

UNIVERSIDAD DE NAVARRA

ESCUELA SUPERIOR DE INGENIEROS

SAN SEBASTIÁN



**Interacción en Sistemas Hápticos
Multisensoriales: Respuesta de Colisión y
Mejoras de Usabilidad**

M E M O R I A

que para optar al Grado de Doctor
presenta

JOSUNE HERNANTES APEZETXEA

bajo la dirección de
Dr. Diego Borro Yagüez

San Sebastián, diciembre 2008

Servicio de Publicaciones de la Universidad de Navarra

ISBN 84-8081-151-X

Nire gurasoei

AGRADECIMIENTOS

Ahora que por fin ha llegado el final de esta tesis me gustaría agradecer a todas las personas que de forma directa o indirecta, han participado en este trabajo haciendo más llevaderos los años transcurridos. Por motivos de espacio o por olvido involuntario no puedo mencionar a toda la gente que me ha ayudado a lo largo de este trabajo, y por ello pido disculpas de antemano a todos ellos.

En primer lugar, me gustaría agradecer a Alejo Avello, Luis Matey y Jordi Viñolas el haberme dado la oportunidad de trabajar en el departamento de Mecánica del CEIT.

Cómo no, quisiera agradecer muy especialmente la labor de mi director de tesis, Diego Borro, y de mi tutor Alex García-Alonso. Gracias de todo corazón por toda la confianza depositada en mí, por sus consejos, ayuda, dedicación y sobre todo paciencia que han llevado a que este trabajo llegue a buen puerto.

Agradecer también a Menchu y a Jose Manuel la oportunidad que me dieron para poder iniciarme en el mundo de la docencia primero como ayudante de Cálculo I y posteriormente con la asignatura de Informática. También al departamento de Organización, muy especialmente a Sarri, Javi y Jose por la paciencia que han tenido durante estos meses de fin de tesis.

A mis compañeros de despacho, que han hecho mucho más llevadero este final de tesis gracias a las charlas sobre bolsa, cine, música...en especial a Aiert y a Iker por ser mis gurús particulares en temas de gráficos por computador.

A Hugo, Carlos, Jairo y Dimas por sus ideas, sugerencias y su inestimable ayuda en la asignatura de Informática y por supuesto por hacer más llevaderos los laboratorios poniendome al día de todos los chismes.

A todos los amigos que he ido haciendo en mis periplos por los diferentes despachos. Primero en el “agujero” y después en robótica. Aitor, Alberto, Ignacio, Erro, Iñigo, Miguel y Mikel gracias por todos los buenos momentos que me habeis hecho pasar con esas tardes de 24, sesiones de

marujeos, cenas y sobre todo por vuestro apoyo y cariño a lo largo de estos años.

A los robóticos, agradecerles que me dejaran redecorar el laboratorio cuando me trasladé y aunque no lo reconozcan todos salimos ganando con el cambio. Ahora en serio, gracias a todos (Eli, Javitxu...) por acoger tan bien a esta intrusa del piso de abajo y por todos los buenos ratos que hemos pasado en robótica con nuestras “discusiones” y dando caña a nuestro vinagres particular.

A Maite y Enrique me gustaría agradecerles de forma muy especial todo su apoyo, comprensión y ánimos que me han dado a lo largo de toda la Tesis pero especialmente en esta última fase. Y al chico del Goeirri, Iñaki, muchísimas gracias por animarme en los momentos de agobio, por hacer más llevadero el trabajo, los experimentos, ficheros .m... y lo que es más importante, gracias por los cafés y la compañía. Ha sido una suerte trabajar contigo.

No me puedo olvidar de todos los componentes del *Haptics Group*, en especial de Emilio, Jorge y Joan por todo su apoyo y por dejarme ser parte de su grupo de trabajo del que he aprendido que las cosas se pueden hacer bien si se ponen ganas. Es una pena pero no pudo ser!!!!

