)$$

- Una vez que se determinó la matriz de calibración, se procede a rectificar las imágenes.
- Con las imágenes rectificadas se calculan los mapas de disparidad y luego se realiza la triangulación.

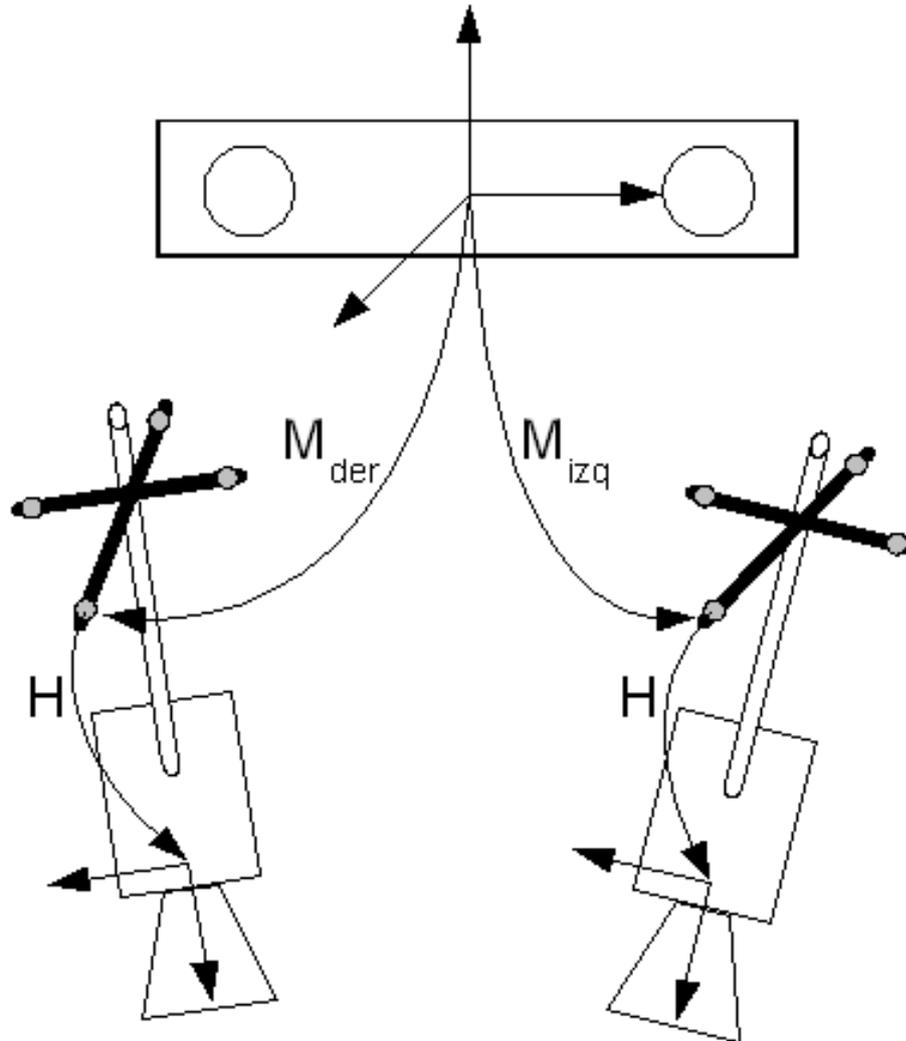


Figura 9.17: Esquema de transformaciones y movimientos vinculados a la realización de reconstrucción estereo con una sola cámara

9.5. Reconstrucción de las superficies

Una vez calculados los mapas de disparidad y la matriz Q definida en la Sección 8.10 (matriz imprescindible para la reconstrucción estereo dado que contiene toda la información de la geometría y de la óptica del par de cámaras cuando éstas están rectificadas) se procede a realizar la triangulación de puntos como se vio en la Ecuación 8.29. De esta forma se obtiene una nube de puntos que pueden ser utilizados para reconstruir una superficie. Para visualizar los mismos se utilizaron 3 métodos:

- Ploteo de puntos con niveles de gris utilizando la función `scatter3` de Matlab [64].

- Construyendo una malla utilizando la triangulación de Delaunay implementada en Matlab por Luigi Giaccari [25].
- Utilizando el programa Meshlab [46].

9.6. Resultados

En esta sección se muestran los resultados obtenidos con el microscopio del Hospital de Tacuarembó, los modelos 3D reconstruidos así como estimaciones de los errores cometidos en la reconstrucción.

Se recuerda que la implementación final resultó en utilizar una sola cámara y el localizador Polaris, éste último para obtener la posición relativa entre las cámaras. En la Figura 9.18 se puede ver el modelo utilizado para realizar la reconstrucción que no es otro que la parte posterior del cráneo, al cual se le agregaron relieves y zonas de mayor textura para facilitar la búsqueda de puntos correspondientes.

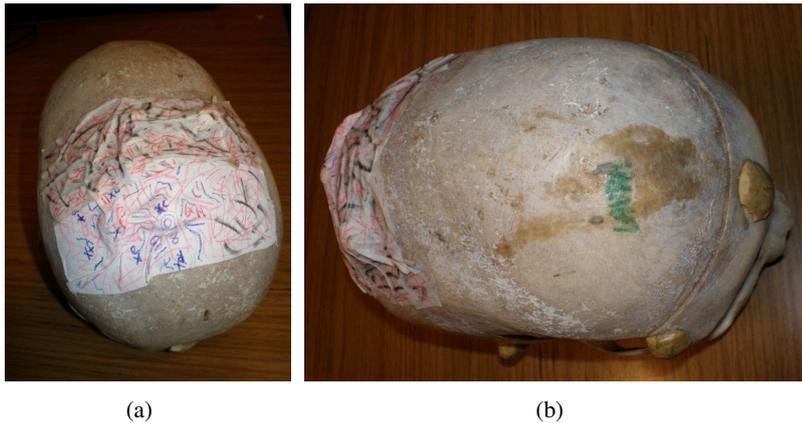
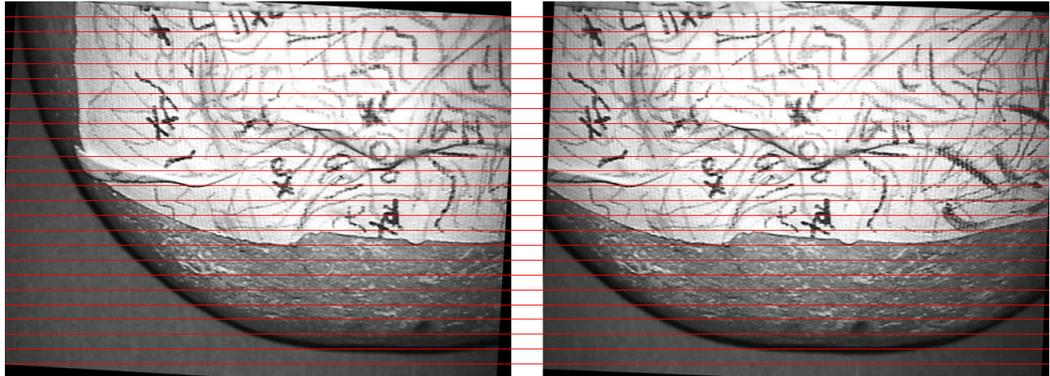


Figura 9.18: Modelo utilizado para la reconstrucción estereo.

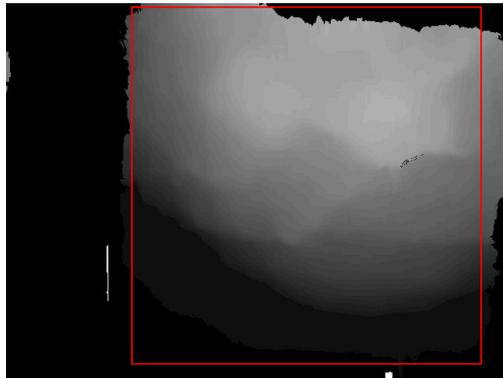
En las Figuras 9.19 y 9.20 se muestran dos ejemplos de rectificación y búsqueda de correspondientes (mapas de disparidad) de imágenes adquiridas en el Hospital utilizando una sola cámara y el localizador. Como se explicó anteriormente, el localizador brinda la posición relativa entre las cámaras, pero expresada en un sistema de coordenadas que no es el de la cámara izquierda. Para conseguir esto último se debe contar también con la calibración de la cámara, tanto los parámetros intrínsecos como el movimiento que ubica a la cámara respecto de los marcadores que están adheridos al microscopio H .

En las Figuras 9.19 y 9.20 se muestran dos ejemplos de rectificación con sus correspondientes mapas de disparidad. A simple vista se observa que las imágenes fueron bien rectificadas, hecho que se puede corroborar con la ayuda de las líneas que se dibujan sobre las imágenes. Esta buena rectificación se ve reflejada en los mapas de disparidad, obtenidos utilizando Block Matching, donde se pueden distinguir

diferentes niveles de profundidad. Estos mapas de disparidad son suaves y muestran relieves que se condicen con la superficie real. Por ejemplo se distinguen en la zona central del mapa de disparidad dos relieves pronunciados que fueron agregados a la superficie del cráneo. La zona gris que se observa en la parte inferior de los mapas de disparidad es debido a que se restringió la búsqueda de disparidades al volumen que se deseaba reconstruir.



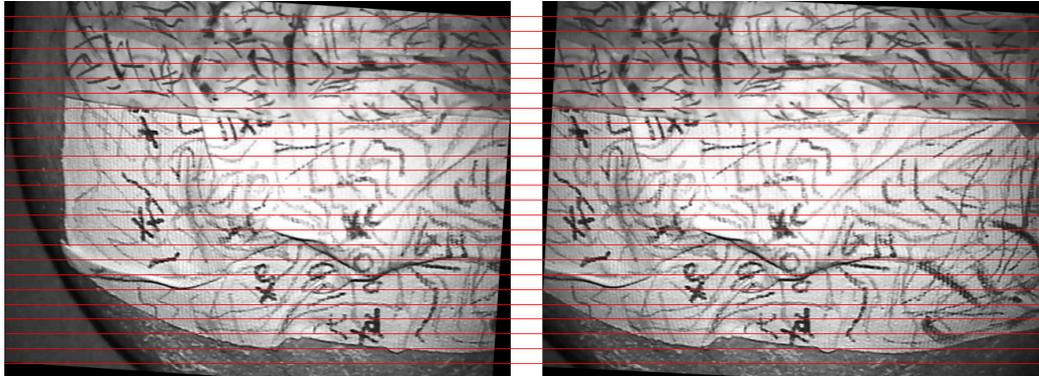
(a) Imágenes rectificadas para el Ejemplo 1



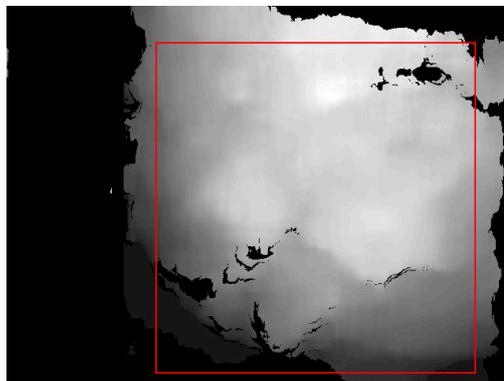
(b) Mapa de disparidad del Ejemplo 1 y zona en la que se realizó la reconstrucción

Figura 9.19: Ejemplo 1 de imágenes rectificadas y el mapa de disparidad de la imagen izquierda utilizando el dispositivo estereo armado en el Hospital de Tacuarembó.

En las imágenes de las Figuras 9.21, 9.22 y 9.23 se observan vistas desde distintas posiciones de las superficies del Ejemplo 1 obtenidas a partir de tres métodos distintos: ploteo de una nube de puntos con textura, triangulación de una nube de puntos y reconstrucción de la superficie utilizando el programa MeshLab respectivamente. La reconstrucción se realizó a partir del par de imágenes de la Figura 9.19(a), pero no se reconstruyó toda la escena sino una región de la misma, que es la región donde el mapa de disparidad da mejores resultados. La región de la imagen que se reconstruyó se puede ver en la Figura 9.19(b). En las Figuras 9.24, 9.25 y 9.26 se observan de la misma manera que para el Ejemplo 1, imágenes de la



(a) Imágenes rectificadas para el Ejemplo 2



(b) Mapa de disparidad del Ejemplo 2 y zona en la que se realizó la reconstrucción

Figura 9.20: Ejemplo 2 de imágenes rectificadas y el mapa de disparidad de la imagen derecha utilizando el dispositivo estéreo armado en el Hospital de Tacuarembó

superficie reconstruida en el Ejemplo 2. Al igual que con el Ejemplo 1, aquí también se reconstruye sólo una parte de la imagen.

Como resultado, se puede decir que se logró reconstruir correctamente la curvatura del cráneo. Dado que se lograron reconstruir los relieves agregados al cráneo, se concluye que la resolución alcanzada fue suficiente. Para ver claramente la resolución lograda se puede observar la Figura 9.21(d).

En cuanto a la zona gris del mapa de disparidad del Ejemplo 1, se observa que se reconstruye en un plano de coordenada z constante, y que esta coordenada es la mayor profundidad obtenida. Esto es así porque a la hora de determinar los niveles de disparidad permitidos, no se tuvieron en cuenta puntos con mayor profundidad.

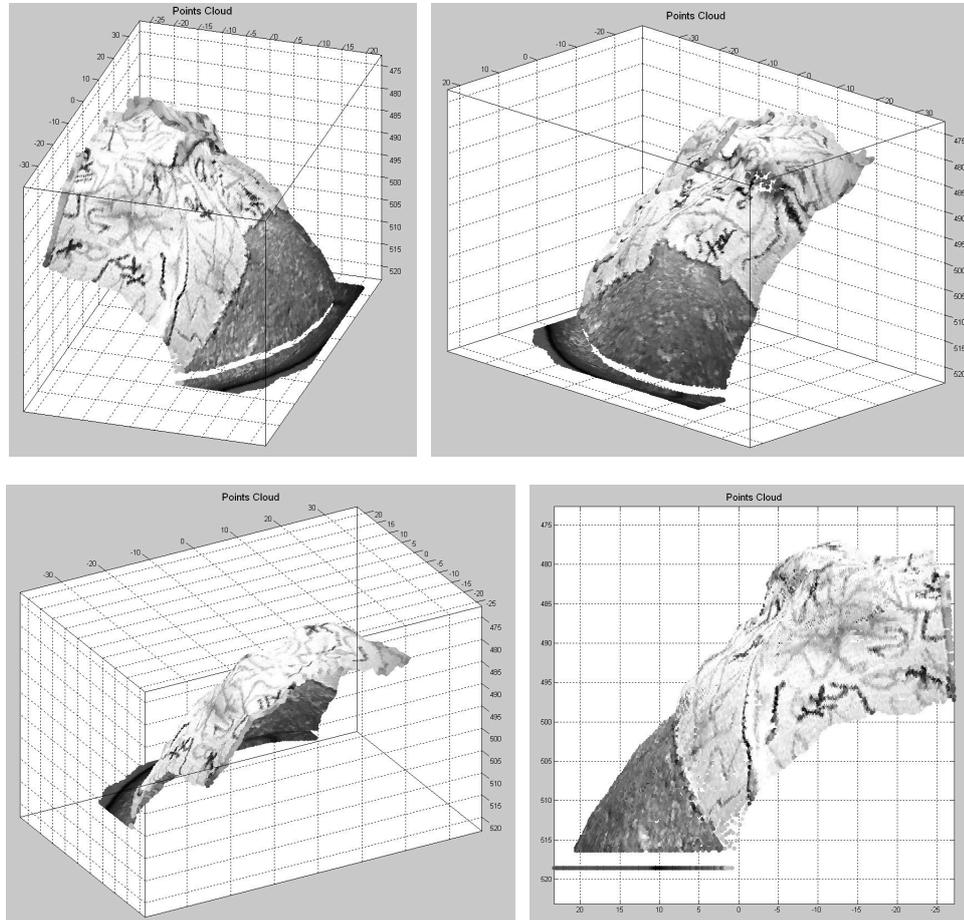


Figura 9.21: Reconstrucción de la superficie generada a partir del par de imágenes del Ejemplo 1 utilizando una nube de puntos con textura en Matlab

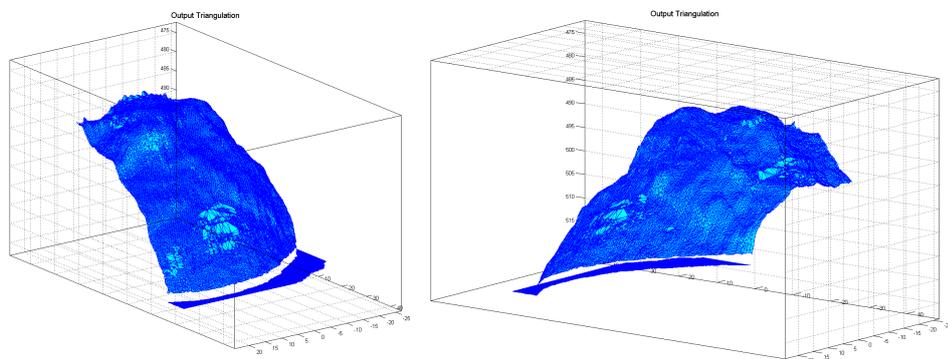


Figura 9.22: Reconstrucción de la superficie del Ejemplo 1 utilizando triangulación de una nube de puntos en Matlab

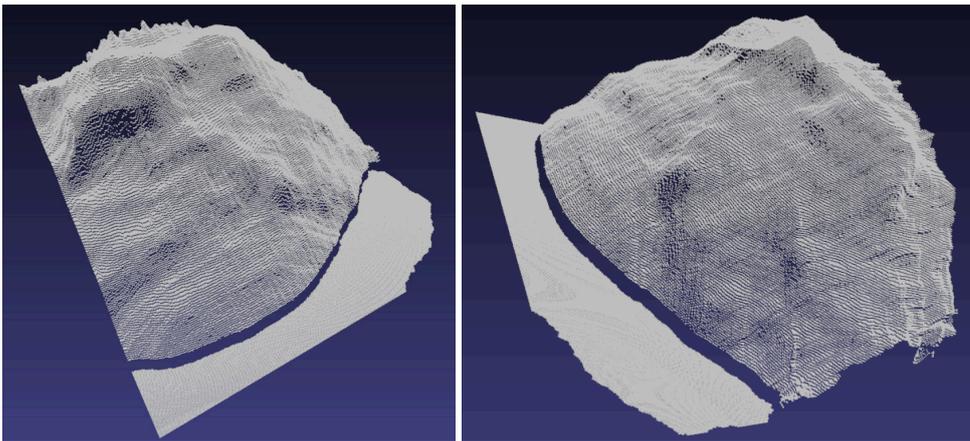


Figura 9.23: Reconstrucción de la superficie del Ejemplo 1 utilizando MeshLab

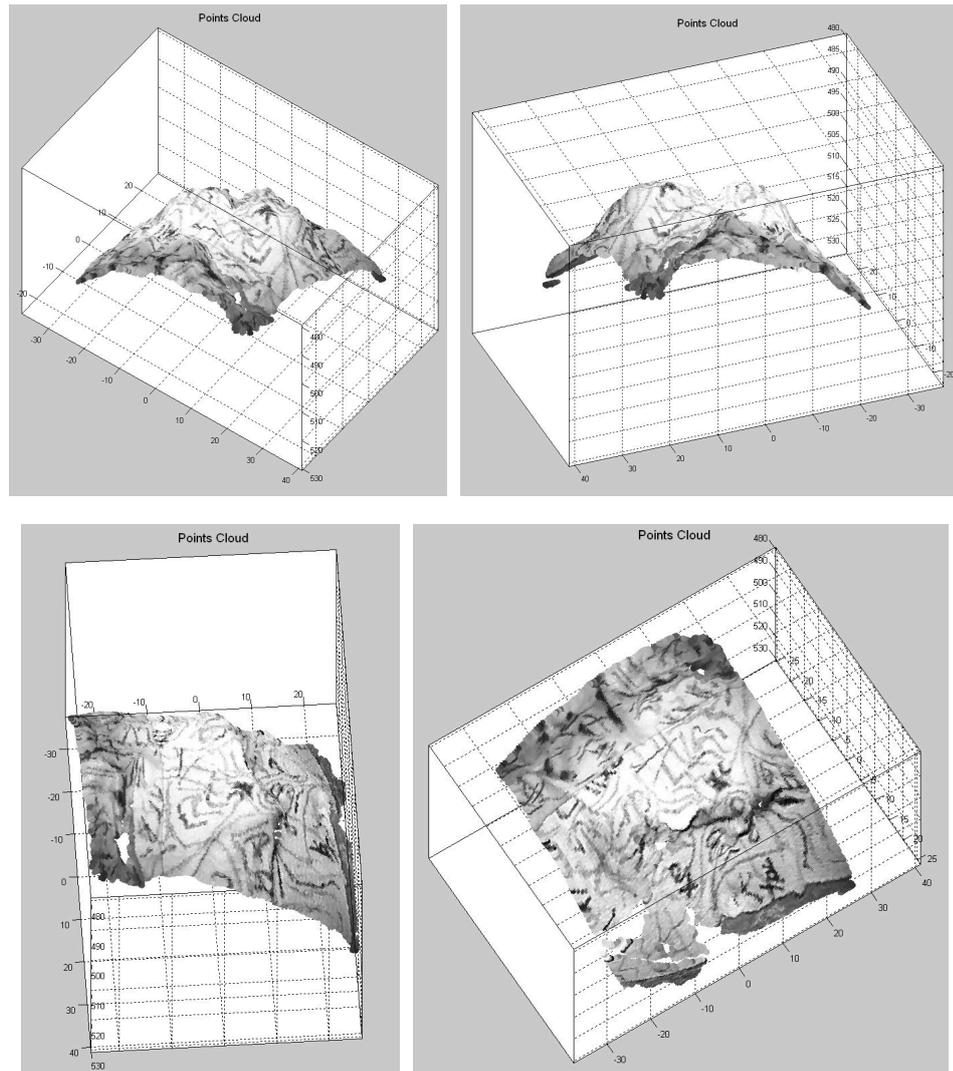


Figura 9.24: Reconstrucción de la superficie generada a partir del par de imágenes del Ejemplo 2 utilizando una nube de puntos con textura en Matlab.

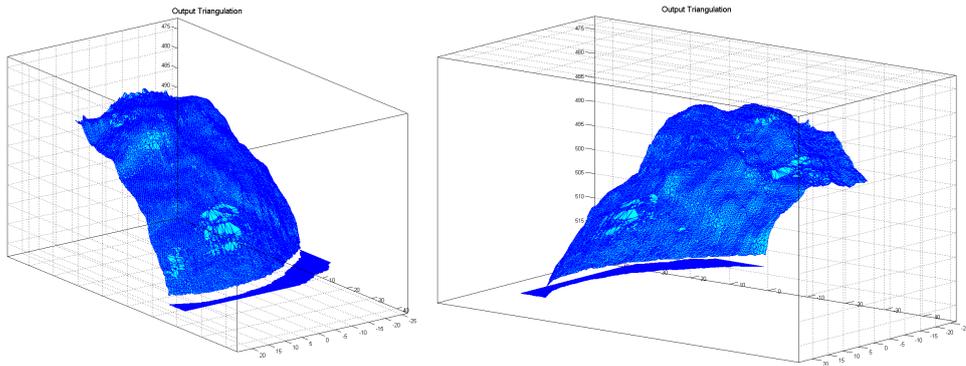


Figura 9.25: Reconstrucción de la superficie del Ejemplo 2 utilizando triangulación de una nube de puntos en Matlab.

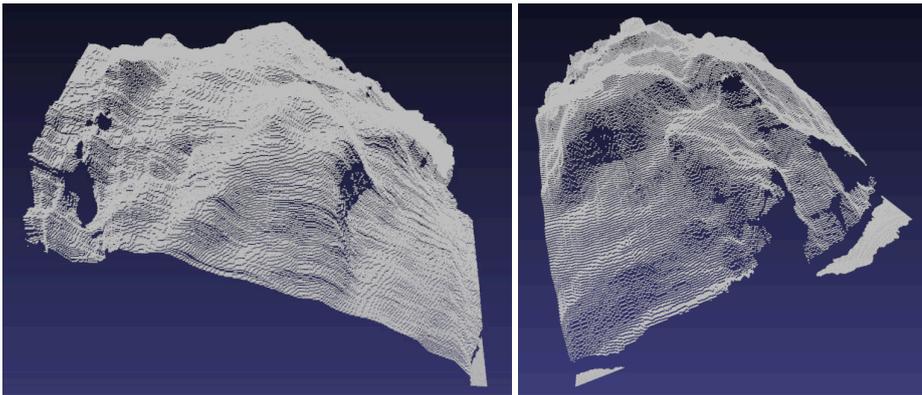


Figura 9.26: Reconstrucción de la superficie del Ejemplo 2 utilizando MeshLab.

9.7. Evaluación de resultados

A continuación se presenta un estudio del error de reconstrucción estéreo cometido para los Ejemplos 1 y 2 mostrados en la sección anterior. La forma de analizar el error al reconstruir es tomar un punto en el espacio del cual se conoce la posición respecto de la cámara, y luego a partir de sus coordenadas en ambas imágenes realizar la triangulación para luego comparar el resultado con la posición previamente conocida. Para analizar el error cometido en la reconstrucción, se realizó el siguiente procedimiento:

- Se toman las dos imágenes estéreo para la reconstrucción.
- Con la escena estática se adquieren, utilizando el localizador, coordenadas en el espacio de puntos específicos de la escena; en este caso se marcaron seis puntos sobre la superficie del cráneo (las cruces que se observan en las imágenes).
- Luego se segmentan manualmente las proyecciones de los seis puntos de la escena en ambas imágenes.
- Se realiza el proceso de reconstrucción estéreo dado que se cuenta con toda la información (matriz Q) para triangular los seis puntos segmentados.
- Con las coordenadas obtenidas con el localizador y las coordenadas obtenidas a partir de la triangulación (previamente se deben modificar para tenerlas en el mismo sistema de referencia) se calcula el Error RMS en la reconstrucción estéreo.

9.7.1. Error cometido en el Ejemplo 1

Para los seis puntos que aparecen en la Tabla 9.1 el error RMS es de 1.209944 mm.

9.7.2. Error cometido en el Ejemplo 2

Para los seis puntos que aparecen en la Tabla 9.2 el error RMS es de 2.386927 mm.

Los parámetros que se utilizaron para realizar la reconstrucción en este caso fueron $f = 4312$, $T = 15,3920$ y los puntos están a una distancia de aproximadamente 500 mm. Con estos valores la resolución teórica en z que se tiene es de 3,8 mm ya que un cambio de uno en la disparidad equivale a un salto de 3,8 mm en z . Como se recordará, Block Matching, que fue el algoritmo de búsqueda de puntos correspondientes, implementa una disparidad subpíxelica x16 de forma que la resolución que se obtiene en los mapas de disparidad resulta en 0.2375 mm.

Puntos adquiridos con Polaris			Puntos Obtenidos		
x	y	z	x	y	z
-30.98	-21.36	506.80	-37.38	-23.34	495.38
-17.20	-23.82	497.32	-23.24	-25.15	488.10
-0.38	-10.00	493.22	-7.51	-11.06	481.02
2.73	6.46	510.99	-5.37	5.30	499.11
-11.80	1.09	508.88	-19.60	-0.60	499.11
-30.97	-7.26	511.80	-38.29	-9.47	502.89

Puntos transformados			Diferencia		
x	y	z	dx	dy	dz
-30.89	-20.98	505.32	-0.10	-0.38	1.48
-16.57	-23.52	498.65	-0.64	-0.30	-1.33
0.13	-10.27	492.22	-0.51	0.27	1.00
2.32	6.06	510.33	0.41	0.40	0.65
-12.18	0.87	509.74	0.37	0.22	-0.86
-31.43	-7.04	512.74	0.46	-0.22	-0.94

Tabla 9.1: Primer Ejemplo

Puntos adquiridos con Polaris			Puntos Obtenidos		
x	y	z	x	y	z
-33.71	-5.91	507.63	-38.75	-8.62	488.10
-18.74	-7.39	498.66	-24.28	-9.30	470.79
-2.24	6.17	494.56	-9.43	4.89	470.79
1.08	22.38	513.04	-8.19	20.69	488.10
-14.05	16.72	510.76	-21.89	14.80	488.10
-33.52	8.21	513.28	-39.72	5.45	491.71

Puntos transformados			Diferencia		
x	y	z	dx	dy	dz
-33.42	-5.07	510.40	-0.29	-0.83	-2.77
-17.41	-7.69	494.70	-1.33	0.30	3.96
-1.49	5.24	495.70	-0.75	0.93	-1.14
-0.62	21.64	512.47	1.70	0.75	0.57
-14.70	16.89	511.28	0.65	-0.17	-0.52
-33.54	9.18	513.39	0.02	-0.97	-0.11

Tabla 9.2: Segundo Ejemplo

9.8. Conclusiones

Se realizó la reconstrucción estéreo de una superficie utilizando el microscopio del Hospital de Tacuarembó y el localizador, todo esto a partir de un desarrollo e implementación propia. Se realizaron pruebas reconstruyendo superficies con resultados satisfactorios teniendo en cuenta el corto período de tiempo en que se pudieron realizar las pruebas. Como resultado se obtuvo que para una disposición de zoom pequeño y una distancia focal de aproximadamente 450 mm se llega a resoluciones en la profundidad de 3.5 mm, que se transforman en 0.22 si se considera la disparidad subpíxelica que implementa el algoritmo Block Matching; por otro lado los errores RMS cometidos están entre 1 y 2.5 mm.

Se considera que las pruebas realizadas no fueron suficientes como para asegurar estas mediciones y poder mejorar el proceso; es por esto que aún hay trabajo por hacer para acreditar las medidas de error. Queda también pendiente la realización de la reconstrucción estéreo con el tipo de imágenes típicas de una operación dado que será fundamental saber si es posible encontrar correspondencias entre las imágenes para la textura de la superficie cerebral. En cuanto a los equipos, falta analizar qué resultados se obtienen al utilizar dos cámaras iguales, con tarjetas adquisidoras iguales, colocadas en los oculares del microscopio.

Capítulo 10

IGSTK - Software para cirugía guiada por imágenes

Un sistema de guiado por imágenes es una combinación de elementos de hardware y software. El software es el encargado de recoger información de diferentes fuentes (sistemas de localización, estudios médicos, dispositivos de video, transductores de ultrasonido, etc), integrarla de forma consistente y utilizarla para generar información que facilite la ejecución de una intervención.

La calidad del software no está asociada a la capacidad de procesamiento de las imágenes o al diseño gráfico de la aplicación sino que existen otros factores que en este tipo de aplicaciones cobran mayor importancia.

Al integrar las distintas fuentes de información es fundamental una perfecta sincronización de las mismas y un chequeo de la validez de los datos que se obtienen. La información que se genera debe ser desplegada en *tiempo real* por lo que los tiempos de procesamiento deben ser necesariamente tenidos en cuenta.

Es deseable la mínima interacción posible entre el cirujano y la aplicación ya que el profesional debe estar enfocado en los aspectos médicos de la intervención.

La principal característica que debe ser tomada en cuenta a la hora de evaluar un sistema de guiado por imágenes es la seguridad. Se debe brindar ayuda sin poner en riesgo al paciente. A medida que aumentan las capacidades del software y la interacción de éste con otros dispositivos se incrementa también el riesgo de que ocurran fallas. Para evitar comportamientos no deseados (desconexiones, fallas en los elementos de hardware, o en la interacción con el usuario, etc) es necesario tener un completo control de los posibles flujos de la aplicación. Esto implica tomar medidas preventivas que influyen directamente en la arquitectura de diseño y en las técnicas de programación [32, 16].

10.1. IGSTK

IGSTK (Image Guided Surgery Toolkit) [35, 22, 10] es una librería open-source escrita en C++ que provee los componentes básicos necesarios para desarrollar una

aplicación de cirugía guiada por imágenes. IGSTK está desarrollado sobre algunas librerías de uso extendido en lo que a aplicaciones de investigación biomédica se refiere. Éstas son:

- ITK (Insight Segmentation and Registration Toolkit) [36]
- VTK (The Visualization Toolkit) [37]
- GUI toolkits (FLTK y Qt) [34][14]

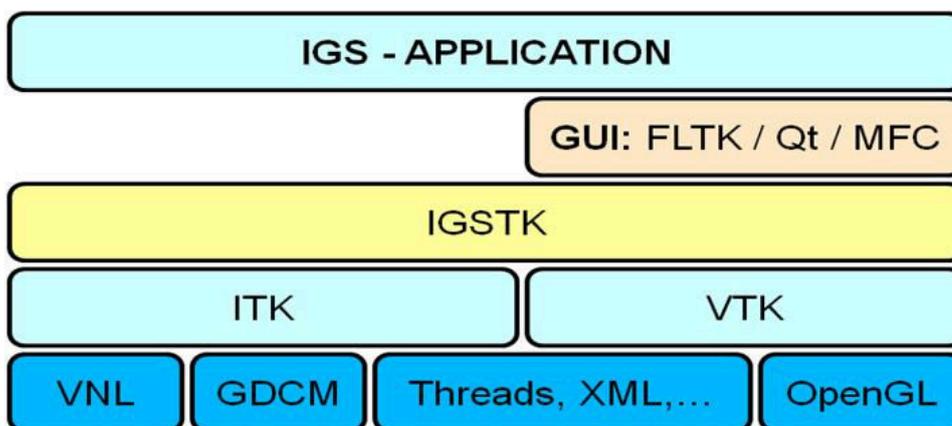


Figura 10.1: Arquitectura utilizada por IGSTK. Figura extraída de [10].

ITK es utilizada fundamentalmente para el procesamiento de las imágenes. VTK se usa, en cambio, para mostrar las imágenes y los modelos geométricos. Para poder utilizar IGSTK fue necesario por tanto aprender los conceptos básicos de las librerías sobre las cuales está construido, tarea que consumió un importante esfuerzo.

IGSTK está diseñado en base a máquinas de estado; la principal razón para esto es que ofrece un mecanismo para realizar una validación formal del software. Es posible determinar, por ejemplo, que la aplicación nunca ingresa en un estado de error. Además se hacen explícitas las interacciones posibles entre los distintos componentes del software así como entre el software y los posibles usuarios.

Implementar el software como una colección de componentes es también una decisión motivada por disminuir los riesgos en el paciente. Cuando el software puede separarse en pequeños componentes se simplifica la tarea de verificar el correcto comportamiento de cada uno de ellos.

Si bien IGSTK está escrito sobre librerías como ITK y VTK, ofrece muchas menos posibilidades que éstas para el desarrollo de aplicaciones. Brinda sólo un conjunto restringido de las funcionalidades de las mismas, esto se debe a que un incremento en las funcionalidades ofrecidas implica también aumentar las posibilidades de incurrir en errores o en, al menos, comportamientos no previstos del software. Es por eso que los comportamientos y funcionalidades ofrecidas tienen interfaces bien definidas y testeadas entre los componentes. Resulta evidente que

esta medida además de aumentar las medidas de seguridad hace que se restrinjan las posibilidades de desarrollo del software.

Componentes de IGSTK

Los componentes del toolkit pueden clasificarse en:

- **Display:** incluye los componentes que interactúan con la aplicación GUI (Guided User Interface). Éstas son las clases que van a mostrar la escena en una o más de las ventanas del GUI.
- **Representación Geométrica:** la escena es representada en dos niveles. Por un lado se encuentran los objetos a ser representados, vistos como figuras geométricas y por el otro lado se tiene la representación de estos objetos. La representación geométrica toma en consideración aspectos como la forma y el tamaño de los objetos a ser representados en la sala de operaciones. Un objeto puede ser, por ejemplo, un cilindro, un cubo, una elipse, etc.
- **Representación Visual:** la representación geométrica de los objetos sólo toma en cuenta factores como la forma, el tamaño y la posición de los mismos, pero no considera como van a verse representados los objetos en la escena. La representación visual de un objeto toma en cuenta aspectos como el color o la opacidad de las superficies a representar.
- **Seguimiento:** el seguimiento es una parte fundamental en las aplicaciones de cirugía guiadas por imágenes. Es lo que permite saber con precisión la ubicación de las herramientas que están siendo utilizadas por el cirujano en un momento dado. IGSTK tiene clases especializadas en la comunicación con los localizadores comerciales, las cuales permiten interactuar eficazmente con estos dispositivos. En particular IGSTK provee una clase que permite adquirir las coordenadas provistas por el localizador Polaris que se utilizó en esta aplicación.
- **Fuentes de datos:** este componente agrupa todas las clases que se utilizan para obtener la información que luego será agregada en la escena. Estas clases se utilizan, por ejemplo, para leer archivos DICOM, tomografías computadas, resonancias magnéticas, estructuras mesh o posiciones de algunos objetos previamente calculadas.

- **Calibración y Registrado:** Uno de los objetivos fundamentales, sino el principal, de las aplicaciones de cirugía guiada por imágenes es la combinación de información obtenida a partir de estudios previos con la posición de las herramientas del cirujano, las cuales son obtenidas durante la intervención. Para combinar estas dos fuentes de información eficientemente se hace imprescindible una buena calibración y registrado.
- **Infraestructura:** agrupa clases que se usan dentro de las clases principales. Estas clases se usan para manejar las máquinas de estados, los eventos, los relojes (utilizados para chequear que la información obtenida no es vieja) y los observers (que son quienes se enteran que sucedió algo nuevo).
- **Servicios:** estas clases son especializadas en la comunicación de IGSTK con otros dispositivos (a través de puertos serie o sockets). También existe aquí una clase especializada en el logger (qué métodos se ejecutaron) de la aplicación, que es de utilidad para detectar errores en la misma.

10.2. Sistema de Seguimiento

Existen en IGSTK clases encargadas de manejar un sistema de seguimiento [17], las mismas se encargan de obtener información del localizador y hacerla disponible para otros componentes de IGSTK. También se manejan mediante IGSTK los posibles errores de comunicación con el hardware y la frecuencia a la que se refrescan los datos.

Las principales clases encargadas del seguimiento de las herramientas son `Tracker` y `TrackerTool`. `Tracker` es una clase abstracta que define el comportamiento de un dispositivo localizador físico y establece la interfaz de interacción del mismo con una aplicación de software. Esta clase está pensada para modelar el comportamiento de cualquiera de los localizadores comerciales disponibles. En particular, existe una clase llamada `PolarisTracker` que hereda de `Tracker` e implementa el comportamiento del localizador `Polaris` que se usó en esta aplicación.

A las herramientas que utiliza el localizador para determinar su posición se les suele llamar `tools`. La clase `TrackerTool` es utilizada por IGSTK para mapear la estructura física de las diferentes `tools` al nivel de software. Al igual que como sucedía con la clase `Tracker`, la clase `TrackerTool` es abstracta siendo la clase `PolarisTrackerTool` la encargada de modelar las herramientas de seguimiento utilizadas por el `Polaris Tracker`.

Un objeto `Tracker` puede estar relacionado con varios objetos `TrackerTool`; son estos últimos los que representan la información que llega desde el sistema de localización para las distintas herramientas que se utilizan en la sala de operaciones.

El sistema de localización cuenta con un procesador al cual se le debe especificar la tasa de refresco de la posición y el estado de las distintas herramientas. Para realizar esto IGSTK cuenta con un objeto `PulseGenerator` que es una forma asíncrona

de generar un pulso visible para el sistema de seguimiento. De esta manera el funcionamiento de la aplicación que corre en IGSTK no influye en la tasa de refresco de la información proveniente del sistema de tracking. La información que proviene del sistema de seguimiento no es utilizada inmediatamente en la aplicación, sino que se almacena en un buffer. La información guardada en este buffer puede ser actualizada si se da la señal del PulseGenerator o puede ser utilizada cuando la aplicación lo requiera.

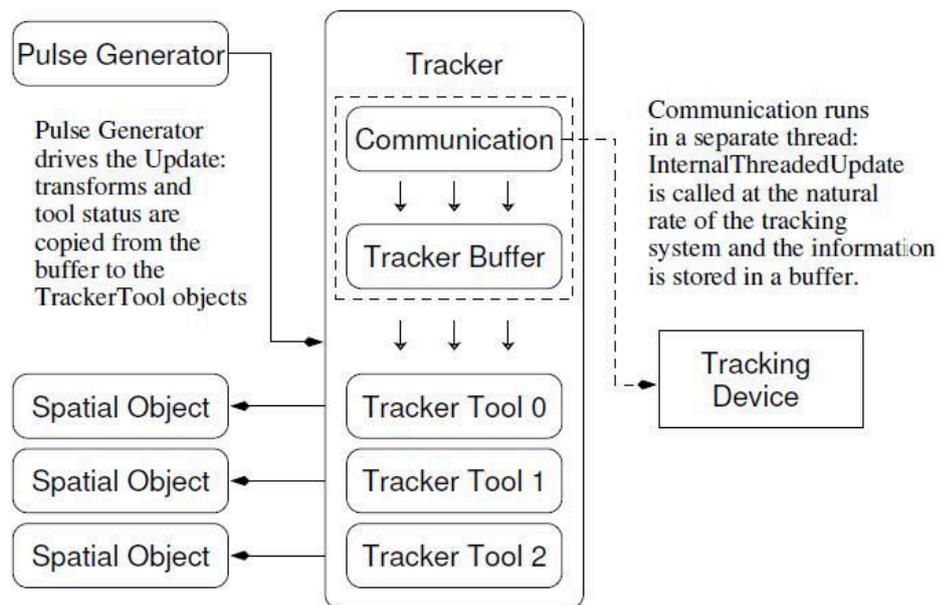


Figura 10.2: Clases encargadas del seguimiento de herramientas en IGSTK. Imagen extraída de [10].

10.2.1. Comunicación con el localizador Polaris

La comunicación con el sistema de seguimiento es realizada por medio de una clase especializada en la comunicación, la clase se llama *Communication*. Esta clase está diseñada para manejar toda la comunicación con el sistema de seguimiento por lo que debe verificar las siguientes condiciones:

- Cada puerto debe tener asociado un objeto de comunicación que actúa como proxy para ese puerto.
- La información no puede ser simplemente congelada en el caso de que se caiga la comunicación, en este caso se debe reportar correctamente el error ocurrido. Este punto se basa en que este dispositivo se encuentra en la sala de operaciones, por lo que puede sufrir una desconexión por distintas causas. Los puertos seriales por defecto se encuentran en estado BLOQUEADO,

si por algún motivo se cae la conexión o se rompe la cadena de datos se congelará la ejecución del programa y esto claramente no es aceptable en una aplicación para cirugía. Es por esto que el puerto de comunicación debe estar configurado en modo NO-BLOQUEADO antes de recibir o transmitir cualquier información.

- Se debe crear un registro completo de todas las comunicaciones que se den.
- El flujo de datos se puede reproducir en cualquier momento con fines de prueba.

Estos últimos dos puntos tratan sobre la capacidad de testeo de los componentes para realizar el seguimiento así como del sistema de seguimiento. También permite la depuración y prueba de una aplicación IGSTK facilitando de esta manera la solución de problemas. Además posibilita simular aplicaciones que utilizan un objeto Tracker sin la necesidad de contar con un localizador.

10.2.2. Creación de herramientas

Físicas Anteriormente se expresó que el localizador puede seguir varias herramientas, esto implica poder identificarlas correctamente. Para distinguir una herramienta de otra es necesario diseñarlas cuidando algunos criterios, el procedimiento de creado de una herramienta se denomina caracterización de la herramienta y se detalla en [3].