A todos mis compañeros del piso de abajo y arriba (Gaizka, Yaiza, Aitor, Sergio, Virginia, Goretti, Alex, Javi, Alex, Marcos...) que han hecho que estos años sean muy agradables y a las nuevas incorporaciones de robótica en especial a Jorge y a Javi por su simpatía desde el primer día. También a Manolo y a Ana por su ayuda y su predisposición ante cualquier duda que he tenido.

Por último, quiero dedicar estas líneas a mis aitas y mi hermana por su cariño y su apoyo incondicional. A mi cuadrilla (Aritz, Josune, Miren, Ritxi, Raúl, Lore, Lauri, Mikel, Javi, Amaia y David) por sus ánimos y en especial a dos personas que siempre han estado ahí: Marta, gracias por tu confianza y apoyo a lo largo de todos estos años y cómo no por esas aventurillas que hemos vivido en tu coche y que nos animaban la mañana. Y a Iñaki De La Fuente que ha sufrido esta Tesis tanto como yo pero que no ha dejado de mimarme y apoyarme en todo este tiempo. No lo hubiese podido hacer sin vosotros.

ÍNDICE

AGRADECIMIENTOS	I
ÍNDICE	III
LISTA DE FIGURAS	VII
LISTA DE TABLAS	XIII
RESUMEN	XV
ABSTRACT	XVII
NOTACIÓN	XIX
CAPÍTULO 1: INTRODUCCIÓN	1
1.1 Elementos de un sistema háptico	2
1.2 Arquitectura del sistema háptico	3
1.3 Aplicaciones.....	5
1.4 Motivación.....	8
1.5 Objetivos.....	10
1.6 Estructura de la memoria	11
CAPÍTULO 2: ESTADO DEL ARTE	13
2.1 Detección de colisiones	15
2.1.1 Volúmenes Contenedores	16
2.1.2 Métodos de subdivisión espacial	16
2.1.3 Métodos de Jerarquía de Volúmenes.....	19
2.1.4 Métodos de proximidad.....	19
2.1.5 Otros métodos	20
2.2 Respuesta de colisión	20
2.2.1 Representación del objeto móvil.....	22
2.2.2 Estrategias de respuesta	23
2.2.3 Respuesta de 6 gdl	31
2.3 Algoritmo de control.....	42
CAPÍTULO 3: DETECCIÓN DE COLISIONES	43
3.1 Descripción general del problema.....	44
3.2 Cálculo de colisiones	45

3.2.1	Etapa de Preproceso	45
3.2.2	Etapa en tiempo de ejecución.....	47
3.3	Modificaciones realizadas.....	48
3.3.1	Detección de triángulos internos móviles.....	50
3.3.2	Detección de Triángulos Frontera Móviles.....	55
3.3.3	Cálculo de los triángulos frontera estáticos.....	57
3.4	Consideraciones finales.....	62
CAPÍTULO 4: PROBLEMA DE LA RESPUESTA DE COLISIÓN.		65
.....		
4.1	Respuesta de colisión.....	66
4.2	Cálculo del número de contactos implicados en la colisión	67
4.2.1	Cálculo de las áreas de contacto.....	68
4.2.2	Cálculo de las zonas de contacto.....	71
4.3	Cálculo de las normales de contacto para cada área	73
4.3.1	Caso de una zona de contacto.....	75
4.3.2	Caso de varias zonas de contacto.....	76
4.4	Cálculo del punto de contacto	79
4.5	Últimas consideraciones.....	81
CAPÍTULO 5: MÉTODOS PARA EL CÁLCULO DE LA PENETRACIÓN.....		83
5.1	Estado del arte de métodos de penetración	84
5.2	Distancia punto-plano: 1ª aproximación	86
5.3	Método de Hippmann modificado.....	88
5.4	Método Rayo -Triángulo.....	92
5.4.1	Precisión del método.....	95
5.4.2	Paralelización del método.....	97
5.5	Método z-buffer	100
5.5.1	Fases del método	102
5.5.2	Factores que influyen en la eficacia del método	106
5.6	Restitución visual	113
5.7	Consideraciones finales y limitaciones	114
CAPÍTULO 6: RESULTADOS EXPERIMENTALES		117
6.1	Arquitectura del sistema.....	118
6.1.1	Dispositivos hápticos.....	121
6.1.2	Módulo de Visualización	123
6.1.3	Módulo de Detección y Respuesta de Colisiones.....	124
6.1.4	Módulo de Control.....	125
6.1.5	Comunicación entre módulos.....	126