Cada herramienta tiene un archivo de definición propio que la describe completamente al sistema. La información almacenada en dicho archivo de definición incluye la geometría de los marcadores, el máximo ángulo admisible entre un marcador y cada sensor del Localizador Polaris y el mínimo número de marcadores necesarios para que se detecte una herramienta, entre otros.

Software Cada vez que se crea una herramienta utilizando IGSTK se siguen los siguientes pasos:

1. Se crea un objeto `PolarisTrackerTool`.
2. Se especifica que se trata de una herramienta pasiva (en nuestro caso).
3. Se asocia el archivo de definición al objeto creado.
4. Se asocia la instancia `PolarisTrackerTool` a una instancia de `PolarisTracker` (en un caso más general podría haber más de un localizador, ya sea Polaris u otra marca).
5. (Opcional) Se asocia una transformación de calibración a la herramienta. El localizador brinda la posición 3D del origen del sistema de coordenadas que define la geometría de la herramienta, pero en general, el interés no

radica en conocer la posición del origen de la herramienta sino de algún otro punto. Por ejemplo, es útil saber la ubicación de la punta del puntero, no así de los marcadores adheridos al puntero, que es la información que brinda el localizador. Si se conoce la transformación que ubica la punta respecto a los marcadores, se puede asociar esta transformación a la instancia de `PolarisTrackerTool` y así obtener directamente la posición de la punta respecto al localizador.

10.3. Objetos y sus representaciones

Objetos Espaciales La clase `SpatialObject` provee una herramienta sencilla con la cual poder manipular, consultar e interconectar objetos en el espacio físico. `SpatialObject` es una estructura de datos con la cual describir la geometría de distintos objetos. Cada objeto espacial contiene una transformación interna que sirve para ubicar y orientar el objeto en el espacio.

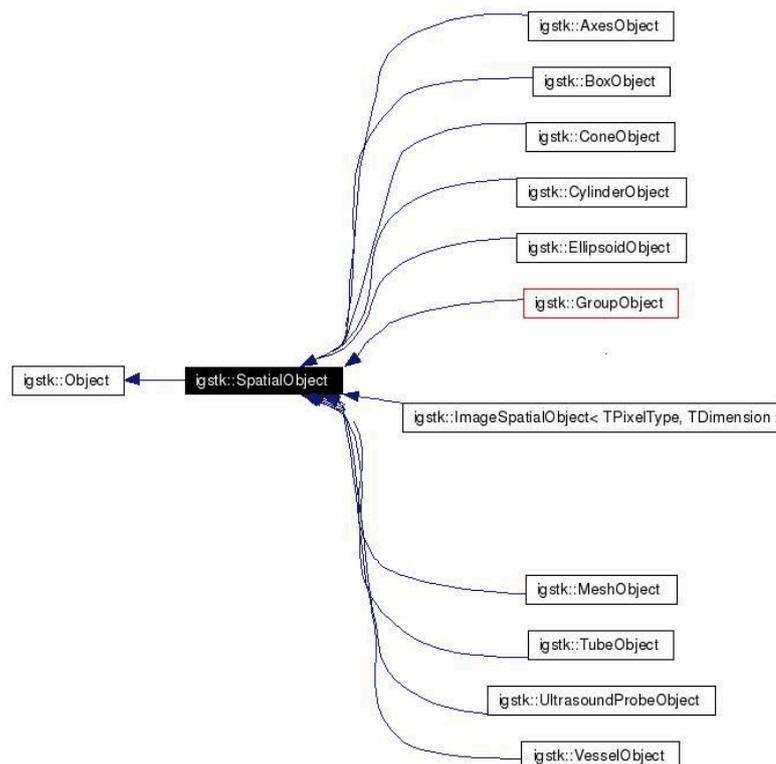


Figura 10.3: Herencia de algunos Objetos Espaciales en IGSTK. Imagen extraída de [10].

IGSTK tiene pre-definidos varios tipos de objetos espaciales, algunos son:

- `AxesObject`: Ejes cartesianos.
- `BoxObject` : Prismas.

- ConeObject: Conos.
- CylinderObject: Cilindros.
- ElipsoidObject: Elipsoides.
- ImageEspacialObject: Imágenes de estudios tomográficos y/o resonancias magnéticas.
- MeshObject: Mallas, es decir superficies en el espacio.

Representación de objetos espaciales Existe en IGSTK una clase llamada `SpatialObjectRepresentation` que se encarga de definir la visualización de cada uno de los objetos espaciales. Mientras que un objeto espacial define la geometría de un objeto dado, la representación del mismo será la que defina cómo se visualiza el objeto en la pantalla. Algunas de las características que se pueden cambiar son color, luminosidad y transparencia.

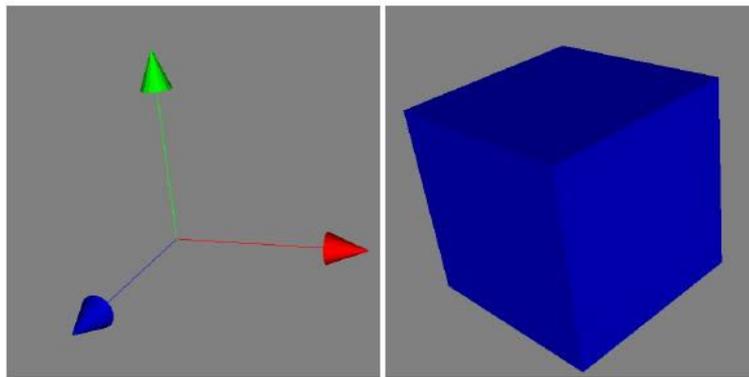


Figura 10.4: Ejemplo de representación de objetos en IGSTK.

10.4. Captura de Video

La captura de video se realizó utilizando la biblioteca OpenCV que a su vez utiliza la librería `videoInput` [65] para capturar en el sistema operativo Windows. IGSTK utiliza OpenCV para la captura de video en Windows pero es un tanto más complicado de implementar que si se utiliza directamente OpenCV o el propio `videoInput`. Esto se debe al especial cuidado que presta IGSTK en su implementación por estar diseñado para aplicaciones médicas. A continuación se describe cómo se implementa en IGSTK la captura de video.

10.4.1. Video Imager component: una clase un poco más general

Las imágenes obtenidas previamente a la operación suelen encontrarse en formato DICOM. En cambio la información obtenida durante la misma (por ejemplo

utilizando el microscopio, endoscopios o aparatos de ultrasonido) suelen no estarlo. Como consecuencia, es necesario poder convertir esas imágenes a un formato que sea entendido por la aplicación. Esa es la tarea del Video Imager Component. Una vez transferidas a la aplicación estas imágenes pueden ser utilizadas para el registrado (la alineación de imágenes obtenidas previamente a la operación con el paciente), para crear modelos 3D de imágenes de ultrasonido (siguiendo además al transductor de ultrasonido con un localizador) o para realidad aumentada. En nuestro caso nos interesa aplicar Realidad Aumentada a las imágenes obtenidas con el microscopio.

El componente Video Imager actúa como conexión entre dispositivos de video y otros componentes de IGSTK.

Es totalmente independiente del dispositivo específico que se esté utilizando para realizar la adquisición; existen sub clases que realizan la implementación específica para un hardware determinado.

Video Imager sólo se encarga de proveer una secuencia de video a otras clases. Concretamente, las clases `VideoFrameSpatialObject` y `VideoFrameRepresentation` son las encargadas de mostrar el video en la escena. La instancia de `VideoFrameSpatialObject` almacena características como resolución de imagen o número de canales (color o blanco y negro). Es un objeto más en la escena y por lo tanto se debe vincular con los demás objetos mediante alguna transformación. Al ser en este caso utilizado para Realidad Aumentada se va a asociar mediante la identidad a la cámara virtual OpenGL de la aplicación.

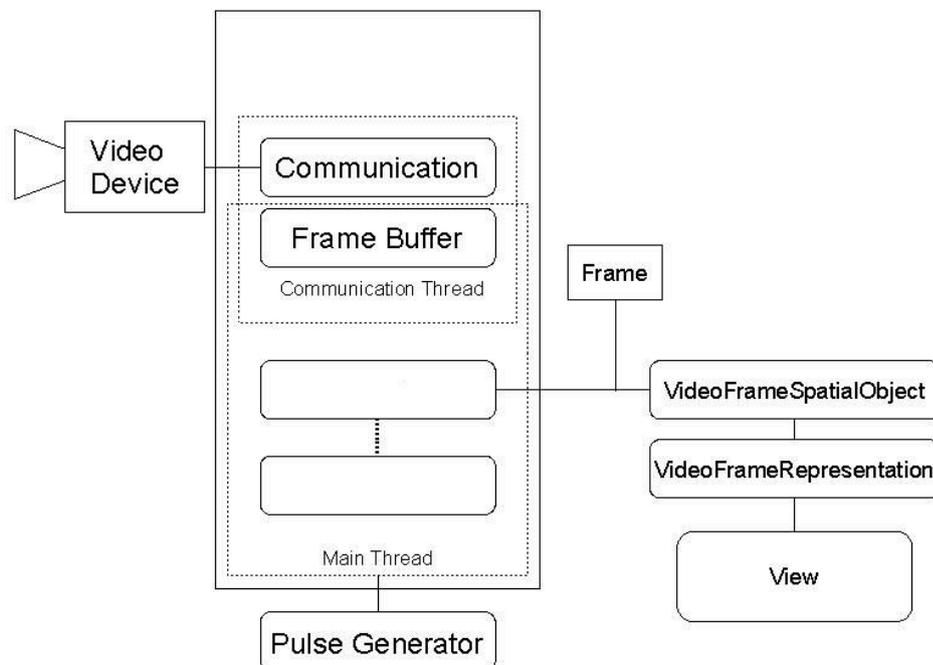


Figura 10.5: Componentes de la clase VideoImager. Figura extraída de [10]

La clase `VideoImager` se encarga sólo de describir una interfaz genérica capaz de adquirir imágenes de dispositivos tales como endoscopios, cámaras de video, etc. Un `VideoImager` es capaz de manejar uno o más `VideoImagerTool`: el `VideoImager` es el responsable de establecer la comunicación mientras que el `VideoImagerTool` es responsable de configurar los parámetros para un dispositivo determinado. `VideoImager` se comunica continuamente, utilizando un hilo separado al de la aplicación, con el hardware de aplicación y almacena cada cuadro temporalmente en un frame buffer. Cada `VideoImagerTool` contiene un frame buffer y es posible configurar la frecuencia con la cual la instancia de `VideoImager` lo actualiza.

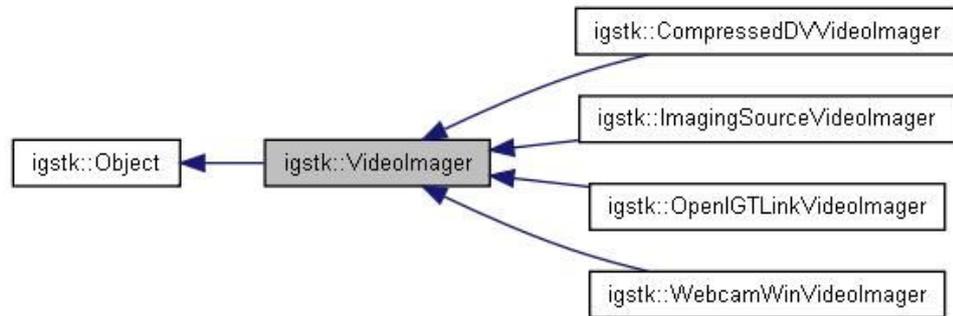
Para poder mostrar un frame deben ser conectados una instancia de `VideoFrameSpatialObject` con la instancia de `VideoImagerTool` que se desee. Ambas instancias deben contener las mismas propiedades en cuanto a resolución de imagen y número de canales. A su vez se debe conectar la instancia `VideoFrameSpatialObject` con una instancia de `VideoFrameRepresentation` (recordar que las representaciones de los objetos son quienes se agregan en la escena).

En resumen, los pasos que se deben seguir para agregar un video en la escena son los siguientes:

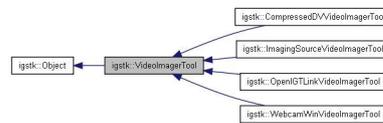
1. Se crea una instancia de `VideoFrameSpatialObject`, se configuran sus propiedades y se la ubica en el grafo de la escena.
2. Se crea una instancia de `VideoFrameRepresentation` y se la asocia al objeto que representa.
3. Se crea una instancia de `VideoImager` y se configura la tasa de envío de datos.
4. Se crea una instancia de `VideoImagerTool` con las mismas propiedades que la instancia de `VideoFrameSpatialObject` a la cual va a ser conectada.
5. Se asocian las instancias de `VideoImager` y `VideoImagerTool`.
6. Se conectan las instancias de `VideoFrameSpatialObject` y `VideoImagerTool`.
7. Se agrega a la escena la instancia de `VideoFrameRepresentation`.

10.4.2. Modificaciones Realizadas

- OpenCV permite modificar la resolución con la que se captura el video utilizando la función `CvCaptureCAM_DShow :: SetProperty()`. Sin embargo sólo permite hacerlo manteniendo una relación de aspecto de 4/3. Dicha función fue modificada para poder trabajar con resoluciones que no cumplan esta relación.



(a) Herencia de la clase VideoImager. Figura extraída de [10]



(b) Herencia de la clase VideoImagerTool. Figura extraída de [10]

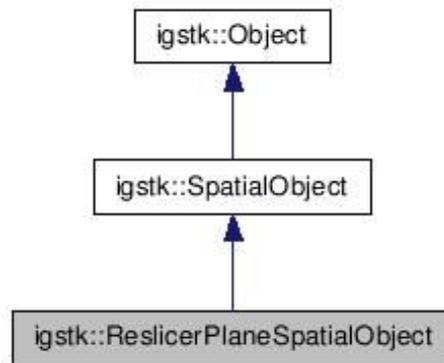
- La clase `igstkWebCamWinVideoImager` sólo funciona para resoluciones de 640×480 , por lo que se creó la clase `igstkMIRAWebCamWinVideoImager`, que agrega a las funcionalidades de `igstkWebCamWinVideoImager` la posibilidad de configurar la resolución.

También se modificó el orden de los canales. Originalmente se mostraban los cuadros en formato BGR, mientras que ahora se muestran en RGB.

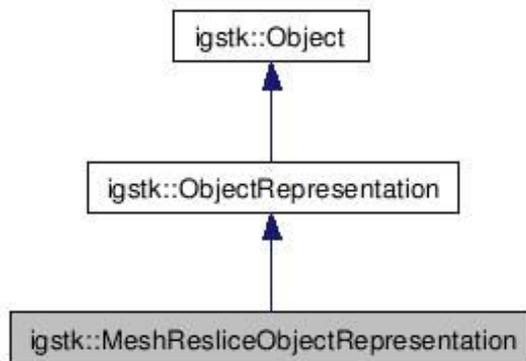
10.5. Reslicing

A un cirujano durante una intervención puede interesarle, por ejemplo, visualizar el corte de un estudio que pasa por determinado punto que en ese momento está tocando con su bisturí. Un estudio, en general, va a contener cortes que pasan cerca de ese punto pero no exactamente por ahí. Es necesario por tanto, adaptar la información que se va a mostrar al cirujano. A ese proceso de seleccionar un corte de un estudio que en principio podría no existir, se le llama *reslice*. El “*reslicing plane*” (plano de corte) puede tener una posición distinta a los planos existentes en la pila de estudios pero además puede diferir en su orientación.

IGSTK provee una implementación para realizar el *reslicing*. El componente encargado de realizar el *reslicing* contiene dos tipos de clases, una encargada de definir la posición y orientación del objeto a representar (`ReslicerPlaneSpatialObject`) y otras encargadas de la representación de los objetos (`ImageResliceObjectRepresentation` y `MeshResliceObjectRepresentation`). Estas últimas describen básicamente cómo debería ser representado el objeto (un estudio o un objeto de tipo *mesh*) en



(a) Herencia de la clase ReslicerPlaneSpatialObject. Figura extraída de [10]



(b) Herencia de la clase MeshResliceObjectRepresentation. Figura extraída de [10]

Figura 10.6: Jerarquía de clases de Reslicing

la pantalla.

En el caso de los estudios se construye el corte a ser mostrado a partir de los existentes mediante técnicas de interpolación. Existen tres tipos diferentes de interpolación: la del pixel más cercano, interpolación lineal e interpolación cúbica.

Para estructuras de tipo mesh lo que se muestra es un contorno resultante de la intersección del plano de corte (reslicer plane) con la estructura mesh. Dependiendo del plano de corte dicho contorno puede ser cerrado o no e incluso puede estar compuesto por varios subcontornos.

En este proyecto se utilizó el reslicing de una estructura que simula ser un tumor, como puede verse en la Figura 10.7. A continuación se muestran los pasos que se siguieron para mostrar el contorno resultante de la intersección del tumor con el plano que pasa por la punta del puntero y es perpendicular al eje óptico del microscopio.

1. Se crea una instancia de la clase `ReslicerPlaneSpatialObject`.
2. Se configuran las propiedades de dicha instancia de forma que el plano intersección se calcule a partir de la posición y orientación de un objeto que actúa de guía. Este objeto se encuentra en la punta del puntero y su orientación está dada por la orientación del eje óptico del microscopio.
3. Se acota el espacio de cálculo del plano de intersección a una caja que se determina a partir de la ubicación del modelo del tumor en el espacio.
4. Se asocia el objeto referencia a la instancia de la clase `ReslicerPlaneSpatialObject` creada en el paso 1.
5. Se crea una instancia de `MeshResliceObjectRepresentation`. Se configuran sus propiedades: opacidad, color, ancho de línea de contorno.
6. Se asocia la instancia de `MeshResliceObjectRepresentation` con el objeto espacial que representa al tumor y con la instancia de `ReslicerPlaneSpatialObject` creada en el paso 1.
7. Se agrega la instancia de `MeshResliceObjectRepresentation` a la escena.



(a)



(b)



(c)

Figura 10.7: Intersección entre el modelo 3D del tumor y el plano que pasa por la punta de la herramienta y es perpendicular al eje óptico, para tres profundidades distintas.

10.6. Lectura de parámetros

Para leer los parámetros de configuración de los distintos proyectos implementados se creó la clase `input_parameters`; la misma fue implementada utilizando la librería `OpenCV`. Esta clase se creó a los efectos de concentrar la configuración de todos los parámetros en un solo archivo y evitar así tener que modificar todos los proyectos que utilicen un determinado parámetro de entrada cada vez que dicho parámetro sea modificado. Cabe mencionar que esta clase no se encuentra dentro de la jerarquía de clases de `IGSTK`. La clase cuenta sólo con un constructor que recibe como parámetro la ubicación del archivo de configuración. Los parámetros que se pueden configurar son:

- Ancho y alto del video a capturar.
- Largo en mm de los lados de las casillas del patrón de calibración.
- Número de casillas con las que cuenta el damero.
- Tipo de patrón que se utilizará para calibrar: patrón plano o cúbico.
- Número de fotos a utilizar para calibrar en caso de utilizar patrón plano.
- Calibrar con todas las imágenes o con grupos de imágenes (caso de patrón plano).
- En caso de calibrar utilizando grupos de imágenes, se determina la cantidad a utilizar en cada grupo.
- Estimar o no el punto principal.
- Estimar o no la distorsión y establecer el modelo de distorsión.
- Forzar que los largos focales calculados sean iguales.
- Determinar si se va a registrar un objeto.

10.7. Integración con el Neuronavegador desarrollado por el IIE

Uno de los requerimientos del proyecto era que el sistema pudiera funcionar como un módulo del Neuronavegador que está siendo desarrollado por el Grupo de Tratamiento de Imágenes del Instituto de Ingeniería Eléctrica. Ésta fue una de las razones por la cual se desarrolló el software de pruebas utilizando la biblioteca `IGSTK`. Para facilitar la inserción del módulo de Realidad Aumentada en el microscopio quirúrgico dentro de la aplicación Neuronavegador fue necesaria la modificación de la clase `igstk::View` de la biblioteca `IGSTK` y el agregado de una clase que hereda de la misma.

La clase `igstk::View` de IGSTK es una clase abstracta que encapsula una serie de funciones de VTK relacionadas con el agregado de representaciones de objetos a la escena y al manejo de la cámara virtual. A diferencia de lo que sucede en VTK la utilización de las funciones en IGSTK se rige por una estructura de máquina de estados. Esta estructura apunta a tener un control total de los diferentes comportamientos de la aplicación dotándola así de una mayor seguridad. Esto a su vez restringe la utilización de las funcionalidades de VTK.

Existen en IGSTK dos clases que heredan de `igstk::View` y se encargan de la representación de los objetos de la escena en 2D y 3D respectivamente, éstas son `igstk::View2D` e `igstk::View3D`. Para superponer la escena virtual a la señal del video fue necesario crear una nueva clase `igstk::ViewAR`. Esto implicó modificar la clase `igstk::View`, pero se tuvo especial cuidado en no alterar el diseño en máquina de estados de la misma. Las modificaciones realizadas fueron las siguientes:

- Se agregaron dos atributos privados del tipo `vtkRender: m_ForegroundRenderer` y `m_BackgroundRenderer`. Originalmente la clase `igstk::View` contaba con un solo objeto del tipo `vtkRender` llamado `m_Renderer` que se encargaba de representar la escena. Ahora se tiene una representación de la escena en dos capas, una de fondo que se encarga de mostrar el video (`m_BackgroundRenderer`) y otra que cumple la función básica de mostrar la escena virtual (`m_ForegroundRenderer`). Cuando se dibuja la escena virtual se hace que el `m_ForegroundRenderer` apunte a lo mismo que el `m_Renderer` y que cuando se dibuja el video sea el `m_BackgroundRenderer` quien apunte a lo mismo que el `m_Renderer`.
- Se modificó el constructor de la clase `igstk::View`. Ahora se crea una ventana para visualizar la escena, se crean los dos objetos `vtkRender` definidos anteriormente, se define para la ventana una representación en dos capas y se asocia el `m_BackgroundRenderer` a la capa 0 (el fondo) y el `m_ForegroundRenderer` a la capa 1. Se configura a la capa del frente como capa de dibujo por defecto.
- Se crearon los métodos `View::SetBackgroundRendererOn` y `View::SetBackgroundRendererOff`. El primero establece como capa de dibujo por defecto la del fondo (*math*`m_BackgroundRenderer` apunta a lo mismo que `m_Renderer`) y como cámara virtual activa la asociada a la capa del fondo. El segundo método hace lo mismo pero con la capa del frente.

10.7.1. Nueva clase: `igstk::ViewAR`

La clase `igstk::ViewAR` hereda de `igstk::View` y cuenta con los siguientes métodos:

- **`ViewAR::ViewAR()`**: Constructor. Establece como por defecto la capa de adelante. Configura a la cámara asociada a la capa del fondo como proyección

paralela (para proyectar el video). La cámara por defecto está en proyección en perspectiva.

- **ViewAR::ViewAR()**: Destructor.
- **ViewAR::RequestAddObjectToBackground(ObjectRepresentation* object)**: Agrega un objeto a la capa del fondo. En particular se utiliza ara agregar una instancia de `VideoFrameRepresentation`.
- **ViewAR::SetBackgroundSize(int width, int height)**: Se configura la cámara del video para que ocupe toda la ventana de visualización.

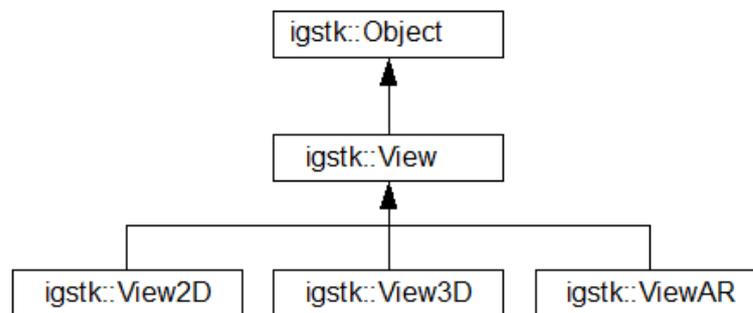


Figura 10.8: Herencia de la clase ViewAR

Utilización de la clase Se utiliza igual que las clases `igstk::View2D` y `igstk::View3D`. La única diferencia radica en que ésta permite agregar video en el fondo de la imagen. Para ello alcanza con agregar la representación del video utilizando la función `RequestAddObjectToBackground()`. Previamente se debe configurar el tamaño del video para que ocupe toda la ventana utilizando la función `SetBackgroundSize()`.

Configuración de la Cámara Virtual Para configurar la cámara virtual se deben seguir los siguientes pasos:

- Se lee de archivo los parámetros de calibración del microscopio (intrínsecos y extrínsecos).
- Se asocia la matriz de parámetros extrínsecos como matriz de calibración del `trackerTool` adherido al microscopio.
- Se configura la matriz de proyección de la cámara a partir de los parámetros intrínsecos.
- Se ubica a la cámara en el origen, mirando hacia los z positivos, con el eje y hacia abajo (posición de calibración).

Capítulo 11

Conclusiones y trabajo a futuro

Este capítulo resume las conclusiones que han sido expuestas en el transcurso de esta documentación. Para promover y facilitar la continuación y mejora de este proyecto también se sugieren líneas de trabajo a futuro.

A continuación se presentan las conclusiones y propuestas de trabajo, organizadas en tres secciones que cubren las siguientes áreas:

- Calibración.
- Realidad aumentada.
- Reconstrucción estéreo.

11.1. Calibración

Si bien en las diferentes visitas a los Hospitales se calibró la óptica de los tres microscopios, las pruebas para la evaluación fueron hechas sobre el microscopio del Hospital de Tacuarembó. Los principales problemas que presentaron los microscopios del Hospital de Clínicas y del Hospital Maciel fueron de tipo mecánico relacionados con el acople entre la óptica y las cámaras. El desgaste de los adaptadores que une ambas partes impide que las cámaras queden firmemente adheridas al microscopio, lo que claramente tiene consecuencias negativas en esta aplicación. Para el correcto funcionamiento del sistema es indispensable que el conjunto adaptador cámara esté rígidamente adherido al microscopio. Si existen movimientos relativos entre la cámara y el microscopio no es posible desarrollar realidad aumentada. Por otro lado, los adaptadores reducen el área que se proyecta sobre la imagen: si se compara la vista que se tiene desde el microscopio con la que se observa en la imagen, se puede comprobar que en esta última se han perdido las partes más exteriores. Esto se da en los tres microscopios, pero es especialmente notorio en el del Hospital de Clínicas. La Figura 11.1 es una imagen de uno de los primeros patrones de calibración, tomada con el microscopio y la cámara de este hospital. Puede verse cómo la parte superior queda recortada, perdiéndose una parte del campo de visión del microscopio.

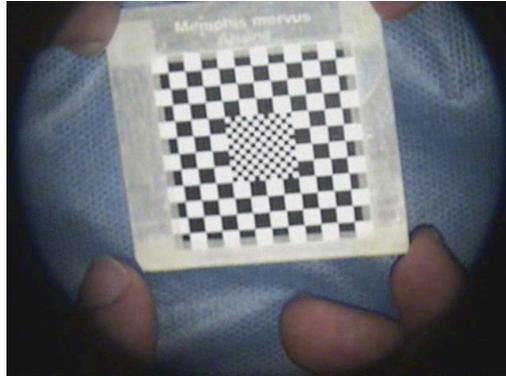


Figura 11.1: Imagen de un patrón de calibración tomada con el microscopio y cámara del Hospital de Clínicas.

De la Figura 11.1 puede hacerse otra observación importante. Fácilmente se comprueba que la imagen proveniente del microscopio no está centrada respecto a la imagen que se extrae de la cámara, ya que el centro del círculo determinado por la región iluminada no coincide con el centro de la imagen. En consecuencia, en estas condiciones la suposición de que el punto principal se encuentra en el centro de la imagen no es válida.

En cuanto a los resultados obtenidos tanto con el microscopio del Hospital de Tacuarembó como con las cámaras del laboratorio, es importante analizar qué factores que influyen sobre el error RMS en la reproyección de puntos. Lo que se busca en principio es evaluar *únicamente* el método de calibración que se está usando, esto es, qué tan bueno es el ajuste de los parámetros y cuál es el error que se comete al modelar la óptica del microscopio como una cámara pinhole sin tener en cuenta la distorsión y fijando el punto principal en el centro de la imagen. Sin embargo, existen otros factores que influyen sobre los resultados:

- Error de detección del localizador. El algoritmo de evaluación implementado requiere conocer la posición espacial de los puntos de control (esquinas del patrón) y un error en la ubicación de estos puntos influye sobre el resultado final. Dado que la posición de los puntos de control se conoce a través de la ubicación de los marcadores adheridos al patrón, el error cometido por el localizador al detectar los marcadores tendrá efecto sobre el error cometido en la reproyección. Puede encontrarse información sobre la precisión del localizador en la Subsección 3.2.1.
- Movimiento del patrón. Tal como se explicó en la Sección 5.7, el método de evaluación compara la posición de las esquinas detectadas sobre el damero con la posición que se obtiene luego de proyectar las coordenadas de su ubicación espacial a través de las matrices resultantes de la calibración. Es necesario entonces sincronizar la detección de los marcadores del patrón con la adquisición de la imagen sobre la que luego se detectarán las esquinas, para

que efectivamente se trate de puntos correspondientes. Sin embargo, entre los instantes en que se procesan una y otra tarea puede producirse un movimiento del patrón, lo cual tendrá influencia sobre la medida del error RMS. En la práctica se vio que esta influencia no es menor, especialmente dado que no se construyó un dispositivo que permitiera fijar el patrón en diferentes posiciones frente a la cámara, sino que el mismo era sostenido con la mano o en el mejor de los casos apoyado sobre la camilla dentro de la sala, lo cual tampoco asegura que el patrón permanezca inmóvil.

- Movimiento relativo entre el patrón o el microscopio y sus marcadores. El método de evaluación de calibración, y todos los algoritmos en general, asumen que tanto el patrón como el microscopio están rígidamente adheridos a sus marcadores asociados y esto es a grandes rasgos verdadero. Sin embargo, dada la construcción de los dispositivos, se pudo constatar en la práctica que luego de algunos minutos de uso pueden darse movimientos de los marcadores, lo cual claramente afecta el error de reproyección, ya sea porque la matriz H calculada deja de ser válida en el caso de que se muevan los marcadores del microscopio, o porque se está ubicando mal las esquinas en el espacio en caso de que se muevan los marcadores del patrón.

La combinación de estos factores se ve reflejada en los valores del error RMS presentados en las Tablas 5.11 a 5.14 correspondientes a la calibración de la cámara del laboratorio, y en las Tablas 5.16 a 5.19 para la calibración del microscopio. Si bien en la mayoría de los casos el error superó el pixel, en algunos se alcanzó una precisión subpixel, lo que muestra que al correr los algoritmos en condiciones óptimas éstos brindan buenos resultados. Por otro lado, las imágenes de las Figuras 5.7 a 5.10 para el caso del microscopio, dan una medida cualitativa del desempeño de los algoritmos, y si se tiene en cuenta la precisión requerida por la aplicación, los resultados obtenidos pueden considerarse aceptables para un número mayor de casos.

Todos los resultados presentados relativos al microscopio corresponden a calibraciones de su óptica realizadas con el patrón tipo damero plano dado que las pruebas hechas con el patrón cúbico no dieron buenos resultados durante el trabajo en la sala de operaciones. En una gran cantidad de casos no se consiguió una buena detección de las esquinas, pero aún cuando éstas se detectaron correctamente, tanto la matriz de parámetros intrínsecos A como la transformación H entre el sistema de coordenadas del mundo y de la cámara adquirieron valores muy alejados de lo esperado; aún no se ha podido determinar el motivo por el cual esto sucede.

El proceso de calibración, cuando exitoso, demostró funcionar correctamente. Sin embargo, no se logró determinar un procedimiento que funcione para el cien por ciento de los intentos. Sí se determinaron empíricamente una serie de factores que influyen en el éxito o no de la misma; estos son:

- Movimiento del patrón: es importante que el patrón permanezca quieto en el momento de sacar las fotos para que el cuadro obtenido coincida exactamente con la posición determinada por el sensor Polaris.

- **Distribución de los puntos:** deben cubrir la mayor parte posible del campo de visión del microscopio.

Como comentario final, cabe mencionar que en la práctica con el microscopio se comprobó que los mejores resultados se tienen cuando se trabaja con aumentos bajos. Para aumentos mayores a 3 o 4 no se logró una calibración aceptable. Si bien la bibliografía consultada sobre aplicaciones con microscopios adopta el modelo de cámara pinhole más coeficientes de distorsión, se plantea la duda de si este modelo sigue siendo adecuado al incrementar el aumento del microscopio o si sería conveniente estudiar otro que ajuste mejor a el comportamiento físico del sistema óptico.

11.2. Realidad aumentada

Se desarrolló e implementó un sistema de Realidad Aumentada utilizando el microscopio quirúrgico que se encuentra en el Hospital de Tacuarembó. Se probó y evaluó positivamente el sistema para valores de aumento menores a 3. Los errores obtenidos, luego de toda la cadena de transformaciones involucradas estuvieron en un entorno de 4 píxeles trabajando con una resolución de 640×480 . Nos parece que mejorando ciertos aspectos se puede disminuir considerablemente el error. En particular, sería deseable contar con estudios realizados a un cráneo nuevo ya que el actual se ha deteriorado; esto mejoraría considerablemente la etapa de registrado.

11.2.1. Trabajo para hacer en el futuro

Hubo algunos aspectos que no fueron estudiados en profundidad y que tal vez deban ser atacados en el futuro. Estos son:

- **Marcadores adheridos al microscopio** Para seguir al microscopio se utilizó una sola herramienta con cuatro marcadores. Si bien el localizador Polaris es muy preciso, entendemos que una mayor cantidad de marcadores, distribuidos de forma tal que su baricentro coincida (idealmente) con el centro óptico del microscopio, podría mejorar los resultados de la calibración. En particular, creemos que la transformación que ubica al centro óptico respecto a los marcadores, se haría más robusta frente a rotaciones del microscopio. Al colocar más marcadores también se podría mejorar la inmunidad al ruido.
- **Actualización de la calibración** La práctica ha demostrado que luego de un tiempo el sistema comienza a variar y los parámetros obtenidos durante la calibración dejan de ser buenos. Una razón para ello podría ser un pequeño movimiento de los marcadores adheridos al microscopio. Esto no debería sorprender ya que una pequeña variación en la posición de los mismos podría introducir un error significativo en la proyección de los puntos espaciales.

Este fenómeno hace dudar sobre si la calibración es un proceso que se puede realizar totalmente fuera de la sala. En principio debería serlo; de hecho

en las pruebas realizadas con cámaras domésticas los parámetros obtenidos funcionan, incluso con el correr de los días.

En [60] se propone un método de actualización de la calibración que asume que los parámetros intrínsecos de la cámara no cambian durante la operación y que los errores de proyección se deben a variaciones en los parámetros extrínsecos. Se propone un método de actualización de los parámetros extrínsecos que básicamente consiste en establecer una correspondencia entre puntos espaciales y puntos en la imagen durante la operación y a partir de ellos, manteniendo fijo los parámetros intrínsecos, encontrar la posición del centro óptico que minimice el error de reproyección.

- **Variaciones en el aumento y la distancia de trabajo** Entendemos que aún no se ha resuelto completamente el sistema para un aumento y una distancia de trabajo determinada por lo que parecería ser un tanto apresurado intentar resolverlo para todo el rango de estas variables. En general lo que se hace es calibrar el sistema para diferentes valores de aumento y distancia de trabajo y luego realizar una interpolación para aquellos valores en que no se cuenta con información [21, 74]. La diferencia entre los métodos radica en la forma de realizar la interpolación.

Cabe mencionar que en ninguno de los microscopios con los que se hicieron pruebas existía la posibilidad de extraer la información de aumento y distancia de trabajo en un momento dado, más allá de la lectura de un display.

Para tener una idea clara de cómo influye el error de cada una de las etapas, sería conveniente realizar una simulación numérica de todo el proceso. Así, por ejemplo, se podría agregar ruido en la detección de cada uno de los marcadores, calcular el error en la detección de cada herramienta, analizar cómo varía este error con la distribución de los marcadores en la herramienta, analizar la variación de los parámetros de cada transformación con el ruido, etc. Un análisis de estas características puede consultarse en [15].

11.3. Reconstrucción estéreo

Se realizó la reconstrucción estéreo de una superficie utilizando el microscopio del Hospital de Tacuarembó y el localizador, todo esto a partir de un desarrollo e implementación propia. Se realizaron pruebas reconstruyendo superficies con resultados satisfactorios teniendo en cuenta el corto período de tiempo en que se pudieron realizar las pruebas. Como resultado se obtuvo que para un aumento pequeño y una distancia de trabajo de aproximadamente 450 mm se alcanzaron resoluciones en la profundidad de 0.22 mm utilizando el algoritmo de búsqueda de correspondientes Block Matching; por otro lado los errores RMS cometidos estuvieron entre 1 y 2.5 mm.

Se considera que las pruebas realizadas no fueron suficientes como para asegurar estas mediciones y poder mejorar el proceso, es por esto que aún hay trabajo por

hacer para acreditar las medidas de error. Queda también pendiente la realización de la reconstrucción estéreo con el tipo de imágenes típicas de una operación dado que será fundamental saber si es posible encontrar correspondencias entre las imágenes para la textura de la superficie cerebral. En cuanto a los equipos, falta analizar que resultado se obtienen al utilizar dos cámara iguales, con tarjetas adquisidoras iguales, colocadas en los oculares del microscopio.

Apéndice A

Descripción del software implementado

A.1. Librerías utilizadas

A.1.1. OpenCV

Es una biblioteca libre de visión por computadora, escrita en los lenguajes C y C++, originalmente desarrollada por Intel [11]. Fundamentalmente se enfoca en el desarrollo de algoritmos de procesamiento de imágenes que corran en tiempo real. Su publicación se da bajo licencia BSD, lo que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

OpenCV es multiplataforma, existiendo versiones para Linux, Mac OS y Windows. Se puede descargar gratuitamente la última versión estable [47] o descargar un snapshot usando SVN.

Prerrequisitos En Windows se necesita contar con una instalación de DirectX SDK 8.0 o superior.

A.1.2. IGSTK

IGSTK (Image Guided Surgery Toolkit) [35, 22, 10] es una librería open-source escrita en C++ que provee los componentes básicos necesarios para desarrollar una aplicación de cirugía guiada por imágenes. Fue descrita en el Capítulo 10.

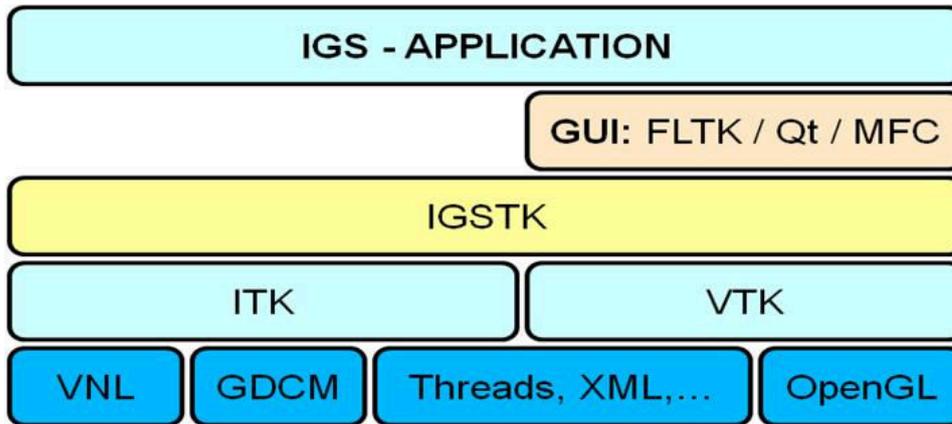


Figura A.1: Arquitectura utilizada por IGSTK.

A.2. Repositorio

Se desarrolló un software que sea capaz de implementar y evaluar las distintas etapas del proceso de agregado de realidad aumentada en un vídeo. Este software es sólo una plataforma de pruebas y puede resultar de utilidad para quienes se encuentren desarrollando aplicaciones que involucren calibración o reconstrucción estéreo. El código fuente desarrollado se encuentra en el repositorio: <https://iie.fing.edu.uy/Neuronavegador/neurosandbox/src/MIRACalibracion>

A.3. Instalación

Para compilar el código se necesita contar con:

- CMake 2.6 o superior
- IGSTK 4.2
- FLTK
- OpenCV

Dentro de la carpeta `MIRACalibracion` se encuentra el archivo `CMakeLists.txt` necesario para crear con CMake la solución (en Windows) que contiene los proyectos implementados. Todas las bibliotecas utilizadas son multi plataforma por lo que en principio podrían utilizarse con cualquier sistema operativo. Sin embargo, la captura de señal de vídeo no es independiente del sistema operativo y se resolvió sólo para el entorno Windows. En concreto, se asegura el funcionamiento de la solución para el sistema operativo Windows y el compilador Microsoft Windows C++ 9.0.

En primer lugar es necesario instalar CMake. CMake es el encargado de generar la solución de Visual Studio que contiene a todos los proyectos creados. La

información utilizada por CMake para crear la solución se encuentra en los archivos `CMakeLists.txt` que se encuentran dentro de los directorios homónimos a alguno de los proyectos creados (ver proyectos creados en Sección A.4). Allí se determinan los caminos a utilizar para enlazar correctamente las distintas librerías.

Luego se debe instalar IGSTK, para ello es necesario contar previamente con las librerías ITK VTK. La instalación de IGSTK ofrece como opcionales la utilización de las librerías FLTK y OpenCV; éstas también deben ser instaladas. Para instalar correctamente IGSTK 4.2 se recomienda descargar y leer el capítulo de instalación del libro de IGSTK.

A.3.1. Páginas de descarga

libro IGSTK: <http://www.igstk.org/IGSTK/help/documentation.html>

IGSTK: <http://www.igstk.org>

CMake: <http://www.cmake.org>

ITK: <http://www.itk.org/HTML/Download.htm>

VTK: <http://www.vtk.org/get-software.php>

FLTK: http://public.kitware.com/IGSTKWIKI/index.php/Download_IGSTK

OpenCV: <http://opencv.willowgarage.com/wiki>

A.4. Contenido de la solución

La solución que se crea utilizando CMake contiene los siguientes proyectos:

- `CalibracionDamero`: Obtiene los parámetros del patrón de calibración, tanto para el patrón plano como para el patrón cubo.
- `TestCalcularH`: Calibración del microscopio utilizando el patrón plano.
- `TestCalcularHvs2`: Calibración del microscopio utilizando el patrón plano, con datos desde archivo.
- `CalibracionConCubo`: Calibración del microscopio utilizando el patrón cubo.
- `RegistradoCabeza`: Registrado del cráneo o de un cubo.
- `HelloWorld`: Visualización completa del proceso de Realidad Aumentada.
- `StereoUnaCamara`: Adquisición de imágenes para la reconstrucción estéreo y cálculo de la calibración estéreo.

- **DisparidadyReconstruccion:** Triangulación de la superficie a partir de los datos obtenidos en *StereoUnaCamara*.
- **EvalCalib:** Estimación del error RMS de reproyección. Se utiliza para evaluar la calibración del microscopio.
- **evalStereo:** Estimación del error RMS del proceso completo de reconstrucción.