6.2	Objetivos y Descripción de los experimentos.....	127
6.3	Criterios de evaluación y factores que los influyen.....	129
6.4	Experimentos con fuerzas.....	131
6.4.1	Path Grabado	133
6.4.2	Experimentos interactivos.....	134
6.4.3	Consideraciones especiales al método del Z-buffer.....	141
6.5	Experimentos con Pares.....	144
6.5.1	Path grabado.....	145
6.5.2	Experimentos interactivos.....	149
6.6	Experimentos en tareas de desmontaje	155
6.7	Lazo de control.....	161
6.8	Resumen de resultados	163
CAPÍTULO 7: ESTRATEGIAS MULTISENSORIALES		165
7.1	Ampliación sensorial.....	167
7.1.1	Influencia del sonido en tareas de accesibilidad.....	168
7.1.2	Descripción del Sistema Multisensorial.....	170
7.1.3	Descripción del Escenario.....	171
7.1.4	Usuarios y Procedimiento	173
7.1.5	Resultados	174
7.1.6	Conclusiones.....	176
7.2	Sustitución sensorial.....	177
7.2.1	Simplificación del diseño de una muñeca háptica	178
7.2.2	Descripción de la estrategia de sustitución sensorial	179
7.2.3	Validación de la estrategia	185
CAPÍTULO 8: CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN.....		191
8.1	Conclusiones	191
8.2	Futuras líneas de investigación	194
ANEXO A: DISPOSITIVOS HÁPTICOS.....		197
A.1	Phantom premium 1.0	197
A.2	Palanca de 3 grados de libertad.....	198
A.3	Guía lineal.....	200
A.4	Lhifam	201
ANEXO B: PUBLICACIONES.....		205
ANEXO C: REFERENCIAS		217

LISTA DE FIGURAS

Figura 1.1: Elementos de un sistema háptico.....	2
Figura 1.2: Sistema de Realidad Virtual con respuesta visual, auditiva y táctil.....	3
Figura 1.3: Arquitectura básica de un sistema háptico multisensorial.....	4
Figura 1.4: Aplicaciones hápticas en medicina. Sistema de entrenamiento de VirtaMed para operaciones de cirugía mínimamente invasiva (a), y sistema de entrenamiento de Virtualis para la palpación de vientres de vacas (b).....	5
Figura 1.5: Sistema de entrenamiento de GMV para operaciones de artroscopia.	6
Figura 1.6: Simulador de Boeing (a) e ITP (b) para tareas de accesibilidad y mantenimiento en el sector aeronáutico.....	6
Figura 1.7: Sistema háptico de teleoperación daVinci.....	7
Figura 1.8: Simulador virtual de una bolera: dispositivo háptico de 5 gdl (a) y entorno virtual (b).....	8
Figura 1.9: Sistema REVIMA.....	10
Figura 2.1: Principales módulos de un algoritmo de renderizado háptico.	14
Figura 2.2: Volumen encerrado por un AABB (a), un OBB (b) y una esfera (c) junto con los datos necesarios para su definición.	16
Figura 2.3: Subdivisión en voxels (a) y subdivisión utilizando una jerarquía de octrees (b). 18	
Figura 2.4: En los métodos de renderizado directo, la posición de la herramienta virtual y la posición real del háptico coinciden siempre (a). En los métodos de acoplamiento virtual, la posición de la herramienta virtual se desacopla de la posición real del háptico de tal manera que permanezca en la superficie del objeto con el que ha colisionado (b).....	22
Figura 2.5: Diferentes representaciones para la herramienta virtual: basada en un punto (a), basado en un rayo (b) y objetos 3D (c) (Basdogan and Srinivasan 2002).	22
Figura 2.6: Ejemplo de colisión entre un punto y una pared virtual.....	25
Figura 2.7: Limitaciones de los métodos basados en un punto. Discontinuidades en la fuerza (a). Pop-Thru. (b) y Falta de localización (c). La esfera azul representa la posición real de la herramienta virtual, la esfera roja la posición del objeto en la superficie y la esfera morada la posición incorrecta resultante. Las líneas discontinuas representan los sub-volumenes internos de los objetos.....	26
Figura 2.8: Cálculo de la posición del <i>god-object</i>	29
Figura 2.9: Modelo de una tetera representado mediante una malla de triángulos (a), superficies (b) y voxels (c).....	32
Figura 2.10: Ejemplo de discontinuidad en los pares.....	34
Figura 2.11: El área de contacto se define mediante varios puntos de contacto.....	34