A.4.1. Aspectos generales de la Solución

Todos los proyectos forman parte de la misma Solución, por lo que comparten ciertas características en común. En particular se destacan tres

- Se utiliza un archivo en el que se guarda gran parte de la parametrización de todos los proyectos. Este archivo es *MiraCalibracion/paramteters.txt* y se descarga junto con la Solución. Los parámetros necesarios para cada proyecto se verán en el Manual de Usuario.
- Las definiciones de las herramientas (.rom) así también como las definiciones de las mallas (.msh) se encuentran en la carpeta *MiraCalibracion/RA_recargada/data/* y se descargan junto con la Solución.
- Para realizar el proceso de realidad aumentada completo (igual para el proceso de reconstrucción estéreo), los parámetros de salida de algunos proyectos sirven como entrada para otros. Para unificar, se reúnen los parámetros de entrada de todos los proyectos en una sola carpeta *MiraCalibracion/Input/*.

Glosario

puntero: es la herramienta puntero.

marcadoresCámara: es la herramienta del localizador Polaris que se adhiere al microscopio o a la cámara, dependiendo del lugar de trabajo.

marcadoresPatrón: es la herramienta del localizador Polaris que se adhiere al patrón de calibración 3D.

patrón: es la herramienta fabricada para la calibración del microscopio, puede ser el patrón plano o el patrón cubo.

microscopio: en el caso del laboratorio es una cámara.

A.4.2. CalibracionDamero

Este proyecto obtiene el movimiento rígido que existe entre los marcadores Patrón y un sistema de coordenadas ideal ubicado en una posición conocida del patrón. Permite calibrar ambos patrones, tanto el patrón cubo así como el patrón damero. A continuación se describe de forma general el funcionamiento del proyecto, mientras que en la Figura A.2 se observa el diagrama de flujo del proyecto.

Entradas

- Archivo de configuración del proyecto, **parameters.txt**.
- Command Arguments: nombre del documento para el logger.
- Command Arguments: numero de puerto en el que IGSTK implementará la comunicación con el localizador.
- Línea de comando: en ejecución, se consulta al usuario cuál de los dos patrones utiliza.

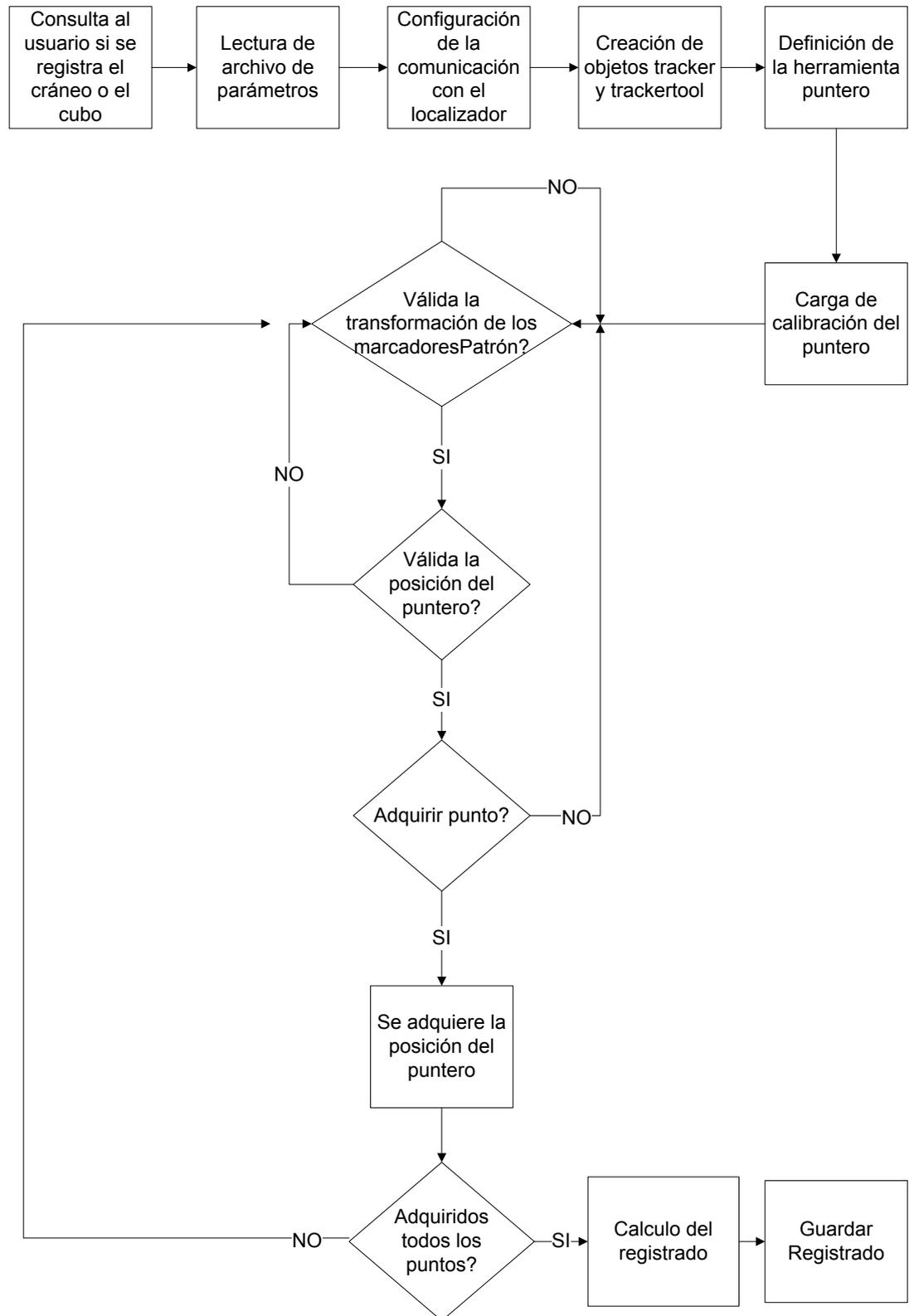
Salidas

- Movimiento rígido que ubica el sistema de referencia del patrón con los marcadores adheridos a él.

Flujo principal

- Se consulta al usuario, mediante línea de comando, qué tipo de patrón va a utilizar.
- Lectura del archivo de configuración de parámetros.
- Configuración de la comunicación con el localizador Polaris.
- Creación de los objetos `tracker` y `trackerTool`.
- Definición de las herramientas: puntero y marcadores. Los marcadores son los que están adheridos al patrón de calibración. En el caso del puntero se configura la calibración de pivot para poder obtener la posición de la punta.
- Se obtienen las posiciones de puntos específicos del patrón de calibración, en un orden determinado. La cantidad de puntos y la posición de los mismos depende del tipo de patrón utilizado.
- Para obtener estas posiciones se chequea si es válida la transformación asociada a los marcadores y de serlo se obtiene la posición del puntero respecto de los marcadores.

- Cuando se obtuvieron todas las coordenadas de los puntos, éstas se ponen en correspondencia con las coordenadas de los mismos puntos en el sistema de referencia del patrón. Las coordenadas en el sistema de referencia del patrón, para ambos tipos de patrón están prefijadas en el proyecto.
- Al poner en correspondencia los puntos en ambos sistemas de referencia se calcula el movimiento rígido que los relaciona.
- Por último se escribe en un archivo la matriz que representa la calibración del patrón.

Figura A.2: Diagrama de flujo del proyecto *CalibrarDamera*

A.4.3. TestCalcularH

Este proyecto resuelve la calibración del microscopio utilizando el patrón plano. Se calculan los parámetros intrínsecos de la cámara así como la transformación que ubica a la cámara respecto de los marcadores adheridos al microscopio. Este proyecto también puede utilizarse en el laboratorio, la diferencia está en que los marcadores adheridos al microscopio estarán adheridos a la cámara. Es importante destacar que este proyecto funciona en condiciones estáticas del microscopio (cámara). Su estructura puede resumirse en los pasos que se describen a continuación. En la Figura A.3 se muestra el diagrama de flujo del proyecto.

Entradas

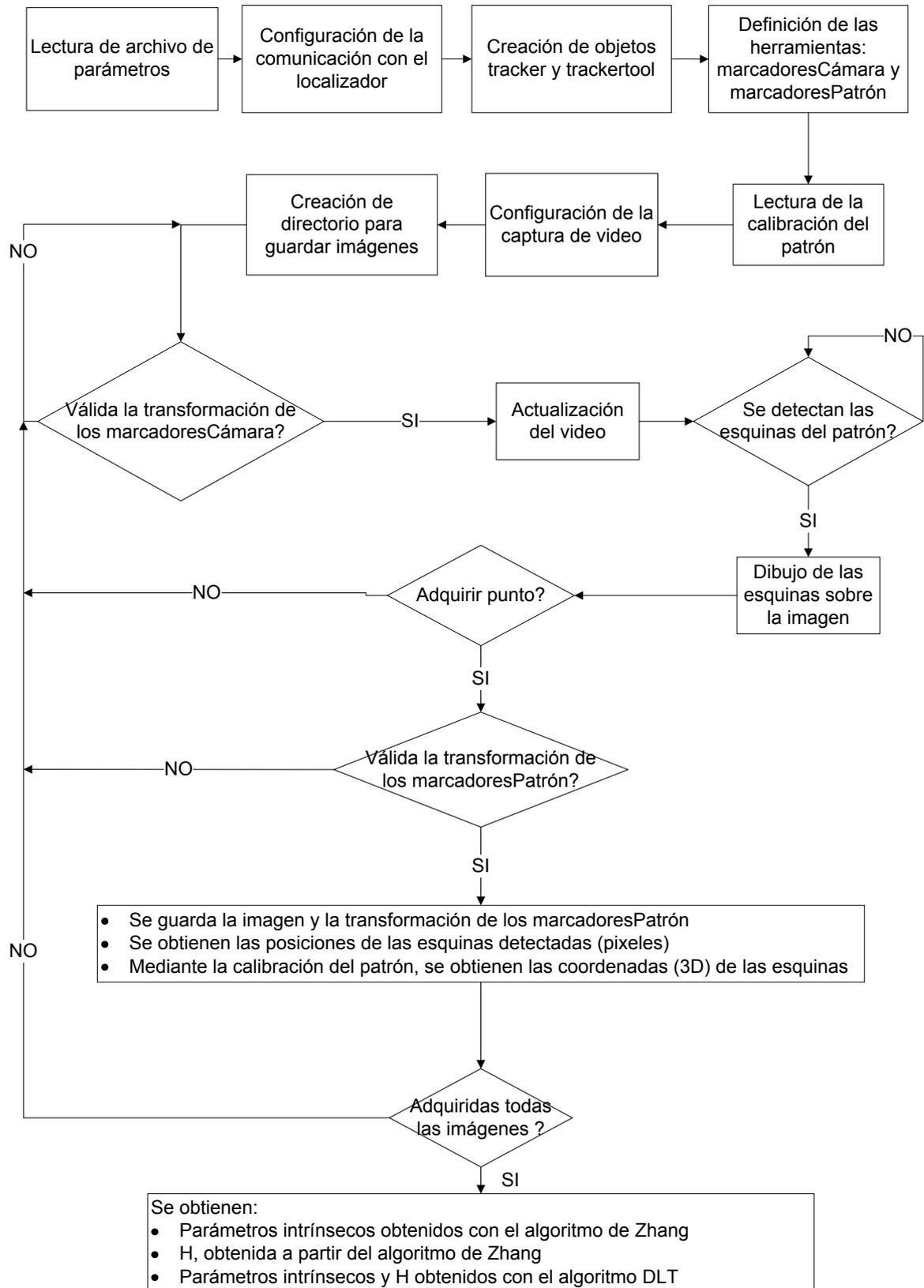
- Archivo de configuración del proyecto, **parameters.txt**.
- Command Arguments: nombre del documento para el logger.
- Command Arguments: numero de puerto en el que IGSTK implementará la comunicación con el localizador.
- Transformación de calibración del patrón plano.

Salidas

- Imágenes que se utilizaron para la calibración.
- Transformación de los marcadoresPatrón para cada uno de los instantes en que se adquirió una imagen.
- Transformación de los marcadoresCámara.
- Matriz de parámetros intrínsecos de la cámara, calculados a partir del método de Zhang.
- Matriz de parámetros intrínsecos de la cámara, calculados a partir del método DLT.
- Movimiento rígido que ubica la cámara respecto del sistema de referencia de los marcadores adheridos al microscopio, utilizando la matriz de parámetros intrínsecos calculados con el método de Zhang.
- Movimiento rígido que ubica la cámara respecto del sistema de referencia de los marcadores adheridos al microscopio, utilizando el método DLT.

Flujo principal

- Lectura del archivo de configuración de parámetros.
- Configuración de la comunicación con el localizador Polaris.
- Creación de los objetos `tracker` y `trackerTool`.
- Definición de las herramientas: `marcadoresCámara` y `marcadoresPatrón`.
- Lectura de la transformación de calibración del patrón.
- Configuración de parámetros para la captura de video.
- Definición del directorio de escritura de los datos.
- Mientras no se hayan tomado la cantidad de imágenes especificadas en el archivo de parámetros se ejecuta el siguiente loop.
 - Si es válida la transformación de los `marcadoresCámara` se actualiza la imagen y se ejecuta el algoritmo de búsqueda de esquinas.
 - Si se detectan las esquinas, se dibujan sobre la imagen.
 - Espera a que el usuario indique el momento de capturar la imagen.
 - Cuando el usuario da la orden de adquirir se guarda la imagen en archivo.
 - Si se detectan las esquinas y es válida la transformación de los `marcadoresPatrón`, se obtiene la posición de las esquinas del patrón respecto del localizador. Para esto es necesario contar con la calibración del patrón.
 - Se guarda la transformación de los `marcadoresPatrón` en archivo.
- Una vez adquiridas las posiciones para todas las imágenes se guarda en archivo la transformación de los `marcadoresCámara`.
- Se calculan los parámetros intrínsecos utilizando el método de Zhang.
- Con los parámetros intrínsecos, se calcula la transformación que ubica la cámara respecto de los marcadores adheridos al microscopio (H).
- Utilizado los intrínsecos calculados con el método de Zhang como semilla, se calculan nuevamente los parámetros intrínsecos y H , pero ahora utilizando el método DLT.
- Se guardan las matrices de parámetros intrínsecos y las transformaciones H calculadas con los dos métodos.

Figura A.3: Diagrama de flujo del proyecto *TestCalcularH*

A.4.4. TestCalcularHvs2

Este proyecto tiene la misma utilidad que *TestCalcularH*, solo difiere en que en este caso se utiliza información desde archivo en vez de ser adquirida en el momento. Esta información es la que se guarda cuando se ejecuta *TestCalcularH* y contiene: Imágenes, transformaciones de los marcadores Patrón para cada una de las imágenes y la transformación de los marcadores Cámara.

Entradas

- Archivo de configuración del proyecto, **parameters.txt**.
- Command Arguments: nombre del documento para el logger.
- Command Arguments: numero de puerto en el que IGSTK implementará la comunicación con el localizador.
- Command Arguments: archivo que contiene la dirección de las imágenes que se utilizan para calibrar. En el mismo directorio en el que están las imágenes deben estar incluidas: las transformaciones de los marcadores Patrón para cada imagen y la transformación de los marcadores Cámara.
- Transformación de calibración del patrón plano.

Salidas

- Matriz de parámetros intrínsecos de la cámara, calculados a partir del método de Zhang.
- Matriz de parámetros intrínsecos de la cámara, calculados a partir del método DLT.
- Movimiento rígido que ubica la cámara respecto del sistema de referencia de los marcadores adheridos al microscopio, utilizando la matriz de parámetros intrínsecos calculados con el método de Zhang.
- Movimiento rígido que ubica la cámara respecto del sistema de referencia de los marcadores adheridos al microscopio, utilizando el método DLT.

A.4.5. CalibracionConCubo

Este proyecto resuelve la calibración del microscopio utilizando el patrón cubo. Se calculan los parámetros intrínsecos de la cámara así como la transformación que ubica a la cámara respecto de los marcadores adheridos al microscopio. Este proyecto también puede utilizarse en el laboratorio, la diferencia está en que los marcadores adheridos al microscopio estarán adheridos a la cámara. Su estructura puede resumirse en los bloques que se describen a continuación, mientras que en la Figura A.4 se observa el diagrama de flujo.

Entradas

- Archivo de configuración del proyecto, **parameters.txt**.
- Command Arguments: nombre del documento para el logger.
- Command Arguments: numero de puerto en el que IGSTK implementará la comunicación con el localizador.
- Transformación de calibración del patrón cubo.
- Dentro del código: Número de imágenes a tomar del patrón.

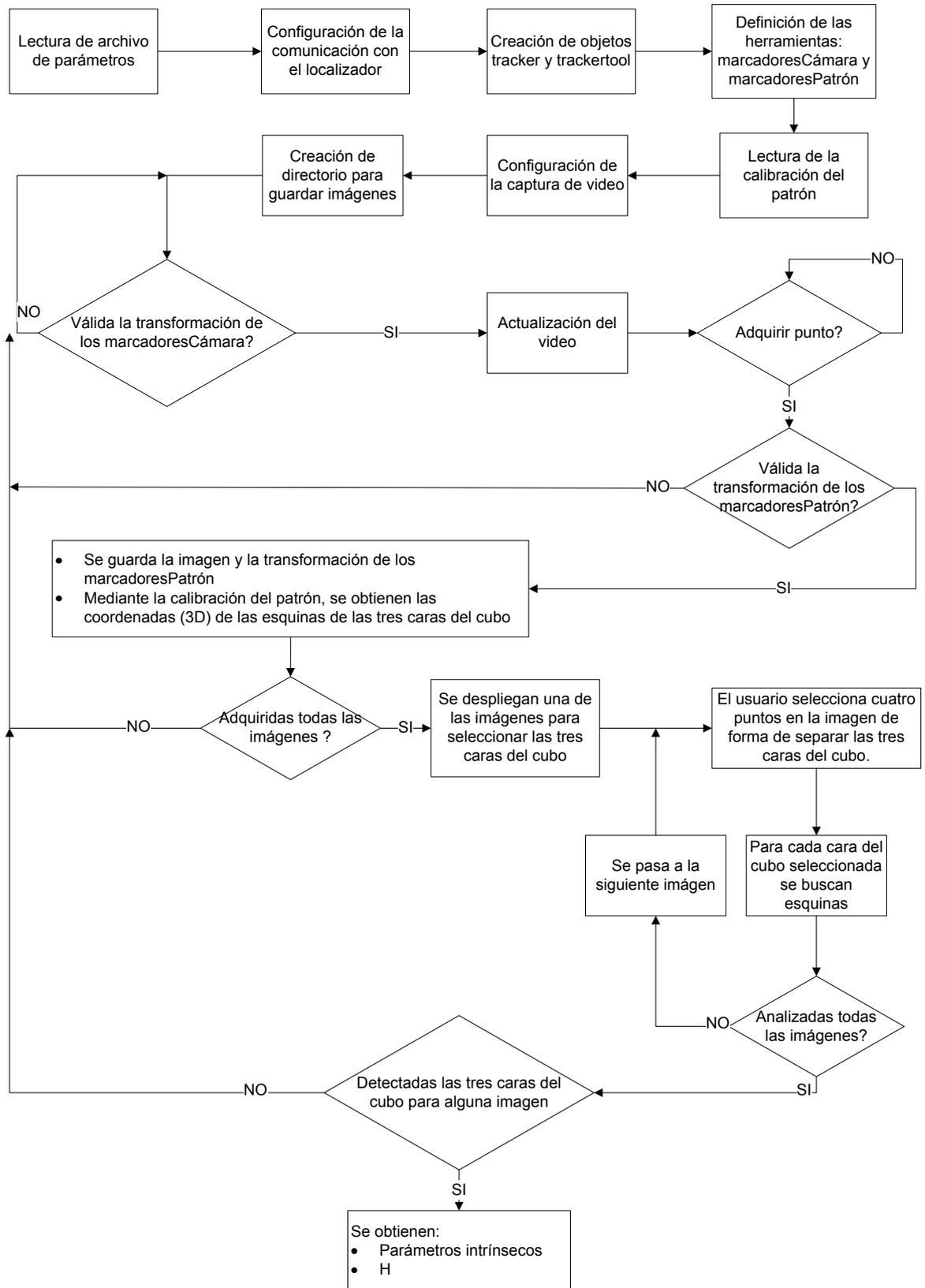
Salidas

- Matriz de parámetros intrínsecos de la cámara, calculados a partir del método DLT.
- Movimiento rígido que ubica la cámara respecto del sistema de referencia de los marcadores adheridos al microscopio, utilizando el método DLT.

Flujo principal

- Lectura del archivo de configuración de parámetros.
- Configuración de la comunicación con el localizador Polaris.
- Creación de los objetos tracker y trackerTool.
- Definición de las herramientas: marcadoresCámara y marcadoresPatrón. marcadoresCámara son los marcadores que están adheridos al microscopio mientras que marcadoresPatrón son los marcadores adheridos al patrón de calibración.
- Lectura de la transformación de calibración del patrón.
- Configuración de parámetros para la captura de video.
- Definición del directorio de escritura de los datos.
- Mientras no se hayan tomado la cantidad de imágenes especificadas se ejecuta el siguiente loop.
 - Si es válida la transformación de los marcadoresCámara se actualiza la imagen.
 - Espera a que el usuario indique el momento de capturar la imagen.
 - Si es válida la transformación de los marcadoresPatrón y el usuario indicó la captura, se guarda la imagen.

- Una vez adquiridas todas las imágenes se muestran una a una y para cada una se realiza lo siguiente.
 - Se muestra en pantalla.
 - El usuario debe marcar cuatro puntos en la imagen para separar una a una las tres caras del cubo.
 - Para cada cara del cubo se buscan las esquinas.
 - Para buscar las esquinas se utilizan tres niveles de umbralización en la binarización.
 - Mientras queden imágenes por analizar se repite lo anterior, de lo contrario se pasa a lo siguiente.
- Si se logró detectar las esquinas de las tres caras para al menos una imagen se continua, de lo contrario se vuelve al paso de adquisición de imágenes.
- Utilizando el método DLT se calcula la matriz de parámetros intrínsecos y H . Como semilla se utiliza el resultado de calibrar el microscopio con el proyecto *TestCalcularH*.
- Se guardan los datos.

Figura A.4: Diagrama de flujo del proyecto *CalibracionConCubo*

A.4.6. RegistradoCabeza

Con este proyecto se calcula el registrado del estudio tomográfico del cráneo que se utilizó como modelo de pruebas. Esto es, encontrar la ubicación del cráneo respecto del localizador para luego calcular la el movimiento rígido que relaciona el sistema de coordenadas del estudio con el localizador. También se puede realizar el registrado del cubo del patrón de calibración, calculando de la misma manera que con el cráneo, el movimiento rígido que relaciona el sistema de referencia del cubo con el localizador. El registrado del cubo tiene como fin analizar visualmente el sistema completo de transformaciones, esto porque es más sencillo ver la proyección de un cubo sobre la imagen que la proyección de una malla. En la Figura A.5 se muestra el diagrama de flujo del proyecto.

Entradas

- Archivo de configuración del proyecto, **parameters.txt**.
- Command Arguments: nombre del documento para el logger.
- Command Arguments: numero de puerto en el que IGSTK implementará la comunicación con el localizador.
- Línea de comando: en ejecución, se le consulta al usuario cuál de los dos registrados va a realizar; el del cráneo o el del cubo.

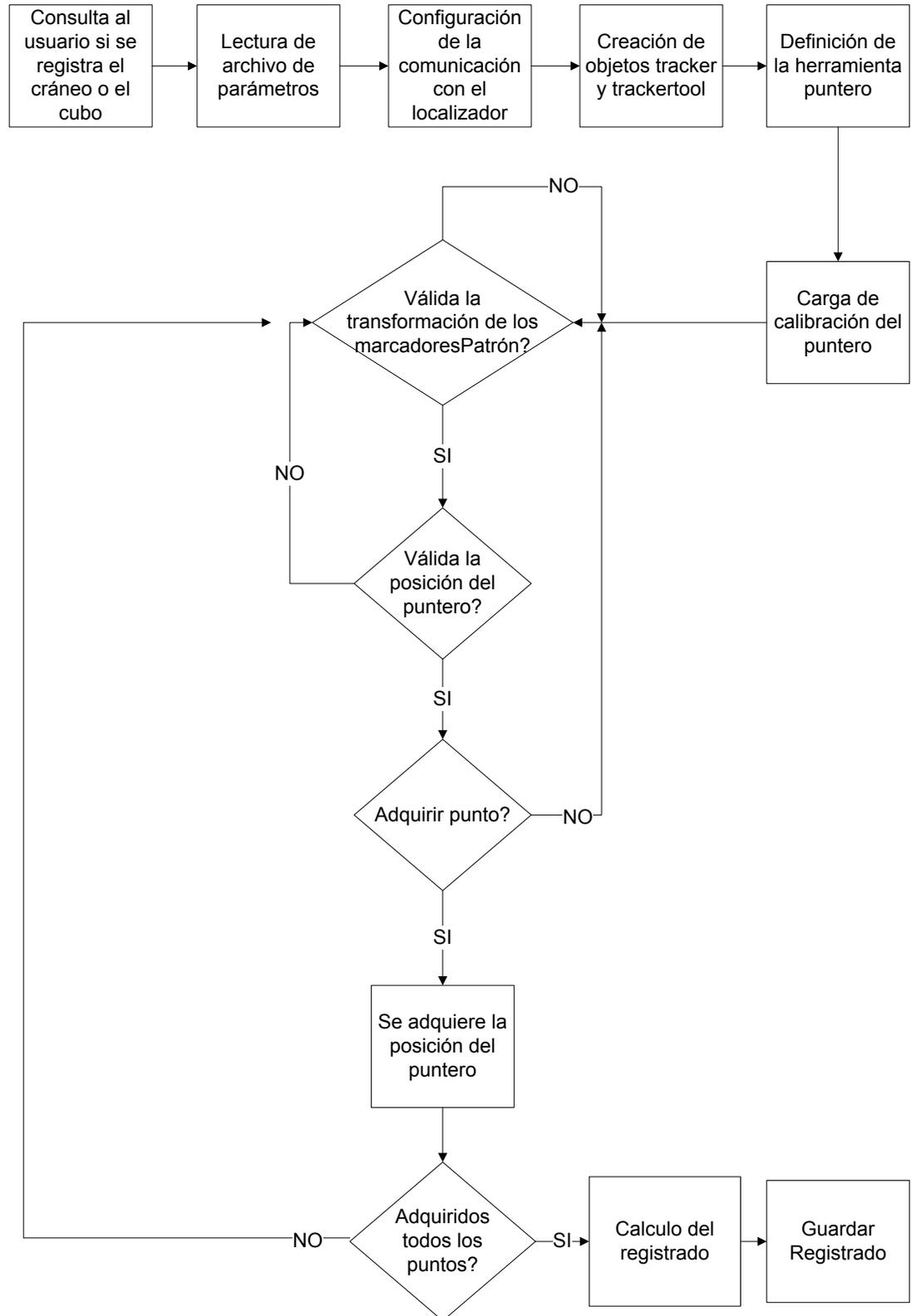
Salidas

- Movimiento rígido que ubica el sistema de referencia del cráneo (o del cubo) con el localizador.

Flujo principal

- Se le consulta al usuario, mediante línea de comando, que registrado desea realizar.
- Lectura del archivo de configuración de parámetros.
- Configuración de la comunicación con el localizador Polaris.
- Creación de los objetos tracker y trackerTool.
- Se define la herramienta puntero y se configura la calibración de pivot para poder obtener la posición de la punta.
- Se obtienen las posiciones de puntos específicos; en el caso del cráneo se adquieren las posiciones de los fiduciaros que tiene adheridos y en el caso del cubo se marcan cuatro vértices. En ambos casos interesa el orden en que se adquieren los puntos.

- Cuando el usuario lo indica, y si la posición del puntero es válida, se adquieren una a una las coordenadas de los puntos.
- Cuando se obtuvieron todas las coordenadas de los puntos, éstas se ponen en correspondencia con las coordenadas de los mismos puntos en el sistema del cráneo (cubo). Las coordenadas en el sistema de referencia del cubo y del cráneo ya están fijadas en el proyecto. En el caso del cráneo, estas coordenadas se obtienen cargando el estudio tomográfico en el programa Slicer y segmentando los puntos manualmente.
- Al poner en correspondencia los puntos en ambos sistemas de referencia se calcula el movimiento rígido que los relaciona.
- Por último se escribe en un archivo la matriz que representa el registrado.

Figura A.5: Diagrama de flujo del proyecto *RegistradoCabeza*

A.4.7. HelloWorld

En este proyecto se encarga de visualizar los resultados del proceso de Realidad Aumentada. Aquí se pueden observar los resultados de todo el proceso, tanto para la calibración del microscopio como para el registrado. Para facilitar algunas opciones de visualización se decidió implementar una interfaz gráfica sencilla que será descrita en el manual de usuario. En la Figura A.6 se muestra el diagrama de flujo del proyecto.

Entradas

- Archivo de configuración del proyecto, **parameters.txt**.
- Command Arguments: nombre del documento para el logger.
- Command Arguments: numero de puerto en el que IGSTK implementará la comunicación con el localizador.
- Definición de herramientas (.rom): puntero, marcadores adheridos al microscopio.
- Parámetros de calibración del microscopio: matriz de parámetros intrínsecos de la cámara, H .
- Registrado del cráneo y/o registrado del cubo.
- Definición de mallas: distintas mallas segmentadas del estudio y la malla que representa al puntero.

Salidas

- No hay salidas, este proyecto es solo de visualización.

Flujo principal

- Lectura del archivo de configuración de parámetros.
- Configuración de la comunicación con el localizador Polaris.
- Creación de los objetos tracker y trackerTool.
- Definición de las herramientas: marcadoresCámara y puntero. En el caso del puntero se configura la calibración de pivot para poder obtener la posición de la punta.
- Se cargan todas las mallas que se utilizan. Entre ellas están las que se segmentaron del estudio utilizando el programa Slicer y la malla que representa al puntero, ésta última realizada en Sketchup 7.

- Carga de los parámetros de calibración: intrínsecos y H
- Configuración de parámetros para la captura de video.
- Creación de los objetos mallas.
- Ubicación a la cámara respecto del localizador.
- Carga del registrado del cráneo. Solo si se configura en el archivo de parámetros.
- Carga del registrado del cubo. Solo si se configura en el archivo de parámetros.
- Si hay registrado del cráneo, se configuran los parámetros para realizar Reslice.
- Se agregan las representaciones de los objetos a la escena.
- Loop
 - Si es válida la transformación de los marcadores Cámara se actualiza la imagen.
 - Si es valida la posición del puntero, este se dibuja sobre la escena.
 - Si está registrado el cráneo, se estarán dibujando sobre la escena las mallas que se hallan definido.
 - Si está registrado el cubo, se va a dibujar sobre la escena.
 - **Observación:** sólo se visualizarán los objetos antes mencionados si están en el campo de visión de la cámara.

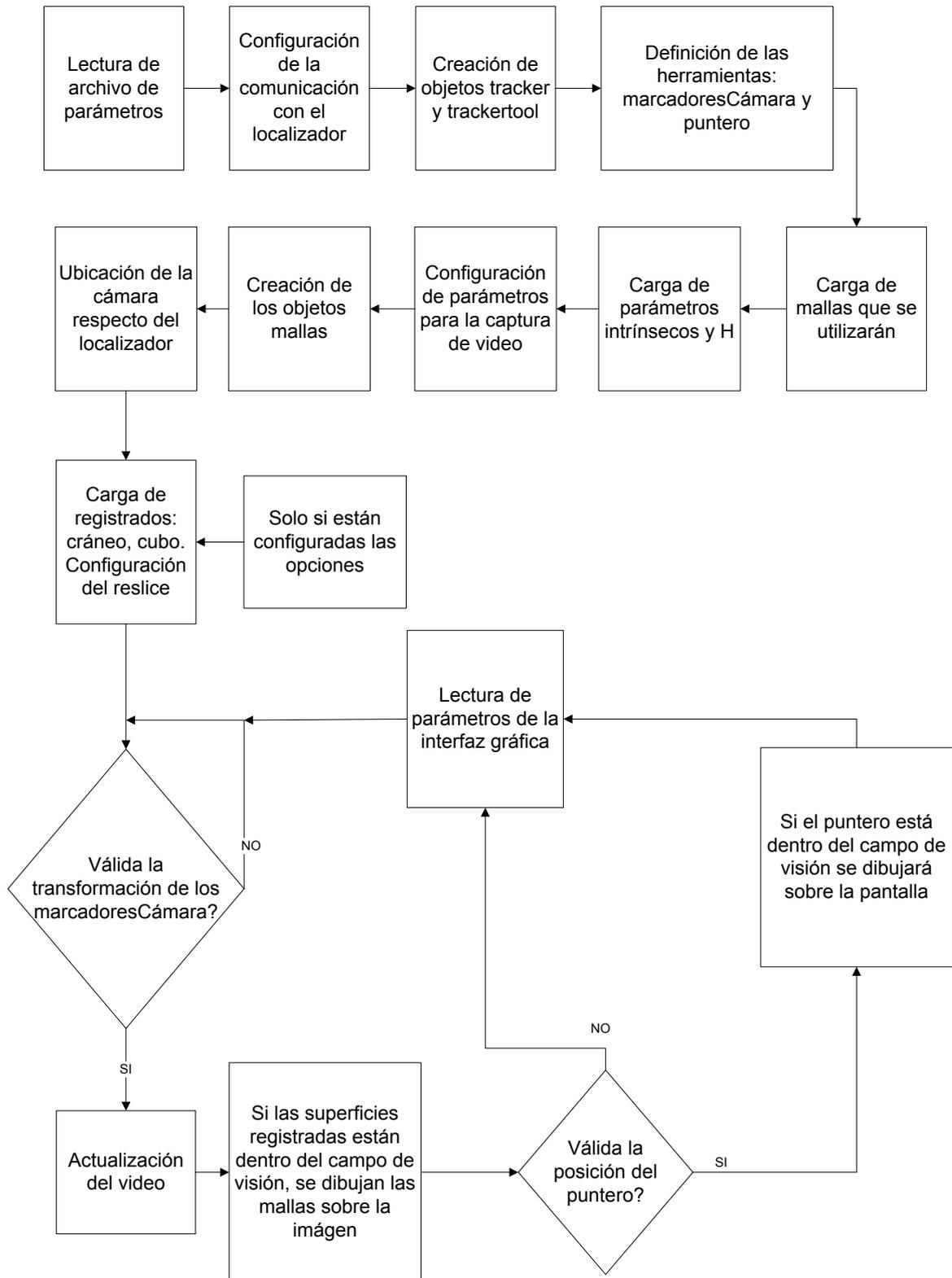
- Comandos desde la interfaz gráfica

Botón - Cargar H: se utiliza para cambiar los parámetros de calibración intrínsecos del microscopio y H . En el caso de que se haya utilizado el patrón plano para calibrar, por defecto se cargan los parámetros obtenidos a partir del método de Zhang; al presionar este botón se cambian los parámetros por los calculados utilizando el método DLT. En el caso de utilizar el parámetro cubo para la calibración, este botón no tiene efecto.

Botones - Mostrar Mesh / Quitar Mesh: son los encargados de seleccionar si se muestran o no las mallas segmentadas del estudio del cráneo.

Botones - Mostrar Reslice / Quitar Reslice: son los encargados de seleccionar si se realiza o no el reslice de las mallas.

Observación: los botones de “Mostrar / Quitar” antes descritos solo intervienen en la visualización. Si desde los parámetros no se dice que hay registrado o no se selecciona la opción del reslice estos botones quedan sin función.

Figura A.6: Diagrama de flujo del proyecto *HelloWorld*

A.4.8. StereoUnaCamara

Este proyecto realiza la adquisición de las imágenes para realizar la reconstrucción estéreo y también calcula la calibración del par estéreo. Para adquirir las imágenes utiliza una cámara de video. Rectifica y guarda las imágenes, también guarda la matriz de reconstrucción Q para que de con esos datos el proyecto *DisparidadyReconstruccion* realice la triangulación. Para la evaluación del proceso de reconstrucción se obtienen con el puntero, las coordenadas de puntos de la superficie respecto de la cámara; además se segmentan también las coordenadas de esos puntos en ambas imágenes rectificadas. En la Figura A.7 se muestra el diagrama de flujo del proyecto.

Entradas

- Archivo de configuración del proyecto, **parameters.txt**.
- Command Arguments: nombre del documento para el logger.
- Command Arguments: numero de puerto en el que IGSTK implementará la comunicación con el localizador.
- Definición de herramientas (.rom): puntero, marcadores adheridos al microscopio.
- Parámetros de calibración del microscopio: matriz de parámetros intrínsecos de la cámara, H , registrado del cráneo, registrado del cubo.

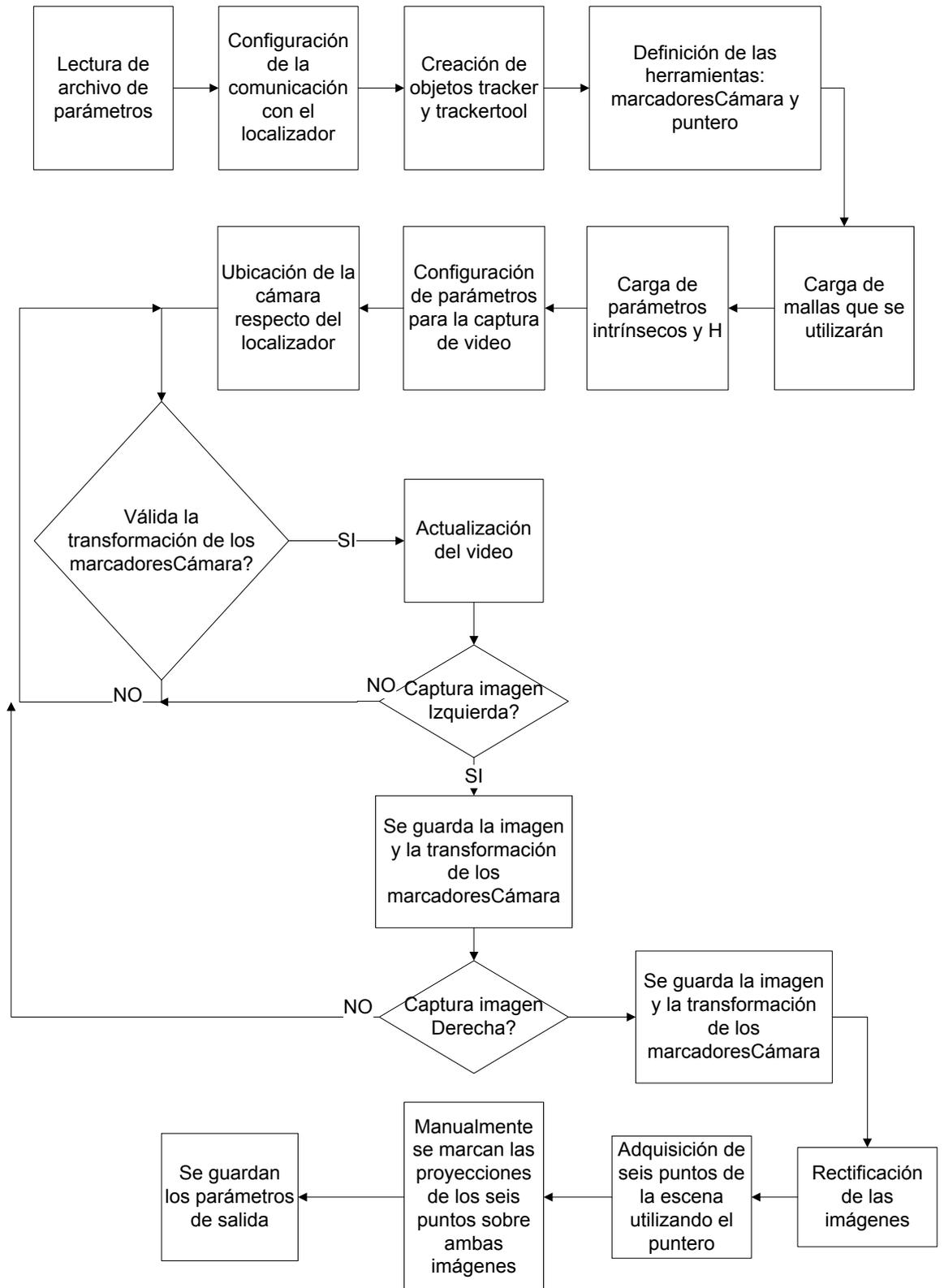
Salidas

- Par de imágenes rectificadas para realizar la reconstrucción.
- Matriz para realizar la triangulación Q .
- Puntos de la superficie adquiridos con el puntero.
- Puntos segmentados en ambas imágenes.

Flujo principal

- Lectura del archivo de configuración de parámetros.
- Configuración de la comunicación con el localizador Polaris.
- Creación de los objetos tracker y trackerTool.
- Definición de las herramientas: marcadoresCámara y puntero. En el caso del puntero se configura la calibración de pivot para poder obtener la posición de la punta.

- Carga de los parámetros de calibración: intrínsecos y H
- Configuración de parámetros para la captura de video.
- Ubicación de la cámara respecto del localizador.
- Loop
- Si es válida la transformación de los marcadoresCámara se actualiza la imagen.
- Espera hasta que el usuario indique la captura de la imagen izquierda. Para poder realizar la captura de la imagen, la transformación de los marcadoresCámara debe ser válida.
- Cuando se captura la imagen izquierda se adquiere la transformación de los marcadoresCámara.
- Espera hasta que el usuario indique la captura de la imagen derecha. Para poder realizar la captura de la imagen, la transformación de los marcadoresCámara debe ser válida.
- Cuando se captura la imagen derecha se adquiere la transformación de los marcadoresCámara.
- A partir de las transformaciones de los marcadoresCámara y utilizando los parámetros de calibración (matriz de parámetros intrínsecos y H) se calcula el movimiento rígido que lleva desde una de las posiciones de la cámara a la otra referido a la posición izquierda. Esto es la calibración del par estéreo.
- Rectificación de las imágenes.
- El usuario debe adquirir con el puntero las posiciones de puntos de la superficie. Estos puntos deben ser visibles en las imágenes rectificadas.
- El usuario debe marcar los puntos antes adquiridos con el puntero en ambas imágenes.
- Cálculo de la matriz Q .
- Guardado de parámetros de salida.

Figura A.7: Diagrama de flujo del proyecto *StereoUnaCamara*

A.4.9. Disparidad y Reconstrucción

Este proyecto se encarga de realizar la triangulación. Para esto toma las imágenes rectificadas, calcula los mapas de disparidad y por último, a partir de una región de la imagen determinada por el usuario, realiza la triangulación. El resultado de la triangulación se guarda tanto en un archivo (.txt) como en un archivo (.obj [71]). El diagrama de flujo del proyecto se muestra en la Figura A.8.

Entradas

- Archivo de configuración del proyecto, **parameters.txt**.
- Par de imágenes rectificadas para realizar la reconstrucción.
- Matriz para realizar la reconstrucción Q .
- Dentro del código: banderas para calcular los mapas de disparidad utilizando Block Matching y/o Graph Cut.

Salidas

- Puntos triangulados en archivo .txt, utilizando Block Matching y/o Graph Cut.
- Puntos triangulados en archivo .obj, utilizando Block Matching y/o Graph Cut.
- Puntos triangulados para evaluar la reconstrucción en archivo .txt.