Figura 2.12: CLODs de una dentadura. En la parte superior izquierda se muestra el modelo original. La parte inferior derecha muestra el modelo simplificado, mientras que el resto representan diferentes resoluciones del mismo modelo.	35
Figura 2.13: Cálculo de la posición G del <i>god-object</i> (a) y fuerzas y pares \mathbf{F} restituidos al usuario (b).....	37
Figura 2.14: Voxmap de un objeto estático colisionando con el Point Shell de un objeto móvil.....	38
Figura 2.15: Cálculo de la penetración mediante los Voxmap-PointShell.....	38
Figura 2.16: Error en el cálculo de la penetración mediante los Voxmap-PointShell.....	39
Figura 3.1: Esquema general del cálculo de colisiones.	45
Figura 3.2: Representación 2D del proceso de voxelización. Reserva de memoria (a), selección de los voxels que intersectan con la caja contenedora de cada primitiva (b), cálculo de la intersección entre el triángulo y cada voxel seleccionado (c), los voxels marcados contendrán por lo menos una primitiva (d).	47
Figura 3.3: Esquema del cálculo de colisiones en tiempo de ejecución.	48
Figura 3.4: Herramienta virtual que colisiona con un objeto. En color rojo se representan los triángulos en colisión asociados al objeto estático y en negro los triángulos en colisión del objeto móvil (a). Se añaden los triángulos del objeto móvil interiores al objeto estático (b).....	50
Figura 3.5: Clasificación de voxels: en azul los voxel internos, en amarillo los externos y finalmente, en rosa los voxels frontera. Inicialmente, todos los voxels se clasifican como internos (a), aquellos voxels que contengan triángulos se clasifican como frontera (b). Se inicia el proceso para encontrar los voxels exteriores (c) y finalmente, la clasificación obtenida (d).	52
Figura 3.6: Ejemplo 3D de la clasificación del voxelizado: modelo (a) y voxelizado generado (b). En rojo se representan los voxels clasificados como externos y en verde los clasificados como frontera. Las celdas internas no se ven por encontrarse en el interior del modelo.	53
Figura 3.7: Ejemplo 3D de la penetración entre dos objetos. En color azul se han representado los triángulos en colisión del objeto móvil, en verde los voxels frontera y en azul los voxels internos.....	54
Figura 3.8: Comparación de valores de penetración obtenidos sin y con la detección de los triángulos internos móviles.	54
Figura 3.9: Ejemplo en el que se muestran los triángulos denominados frontera.....	55
Figura 3.10: Colisión en el instante t (a) y en el instante $t+\Delta t$ (b). En el eje Y se representa el nivel de penetración en mm.....	56
Figura 3.11: Influencia del tamaño del voxelizado en el cálculo de la penetración.	57
Figura 3.12: Ejemplo de dos objetos en colisión. Los triángulos del objeto estático de la zona central del área de intersección no son detectados.....	58
Figura 3.13: Ejemplo de la influencia de los triángulos frontera estáticos en el cálculo de la penetración.....	59