Flujo principal

- Lectura del archivo de configuración de parámetros.
- Carga de la matriz de triangulación Q .
- Definición del directorio de escritura de los datos.
- Carga de las imágenes rectificadas.
- Cálculo de la disparidad mínima y de número de disparidades. Para esto se muestran ambas imágenes y el usuario debe marcar en ambas, un punto lejano y otro cercano a la cámara.
- Cálculo de mapas de disparidad. Se ejecutan ambos métodos o sólo el que esté seleccionado.
- Se muestran los mapas de disparidad para ambos métodos y el usuario debe seleccionar para cada uno de ellos la región a reconstruir.
- Triangulación.

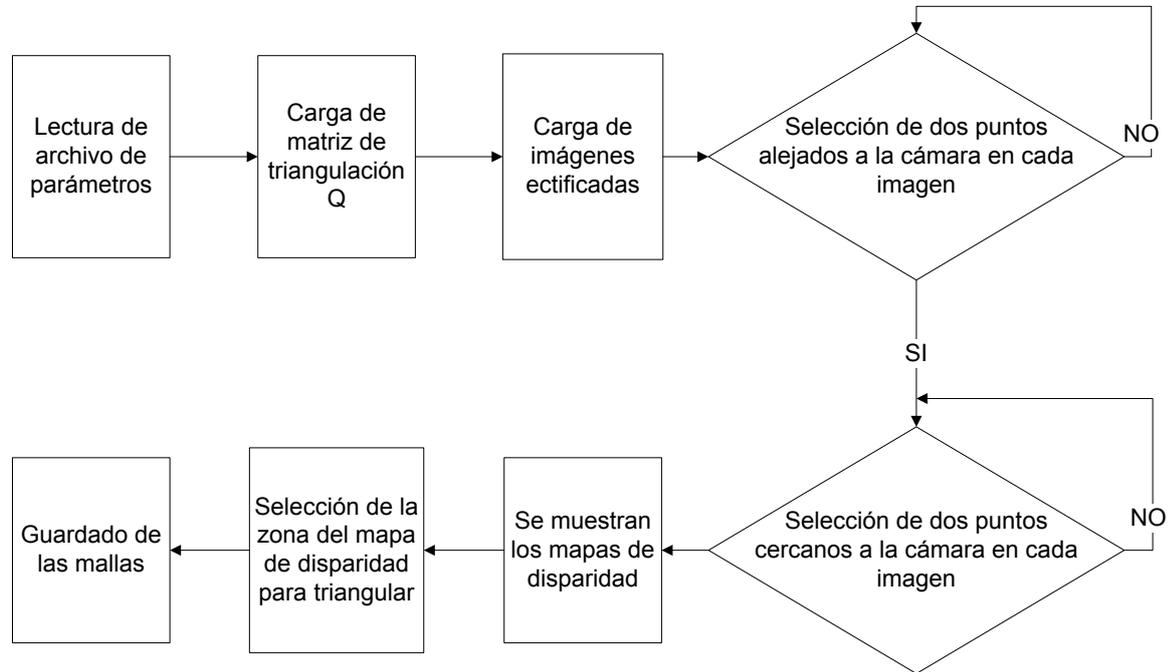


Figura A.8: Diagrama de flujo del proyecto *DisparidadYReconstruccion*

- Tanto si se seleccionó Block Matching o Grap Cut, para cada uno se guardan los datos en dos formatos:
 - Coordenadas 3D con nivel de grises en archivo .txt.
 - Coordenadas 3D en archivo .obj.

A.4.10. EvalCalib

Este proyecto evalúa la calibración del microscopio, esto es la matriz de parámetros intrínsecos y H . Para esto se toma una foto del patrón plano (previamente calibrado) y se detectan las esquinas del mismo. A partir de la calibración del patrón y de la transformación de sus marcadores se obtienen las posiciones en el espacio de las esquinas. Utilizando la calibración del microscopio se proyectan estos puntos sobre la imagen para compararlos con los detectados en la imagen.

Entradas

- Archivo de configuración del proyecto, **parameters.txt**.
- Command Arguments: nombre del documento para el logger.

- Command Arguments: numero de puerto en el que IGSTK implementará la comunicación con el localizador.
- Definición de herramientas (.rom): marcadores adheridos al patrón plano y marcadores adheridos al microscopio.
- Calibración del patrón plano.
- Parámetros de calibración del microscopio: matriz de parámetros intrínsecos de la cámara y H

Salidas

- Error RMS de proyección.

Flujo principal

- Lectura del archivo de configuración de parámetros.
- Configuración de la comunicación con el localizador Polaris.
- Creación de los objetos tracker y trackerTool.
- Definición de las herramientas: marcadoresCámara y marcadoresPatrón.
- Lectura de la transformación de calibración del patrón.
- Configuración de parámetros para la captura de video.
- Definición del directorio de escritura de los datos.
- Mientras no se haya tomado la imagen se ejecuta el siguiente loop.
 - Si es válida la transformación de los marcadoresCámara se actualiza la imagen y se ejecuta el algoritmo de búsqueda de esquinas.
 - Si se detectan las esquinas, se dibujan sobre la imagen.
 - Espera a que el usuario indique el momento de capturar la imagen.
 - Cuando el usuario lo indica, si se detectan las esquinas y es válida la transformación de los marcadoresPatrón, se obtiene la posición de las esquinas del patrón respecto del localizador. Para esto es necesario contar con la calibración del patrón.
 - Se obtiene la transformación de los marcadoresCámara.
- Utilizando la calibración del microscopio se proyecta la posición de las esquinas del patrón sobre la imagen.
- Cálculo del error RMS entre las esquinas proyectadas y las esquinas detectadas.

A.4.11. evalStereo

Este proyecto evalúa la reconstrucción estéreo. Para esto se triangulan puntos de la superficie segmentados manualmente y se compara con la posición obtenida utilizando el localizador. Por último se calcula el error RMS del sistema de reconstrucción estéreo.

Entradas

- Parámetros de calibración del microscopio: matriz de parámetros intrínsecos de la cámara y H
- Matriz para realizar la triangulación Q .
- Puntos de la superficie adquiridos con el puntero.
- Puntos segmentados en ambas imágenes.

Salidas

- Error RMS del sistema de reconstrucción estéreo.

Flujo principal

- Lectura del archivo de configuración de parámetros.
- Lectura de la matriz de triangulación Q .
- Lectura de puntos segmentados en ambas imágenes.
- Triangulación de los puntos segmentados.
- Lectura de puntos adquiridos con el localizador.
- Cálculo del error RMS entre ambos conjuntos de puntos.

Apéndice B

Calibración de patrones

Frecuentemente en las rutinas implementadas es necesario conocer las coordenadas espaciales de las esquinas del patrón de calibración referidas al sistema de coordenadas del localizador. Es posible extraer esta información marcando cada esquina individualmente con el puntero; sin embargo, este método consume mucho tiempo y es poco práctico, ya que es muy difícil marcar las esquinas con precisión debido al tamaño reducido del patrón de calibración. Además el apoyo del puntero sobre las esquinas termina deteriorándolas, lo que dificulta la detección de las mismas en la imagen.

El proyecto `CalibracionDamero`, cuya descripción puede consultarse en A.4.2, permite obtener las coordenadas espaciales de las esquinas del damero de una forma diferente, si se incorpora al patrón una de las herramientas del localizador. En la Figura B se muestran los diferentes patrones con la herramienta adherida. En todos los casos, ésta está unida rígidamente al patrón, con lo cual conociendo la posición de la herramienta respecto al localizador y habiendo calibrado el dispositivo es posible determinar las coordenadas espaciales de las esquinas sin necesidad de recorrerlas una a una con el puntero. El conjunto patrón-herramienta se considera calibrado si se conoce la transformación que vincula el sistema de coordenadas de la herramienta con el sistema asociado al patrón, como se muestra en la Figura B.2 para el caso de patrón plano. Hecho esto, se puede determinar en todo momento la posición de las esquinas respecto al localizador conociendo únicamente la posición de la herramienta.

Ya sea que se esté trabajando con un patrón plano o con uno tridimensional, el procedimiento a seguir para calibrarlo es el siguiente:

- A priori se eligen ciertos puntos de registrado cuya posición en la estructura del patrón es conocida con precisión. Estos puntos se expresan en términos de un sistema de referencia propio del patrón, y sus coordenadas en este sistema se denominan de aquí en adelante “coordenadas ideales”, $X_{ideales}$.
- Se obtiene la posición de los puntos de registrado respecto al localizador utilizando el puntero, lo que da lugar a las coordenadas $X_{localizador}$.

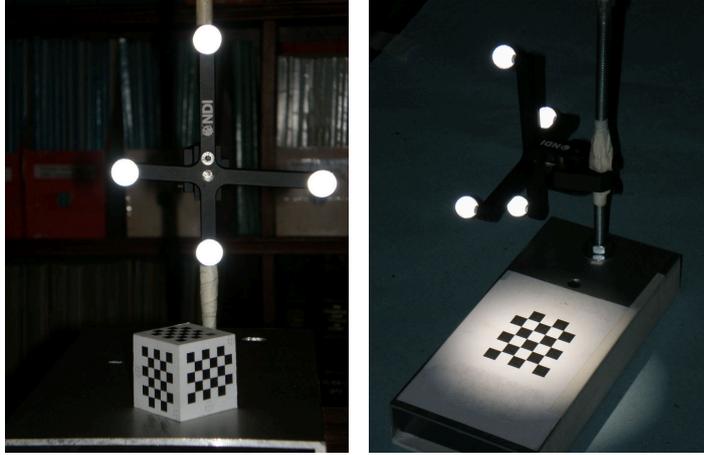


Figura B.1: Patrón plano y patrón tridimensional con herramientas adheridas.

- Se hace el registrado entre los puntos expresados en el sistema propio del patrón y en el sistema asociado al localizador, esto es, se calcula $T_{registrado}$ tal que

$$X_{localizador} = T_{registrado} \cdot X_{ideales}$$

Para hallar dicha transformación rígida se utiliza el método propuesto por Horn [7], implementado en IGSTK.

- Se extrae la transformación de los marcadores adheridos al patrón, $T_{herramienta}$, que lleva coordenadas del sistema de la herramienta en coordenadas respecto al localizador:

$$X_{localizador} = T_{herramienta} \cdot X_{herramienta}$$

- Se halla la transformación de calibración del patrón:

$$T_{calibración_damero} = T_{herramienta}^{-1} \cdot T_{registrado}$$

de manera que

$$X_{herramienta} = T_{calibración_damero} \cdot X_{ideales} \quad (\text{B.1})$$

como puede deducirse combinando las ecuaciones anteriores.

La Figura B.2 muestra los sistemas de coordenadas involucrados y las transformaciones que los vinculan.

Una vez que se ha encontrado la transformación $T_{calibración_damero}$, ésta permanece constante siempre y cuando no haya movimiento relativo entre el patrón y la herramienta adherida a él. Con esta suposición, en cualquier momento en que se quiera conocer la posición de las esquinas del patrón con respecto al localizador alcanzará con:

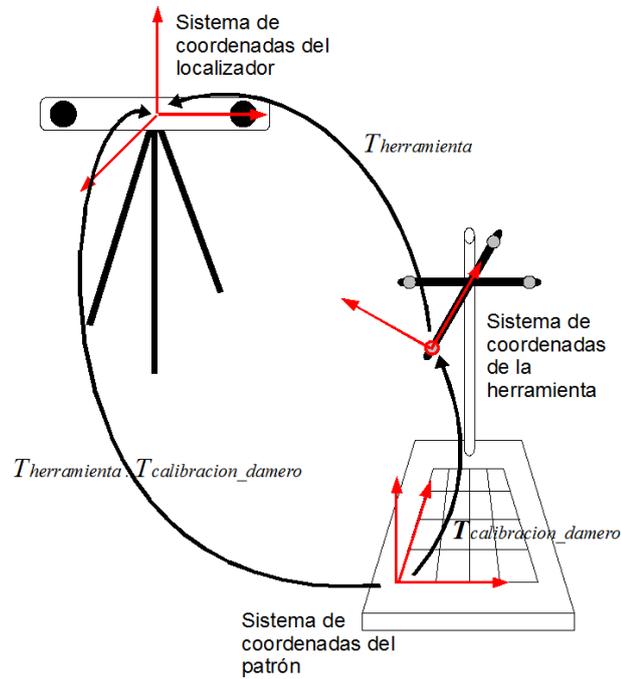


Figura B.2: Calibración de patrones.

- expresar las coordenadas de las esquinas en el sistema de referencia del patrón: $X_{esquinas_ideales}$.
- extraer la transformación de la herramienta del patrón en ese instante, $\hat{T}_{herramienta}$, tal que

$$X_{esquinas_localizador} = \hat{T}_{herramienta} \cdot X_{esquinas_herramienta} \quad (B.2)$$

Nótese que no necesariamente esta transformación es igual a la utilizada para el cálculo de $T_{calibración_damero}$, sino que el patrón puede haberse movido respecto al localizador pero aún así la transformación de calibración es válida.

La ubicación de las esquinas en el sistema de coordenadas del localizador estará dada por

$$X_{esquinas_localizador} = \hat{T}_{herramienta} \cdot T_{calibración_damero} \cdot X_{esquinas_ideales}$$

donde se ha hecho uso de la Ecuación B.1, que lleva puntos desde el sistema de coordenadas ideal al de la herramienta, y de la Ecuación B.2, que lleva puntos del sistema de la herramienta al del localizador.

Extracción de las coordenadas espaciales de los puntos de referencia

Patrón plano. En el caso del patrón plano, se consideran como puntos de referencia los vértices rodeados por un círculo que se muestran en la Figura B.3, los cuales deben ser marcados uno a uno con el puntero respetando el orden que se indica. Los dameros planos tienen asociado el sistema de coordenadas que también puede verse en la Figura B.3, cuyo origen se encuentra desplazado 15 mm en la dirección horizontal y 15 mm en la dirección vertical respecto al vértice superior izquierdo. Esto hace que se pueda cambiar el tamaño de las casillas del damero (lo cual es necesario para calibrar el microscopio con diferentes aumentos) sin cambiar la posición del sistema de referencia del damero, aunque sí cambia la posición de las esquinas (a excepción de la que coincide con el origen).

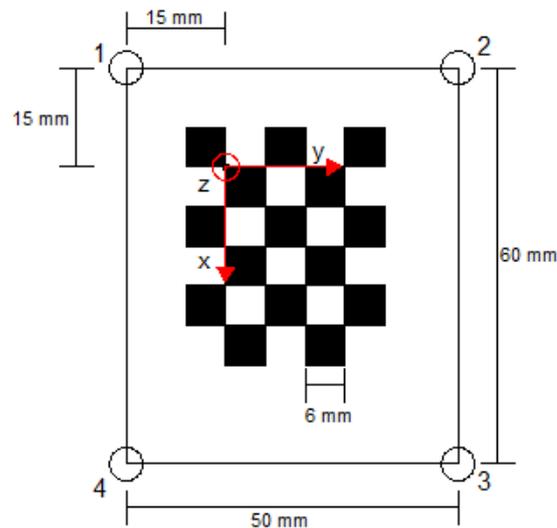


Figura B.3: Patrón plano de casillas de 6 mm y su sistema de coordenadas asociado.

Con el sistema de coordenadas elegido para el patrón, donde el plano en el que está contenido coincide con el plano $z = 0$, las coordenadas ideales de las esquinas resultan:

$$X_{ideales_{ij}} = (i \cdot d \quad j \cdot d \quad 0)^T$$

donde d es el tamaño de las casillas del patrón (6 mm en el ejemplo de la Figura B.3), i es el número de esquina en la dirección x y j en la dirección y .

Patrón tridimensional. Para calibrar el patrón tridimensional (cubo) se consideran los seis puntos de referencia que se muestran en la Figura B.4(a), ubicados dos en cada cara. Los mismos deben marcarse en el orden que se indica en la figura. El sistema de referencia asociado al cubo se muestra superpuesto a la imagen; como

puede observarse, su origen se encuentra en el vértice intersección de las tres caras visibles, con las x creciendo en el sentido de la cara derecha, las y creciendo hacia la cara izquierda y z creciendo hacia arriba.

En la Figura B.4(b) se muestra la ubicación de los puntos de registrado para el cubo de 3 mm de arista y casillas de 4 mm. Notar que en las tres caras se usa el mismo patrón tipo damero, rotado.

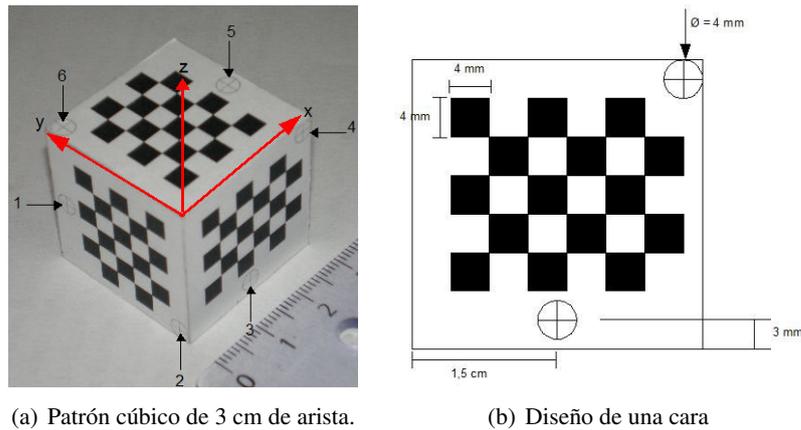


Figura B.4: Patrón cúbico de casillas de 4 mm y su sistema de coordenadas asociado.

Apéndice C

Bases matemáticas

C.1. Representación de puntos y líneas en coordenadas homogéneas

Representación homogénea de rectas

Una recta en el plano puede representarse por una ecuación del tipo $ax + by + c = 0$, donde diferentes combinaciones de a , b y c dan lugar a rectas diferentes. Por lo tanto, es posible representar la recta por el vector $(a, b, c)^T$. Pero la correspondencia entre rectas y vectores no es única, dado que cualquier recta $(ka)x + (kb)y + (kc) = 0$ es igual a $ax + by + c = 0$, para cualquier $k \neq 0$, y como consecuencia los vectores (a, b, c) y $k(a, b, c)$ representan la misma recta. Más aún, todos los vectores relacionados por un factor de escala se consideran equivalentes. Esta relación de equivalencia define clases de equivalencia que reúne a todos los vectores relacionados por un factor de escala, y a estas clases se les llama *vector homogéneo*. El conjunto de clases de equivalencias de vectores en $\mathbb{R}^3 - (0, 0, 0)^T$ constituye el espacio proyectivo \mathbb{P}^2 .

Representación homogénea de puntos

Un punto $x = (x, y)^T$ pertenece a la recta $r = (a, b, c)^T$ si y sólo si $ax + by + c = 0$. Esto puede expresarse en términos de un producto interno entre el vector que representa la recta, (a, b, c) y un vector que representa al punto, $(x, y, 1)$:

$$(x, y, 1)(a, b, c)^T = (x, y, 1)r = 0$$

De esta forma, un punto $(x, y) \in \mathbb{R}^2$ se representa como un vector de 3 componentes agregando una coordenada igual a 1 al final. Obsérvese que para cualquier constante $k \neq 0$ y para cualquier recta r , se cumple que $(kx, ky, k)r = 0$ si y sólo si $(x, y, 1)r = 0$, por lo que parece natural considerar al conjunto de vectores $k(x, y, 1)$, con $k \neq 0$, como una representación del punto $(x, y) \in \mathbb{R}^2$, por lo que, al igual que una recta, un punto se representa por un vector homogéneo. Un vector homogéneo arbitrario $(x_1, x_2, x_3)^T$ representa al punto $(x_1/x_3, x_2/x_3)$.

Resultado 1 *El punto x pertenece a la recta r si y sólo si $x^T r = 0$*

La demostración puede verse en los párrafos anteriores.

Resultado 2 *La intersección de dos líneas r y r' es el punto $x = r \times r'$*

Recordando que $r \cdot (r \times r') = r' \cdot (r \times r') = 0$, se puede ver que $r^T x = r'^T x = 0$. Por lo tanto, si se piensa en x como la representación de un punto, entonces por el resultado anterior este punto pertenece a ambas rectas, o sea, es su intersección.

Resultado 3 *La recta que une dos puntos x y x' es $r = x \times x'$*

Esta expresión puede deducirse utilizando un argumento análogo al del Resultado anterior: definiendo la línea $r = x \times x'$, puede verificarse que tanto x como x' pertenecen a ella.

Hasta ahora se dijo que un punto puede representarse en forma homogénea agregando una tercera coordenada igual a 1 y dada una representación $(x_1, x_2, x_3)^T$ de un punto pueden recuperarse sus coordenadas no homogéneas dividiendo las dos primeras entre x_3 . Pero ¿qué sucede si la tercera coordenada es 0? Considérense dos rectas paralelas $ax + by + c = 0$ y $ax + by + c' = 0$, representadas por los vectores homogéneos $r = (a, b, c)^T$ y $r' = (a, b, c')^T$. Si se calcula su intersección utilizando el Resultado 2 se tiene que $r \times r' = (c' - c)(b, -a, 0)^T$; dejando de lado el factor de escala, éste es el punto $(b, -a, 0)^T$. Si ahora se intenta encontrar la representación no homogénea del punto, se estaría dividiendo las primeras coordenadas entre 0, lo cual sugiere que las coordenadas son infinitamente grandes, o que el punto está en el infinito. Esto coincide con la idea de que “las rectas paralelas se juntan en el infinito”.

La utilidad de la representación mediante vectores homogéneos es evidente particularmente cuando se trabaja con las ecuaciones de proyección perspectiva en el plano de la imagen. Estas ecuaciones son **no lineales** cuando se expresan en coordenadas no homogéneas, pero se transforman en problemas **lineales** cuando se representan en forma homogénea, los cuales son numéricamente más fáciles de tratar.

C.1.1. Método de Bouguet: cálculo de la estimación inicial de los parámetros intrínsecos

En esta sección se mostrará cómo es posible obtener una estimación inicial de los parámetros intrínsecos y extrínsecos utilizando la geometría del espacio B-dual [8], en particular los conceptos de puntos y líneas de fuga. Dado que se busca una estimación para utilizar como semilla de un algoritmo iterativo que refine los valores de los parámetros, se dejará fuera del cálculo al punto principal, suponiendo que se encuentra en el centro de la imagen. En lo que sigue se utilizarán los conceptos de punto y línea de fuga y algunas propiedades de la geometría del espacio B-dual; estos temas pueden consultarse en [8].

Se considera un patrón de calibración plano tipo damero; suponiendo que la imagen no contiene ruido ni distorsión, la grilla de puntos del patrón puede

representarse por sus cuatro esquinas (si bien en la práctica se usan todos los puntos para lograr mayor inmunidad al ruido). Lo que se observa en la imagen es entonces una vista en perspectiva de un rectángulo de dimensiones conocidas $L \times W$. Más aún, sin perder generalidad, es posible asumir que el rectángulo es un cuadrado, dado que conociendo las dimensiones del rectángulo original siempre es posible encontrar una transformación perspectiva que convierta la vista perspectiva de un rectángulo en la vista perspectiva de un cuadrado.

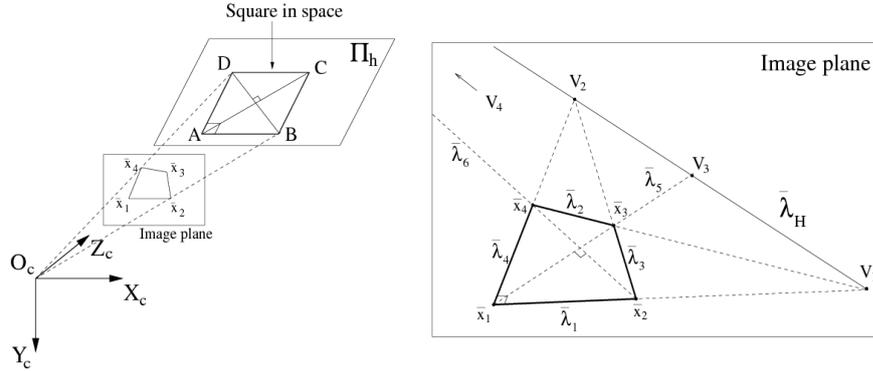


Figura C.1: Vista perspectiva del patrón de calibración, junto con los puntos de fuga y la recta de fuga asociada al plano del patrón.

Sea la vista perspectiva del cuadrado tal como se muestra en la Figura C.1, donde los puntos \bar{x}_1 , \bar{x}_2 , \bar{x}_3 y \bar{x}_4 son las coordenadas normalizadas de las esquinas en el plano de la imagen. Si $\bar{p}_i = [p_{x_i}, p_{y_i}, 1]^T$ son las coordenadas en píxeles de las esquinas, con el origen en el centro óptico, las coordenadas normalizadas \bar{x}_i se obtienen mediante la transformación lineal

$$\bar{x}_i = \begin{bmatrix} \frac{1}{f_x} & 0 & 0 \\ 0 & \frac{1}{f_y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \bar{p}_i = K \bar{p}_i \quad (\text{C.1})$$

En la Figura C.1 se observan las cuatro esquinas \bar{x}_i . A partir de las líneas $\bar{\lambda}_i$ se obtienen los puntos de fuga V_1 y V_2 , y es posible utilizar estos puntos para calcular la proyección $\bar{\lambda}_H$ de la línea de fuga asociada al plano Π_d que contiene al patrón [8] como

$$\bar{\lambda}_H \simeq V_1 \times V_2 \quad (\text{C.2})$$

donde

$$\begin{cases} \bar{\lambda}_1 \simeq \bar{x}_1 \times \bar{x}_2 & \rightarrow V_1 \simeq \bar{\lambda}_1 \times \bar{\lambda}_2 \\ \bar{\lambda}_2 \simeq \bar{x}_3 \times \bar{x}_4 & \\ \bar{\lambda}_3 \simeq \bar{x}_2 \times \bar{x}_3 & \rightarrow V_2 \simeq \bar{\lambda}_3 \times \bar{\lambda}_4 \\ \bar{\lambda}_4 \simeq \bar{x}_4 \times \bar{x}_1 & \end{cases} \quad (\text{C.3})$$

Utilizando la ecuación C.1 y recordando que si K^* es la matriz adjunta (o de cofactores) de K se cumple que

$$(Ku) \times (Kv) = K^*(u \times v)$$

entonces pueden reescribirse las Ecuaciones C.2 y C.3 como:

$$\bar{\lambda}_H \simeq K^* \bar{\lambda}_H^p \quad (\text{C.4})$$

y

$$\begin{cases} \bar{\lambda}_1 \simeq K^* \bar{\lambda}_1^p \\ \bar{\lambda}_2 \simeq K^* \bar{\lambda}_2^p \\ \bar{\lambda}_3 \simeq K^* \bar{\lambda}_3^p \\ \bar{\lambda}_4 \simeq K^* \bar{\lambda}_4^p \end{cases} \rightarrow \begin{cases} V_1 \simeq KV_1^p \\ V_2 \simeq KV_2^p \end{cases} \quad (\text{C.5})$$

donde $\bar{\lambda}_1, \bar{\lambda}_2, \bar{\lambda}_3, \bar{\lambda}_4, V_1^p, V_2^p$ y $\bar{\lambda}_H^p$ son las coordenadas en píxeles de los puntos y líneas sobre el plano de la imagen, calculadas directamente a partir de las coordenadas de \bar{p}_i . Al calcular los puntos de fuga a partir de las líneas $\bar{\lambda}_1$ y $\bar{\lambda}_2$ se hace uso del hecho que el patrón es rectangular. A continuación se impone la condición de ortogonalidad de los puntos de fuga:

$$V_1 \perp V_2 \Leftrightarrow (KV_1^p) \perp (KV_2^p) \Leftrightarrow (V_1^p)^T (K^T K) (V_2^p) = 0$$

lo que da una restricción escalar sobre los largos focales f_x y f_y :

$$\frac{a_1 a_2}{f_x^2} + \frac{b_1 b_2}{f_y^2} + c_1 c_2 = 0 \quad (\text{C.6})$$

donde a_1, a_2, b_1, b_2, c_1 y c_2 son las coordenadas en píxeles de los puntos de fuga conocidos: $V_1^p \simeq [a_1 \ b_1 \ c_1]$ y $V_2^p \simeq [a_2 \ b_2 \ c_2]$.

Finalmente, como se dijo antes, se asume además que el patrón es cuadrado, y esto implica que sus diagonales también son perpendiculares. Al igual que se hizo antes, esta condición se impone haciendo que los puntos de fuga asociados sean ortogonales, lo que da una segunda restricción sobre los largos focales:

$$\frac{a_3 a_4}{f_x^2} + \frac{b_3 b_4}{f_y^2} + c_3 c_4 = 0 \quad (\text{C.7})$$

con $V_3^p \simeq [a_3 \ b_3 \ c_3]$ y $V_4^p \simeq [a_4 \ b_4 \ c_4]$.

Escribiendo las Ecuaciones C.6 y C.7 en forma matricial, se tiene una ecuación lineal en $\bar{u} = [1/f_x^2, 1/f_y^2]^T$:

$$A\bar{u} = \bar{b} \quad (\text{C.8})$$

con $A = \begin{bmatrix} a_1 a_2 & b_1 b_2 \\ a_3 a_4 & b_3 b_4 \end{bmatrix}$ y $\bar{b} = - \begin{bmatrix} c_1 c_2 \\ c_3 c_4 \end{bmatrix}$. De esta ecuación se pueden extraer explícitamente los valores de f_x y f_y , siempre que la matriz A sea invertible:

$$f_x = \sqrt{\frac{a_1 a_2 b_3 b_4 - a_3 a_4 b_1 b_2}{b_1 b_1 c_3 c_4 - b_3 b_4 c_1 c_2}}$$

$$f_y = \sqrt{\frac{a_1 a_2 b_3 b_4 - a_3 a_4 b_1 b_2}{a_3 a_4 c_1 c_2 - a_1 a_2 c_3 c_4}}$$

Bibliografía

- [1] Carlzeiss, inc. Sitio web: <http://www.zeiss.com/>.
- [2] Northern digital inc. Sitio web: <http://www.ndigital.com/>.
- [3] *Polaris Tool Design Guide*. 2007.
- [4] OpenGL Redbook Capítulo 3. Frustum. Sitio web: <http://www.glprogramming.com/red/chapter03.html>.
- [5] KS Arun, TS Huang, and SD Blostein. Least-squares fitting of two 3-d point sets. *IEEE TRANS. PATTERN ANAL. MACH. INTELLIG.*, 9(5):698–700, 1987.
- [6] Daniel Bardsley. *A Practical Framework for 3D Reconstruction and Its Applications*. PhD thesis, School of Computer Science and IT University of Nottingham, 2008.
- [7] J. Y. Bouguet. Camera calibration toolbox for matlab. Sitio web: http://www.vision.caltech.edu/bouguetj/calib_doc/.
- [8] J.Y. Bouguet. *Visual methods for three-dimensional modeling*. PhD thesis, California Institute of Technology, 1999.
- [9] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.
- [10] Kevin Cleary, Patrick Cheng, Andinet Enquobahrie, and Ziv Yaniv, editors. *IGSTK: The Book*. Signature Book Printing, 2009.
- [11] INTEL Corporation. Intel. Sitio web: <http://www.intel.com/>.
- [12] R. Szeliski y H. Hirschmüller D. Scharstein. Middlebury stereo datasets. Sitio web: <http://vision.middlebury.edu/stereo>.
- [13] Nothern Digital. The aurora electromagnetic measurement system. Sitio web: <http://www.ndigital.com/medical/aurora.php>.
- [14] Nokia's Qt Development Framework division. Qt. Sitio web: <http://qt.nokia.com/>.

- [15] P. J. Edwards, A. P. King, C. R. Maurer Jr, D. A. De Cunha, D. J. Hawkes, D. L. G. Hill, R. P. Gaston, M. R. Fenlon, A. Juszczak, A. J. Strong, et al. Design and evaluation of a system for microscope-assisted guided interventions (magi). *IEEE Transactions on Medical Imaging*, 19(11):1082–1093, 2000.
- [16] Andinet Enquobahrie, Patrick Cheng, Kevin Gary, Luis Ibáñez, David Gobbi, Frank Lindseth, Ziv Yaniv, Stephen Aylward, Julien Jomier, and Kevin Cleary. The image-guided surgery toolkit IGSTK: An open source C++ software toolkit. *Journal of Digital Imaging*, 20(Suppl. 1):21–33, 2007.
- [17] Andinet Enquobahrie, David Gobbi, Matthew W. Turek, Patrick Cheng, Ziv Yaniv, Frank Lindseth, and Kevin Cleary. Designing tracking software for image guided surgery applications: IGSTK experience. *International Journal of Computer Assisted Radiology and Surgery*, 3(5):395–403, 2008.
- [18] R. Evans. Vislan computer-aided surgery. *IEE Review*, 41(2):51–54, 1995.
- [19] J.M. Fitzpatrick, DL Maurer, and C.R.J. Maci. Measurement of intraoperative brain surface deformation under a craniotomy. *Neurosurgery*, 43(3):514–526, 1998.
- [20] J.M. Fitzpatrick, J.B. West, and CR Maurer. Predicting error in rigid-body point-based registration. *IEEE Transactions on Medical Imaging*, 17(5):694–702, 1998.
- [21] J. García, R. Thoranaghatte, G. Marti, G. Zheng, M. Caversaccio, and M.A. González Ballester. Calibration of a surgical microscope with automated zoom lenses using an active optical tracker. *International Journal of Medical Robotics and Computer Assisted Surgery*, 4(1):87, 2008.
- [22] Kevin Gary, Patrick Cheng, Ziv Yaniv, Andinet Enquobahrie, Luis Ibáñez, and Kevin Cleary. The development of the image-guided surgical toolkit (igstk): An open source package for medical interventions. Workshop on Software and Systems for Medical Devices and Services in conjunction with IEEE Real-Time Systems Symposium (RTSS), 2007.
- [23] Adrian Kaehlet Gary Bradski. *Learning OpenCV - Computer Vision with the OpenCV Library*, page 421. O'REILLY, 2008.
- [24] Adrian Kaehlet Gary Bradski. *Learning OpenCV - Computer Vision with the OpenCV Library*, page 435. O'REILLY, 2008.
- [25] Luigi Giaccari. Rodrigues' rotation formula. Sitio web: <http://www.mathworks.com/matlabcentral/fileexchange/authors/31779>.
- [26] Google. Sketchup. Sitio web: <http://sketchup.google.com/>.
- [27] C. Harris and M. Stephens. A combined edge and corner detector. In *4th Alvey Vision Conference*, 1998.

- [28] T. Hartkens, D.L.G. Hill, A.D. Castellano-Smith, D.J. Hawkes, CR Maurer, A.J. Martin, W.A. Hall, H. Liu, and C.L. Truwit. Measurement and analysis of brain deformation during neurosurgery. *IEEE transactions on medical imaging*, 22(1):82–92, 2003.
- [29] A. Hartley, R. I. y Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [30] J. Heikkilä and O. Silven. A four-step camera calibration procedure with implicit image correction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1106–1112. IEEE, 1997.
- [31] B.K.P. Horn et al. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.
- [32] Luis Ibáñez, Andinet Enquobahrie, Matt Turek, Julien Jomier, Rick Avila, Patrick Cheng, Ziv Yaniv, Frank Lindseth, Kevin Gary, and Kevin Cleary. IGSTK: Building high quality roads with open source software. Systems and Architecture for Computer Assisted Intervention Workshop, (MICCAI), 2008.
- [33] Claron Technology Inc. Micron tracker. Sitio web: <http://www.clarontech.com/measurement.php>.
- [34] Kitware. Fltk. Sitio web: <http://www.ftk.org/>.
- [35] Kitware. Igestk (image guided surgery toolkit). Sitio web: <http://www.igstk.org/>.
- [36] Kitware. Itk. Sitio web: <http://http://www.itk.org/>.
- [37] Kitware. Vtk. Sitio web: <http://http://www.vtk.org/>.
- [38] V. Kolmogorov. *Graph based algorithms for scene reconstruction from two or more views*. PhD thesis, Citeseer, 2004.
- [39] K. Konolige. Small vision systems: Hardware and implementation. In *ROBOTICS RESEARCH-INTERNATIONAL SYMPOSIUM-*, volume 8, pages 203–212. Citeseer, 1998.
- [40] Federico Lecumberry. Cálculo de disparidad y segmentación de objetos en secuencias de video. Master’s thesis, Facultad de Ingeniería - UDELAR, 2005.
- [41] Q. Li, L. Zamorano, Z. Jiang, JX Gong, A. Pandya, R. Perez, and F. Diaz. Effect of optical digitizer selection on the application accuracy of a surgical localization system—a quantitative comparison between the optotrak and flashpoint tracking systems. *Computer aided surgery: official journal of the International Society for Computer Aided Surgery*, 4(6):314, 1999.
- [42] Y. Ma, S. Soatto, J. Kosecka, Y. Ma, S. Soatta, J. Kosecka, and S. Sastry. *An invitation to 3-D vision*. Springer, 2004.

- [43] N. Maitland and C. Harris. A video based tracker for use in computer aided surgery. In *Proc 5th British Machine Vision Conf, York*, pages 609–681, 1994.
- [44] C.R. Maurer Jr, J.M. Fitzpatrick, M.Y. Wang, R.L. Galloway Jr, R.J. Maciunas, and G.S. Allen. Registration of head volume images using implantable fiducial markers. *IEEE Transactions on Medical Imaging*, 16(4):447–462, 1997.
- [45] MIT. 3D Slicer. Open-source software for visualization, registration, segmentation, and quantification of medical data . Sitio web: <http://www.slicer.org/>.
- [46] Visual Computing Lab of the Italian National Research Council. Sitio web oficial. Sitio web: <http://meshlab.sourceforge.net/>.
- [47] OpenCV. Sitio web: <http://opencv.willowgarage.com/wiki/>.
- [48] WonWood's Augmented Reality Web Page. Camera models. Sitio web: <http://old.uvr.gist.ac.kr/wlee/web/techReports/ar/Camera%20Models.html>.
- [49] P. Paul, O. Fleig, and P. Jannin. Augmented virtuality based on stereoscopic reconstruction in multimodal image-guided neurosurgery: Methods and performance evaluation. *IEEE transactions on medical imaging*, 24(11):1500–1511, 2005.
- [50] T.M. Peters. Image-guided surgery: from x-rays to virtual reality. *Computer Methods in Biomechanics and Biomedical Engineering*, 4(1):27–57, 2000.
- [51] T.M. Peters and K. Cleary. *Image-Guided Interventions: Technology and Applications*. Springer Verlag, 2008.
- [52] Pinnacle. Pinnacle, a division for avid. Sitio web: http://www.pinnacleal.com/Site_espanol/inicio_espo.htm.
- [53] J. Preciozzi. Dense urban elevation models from stereo images by an affine region merging approach. *Master's Thesis, Universidad de la República, Montevideo, Uruguay*, 2006.
- [54] A. Raabe, R. Krishnan, R. Wolff, E. Hermann, M. Zimmermann, and V. Seifert. Laser surface scanning for patient registration in intracranial image-guided surgery. *Neurosurgery*, 50(4):797, 2002.
- [55] D.W. Roberts, A. Hartov, F.E. Kennedy, M.I. Miga, and K.D. Paulsen. Intraoperative brain shift and deformation: a quantitative analysis of cortical displacement in 28 cases. *Neurosurgery*, 43(4):749, 1998.
- [56] D.W. Roberts, J.W. Strohbehn, J.F. Hatch, W. Murray, and H. Kettenberger. A frameless stereotaxic integration of computerized tomographic imaging and the operating microscope. *Journal of Neurosurgery*, 65(4):545–549, 1986.

- [57] DF VASCONCELLOS SA. Óptica e mecânica de alta precisão. Sitio web: <http://www.dfv.com.br>.
- [58] Y. Sato, M. Nakamoto, Y. Tamaki, T. Sasama, I. Sakita, Y. Nakajima, M. Monden, and S. Tamura. Image guidance of breast cancer surgery using 3-d ultrasound images and augmented reality visualization. *IEEE Transactions on Medical Imaging*, 17(5):681–693, 1998.
- [59] SourceForge. Open computer vision library. Sitio web: <http://sourceforge.net/projects/opencvlibrary/>.
- [60] H. Sun, H. Farid, A. Hartov, K. Lunn, D. Roberts, and K. Paulsen. Real-time correction scheme for calibration and implementation of microscope-based image-guided neurosurgery. In *Proceedings of SPIE Medical Imaging*, pages 47–54. Citeseer, 2002.
- [61] Hai Sun. *Stereopsis-guided brain shift compensation*. PhD thesis, Thayer School of Engineering, Dartmouth College.
- [62] S.A. Tebo, D.A. Leopold, D.M. Long, S.J. Zinreich, and D.W. Kennedy. An optical 3d digitizer for frameless stereotactic surgery. *IEEE Computer graphics and applications*, pages 55–64, 1996.
- [63] Scot A. Tebo, Donald A. Leopold, Donlin M. Long, S. James Zinreich, and David W. Kennedy. An optical 3d digitizer for frameless stereotactic surgery. *IEEE Computer Graphics and Applications*, 16:55–64, 1996.
- [64] Accelerating the pace of engineering The MathWorks and science. 3-d scatter plot. Sitio web: <http://www.mathworks.com/access/helpdesk/help/techdoc/ref/scatter3.html>.
- [65] Theodore Watson. Videoinput library. Sitio web: <http://muonics.net/school/spring05/videoInput/>.
- [66] J.B. West, J.M. Fitzpatrick, S.A. Toms, C.R. Maurer Jr, and R.J. Maciunas. Fiducial point placement and the accuracy of point-based, rigid body registration. *Neurosurgery*, 48(4):810, 2001.
- [67] the free encyclopedia Wikipedia. Augmented reality. Sitio web: http://es.wikipedia.org/wiki/Realidad_aumentada.
- [68] the free encyclopedia Wikipedia. Brainlab. Sitio web: http://www.brainlab.com/scripts/website_english.asp.
- [69] the free encyclopedia Wikipedia. Diodo emisor de luz. Sitio web: http://es.wikipedia.org/wiki/Diodo_emisor_de_luz.
- [70] the free encyclopedia Wikipedia. Levenberg-marquardt algorithm. Sitio web: http://en.wikipedia.org/wiki/Levenberg-Marquardt_algorithm.

-
- [71] the free encyclopedia Wikipedia. Obj. Sitio web: <http://en.wikipedia.org/wiki/Obj>.
- [72] the free encyclopedia Wikipedia. Opengl. Sitio web: <http://es.wikipedia.org/wiki/OpenGL>.
- [73] the free encyclopedia Wikipedia. Rodrigues' rotation formula. Sitio web: http://en.wikipedia.org/wiki/Rodrigues'_rotation_formula.
- [74] R.G. Willson. *Modeling and calibration of automated zoom lenses*. PhD thesis, The Robotics Institute Carnegie Mellon University Pittsburgh University, 1994.
- [75] Ziv Yaniv and Kevin Cleary. Image-guided procedures: A review. Technical Report CAIMR TR-2006-3, Image Science and Information Systems Center, Georgetown University, April 2006.
- [76] Ziv Yaniv, Emmanuel Wilson, David Lindisch, and Kevin Cleary. Electromagnetic tracking in the clinical environment. *Med. Phys.*, 36(3):876–892, 2009.
- [77] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *International Conference on Computer Vision*, volume 1, pages 666–673, 1999.