Figura 3.14: Comparación de valores obtenidos en el cálculo de la penetración con y sin la detección de los triángulos frontera estáticos.....	59
Figura 3.15: Comparación del tiempo de detección de colisión para cada una de las opciones.	61
Figura 3.16: Número de triángulos estáticos en contacto para cada una de las opciones....	61
Figura 3.17: Triángulos estáticos detectados con la opción 1 (a) y con la opción 2 (b).	62
Figura 4.1: Flujo de información entre los módulos de detección, respuesta de colisiones y control.	66
Figura 4.2: Comparativa del coste computacional del cálculo de triángulos vecinos	69
Figura 4.3 Comparativa de la memoria consumida en el cálculo de triángulos vecinos.....	70
Figura 4.4: Separación por áreas de los triángulos en contacto del objeto estático y móvil. Representación 2D (a) y 3D (b).	71
Figura 4.5: Ejemplo 2D de la división en varias zonas de contacto.....	72
Figura 4.6: Ejemplo 2D (a) y 3D (b) de la subdivisión en áreas y zonas de contacto.....	73
Figura 4.7: Colisión de un objeto móvil con un objeto estático en un área de contacto: el área de contacto se divide en una zona de contacto (a), el área se divide en dos zonas de contacto representadas en color rojo y verde (b).....	75
Figura 4.8: Colisión de la herramienta atravesando un tubo.	76
Figura 4.9: Primera aproximación (a) y segunda aproximación (b) para el cálculo de la dirección de la fuerza resultante en caso de existir una discontinuidad en la superficie.....	78
Figura 4.10: La variación en el punto de contacto del objeto móvil provoca que el objeto oscile.	79
Figura 4.11: Respuesta háptica distinta en función del punto de contacto seleccionado: momento y fuerza (a), sólo una fuerza (b).....	80
Figura 4.12: Punto de contacto calculado a partir de los voxels que contienen a los triángulos en contacto del objeto estático.	80
Figura 5.1: Error de cálculo del método de Redon. Se ha representado la penetración calculada (<i>dIncorrecta</i>) y la penetración real (<i>dReal</i>).	85
Figura 5.2: Plano de contacto Π situado en el vértice más alejado en la dirección de la normal $\mathbf{n}_{\text{área}}$	86
Figura 5.3: Error en el cálculo de la penetración utilizando únicamente las facetas en colisión del objeto móvil. Ejemplo 2D (a) y ejemplo 3D (b).....	87
Figura 5.4: Posibles discontinuidades debidas al plano de contacto.	88
Figura 5.5: Elemento de contacto. Transformación del triángulo móvil al sistema de referencia definido por el triángulo estático.	89
Figura 5.6: Coste computacional del cálculo de la penetración para el método de Hippmann y el método de Hippmann modificado.	91
Figura 5.7: Número de triángulos en contacto.....	91
Figura 5.8: Discretización del volumen de intersección: utilizando rayos definidos con la normal asociada a cada triángulo estático representados en color verde (a) y utilizando	

rayos definidos mediante la normal resultante asociada a esa área de contacto (b). Las líneas denominadas b_i representan las alturas con las que se muestrea el volumen.....	93
Figura 5.9: Ejemplo 3D del cálculo de la penetración con la técnica de los rayos.	94
Figura 5.10: Esquema del método Rayo Triángulo.....	95
Figura 5.11: Error en el cálculo de la penetración ya que ningún rayo intersecta los triángulos del objeto móvil.....	96
Figura 5.12: Tiempos de cálculo de la penetración con dos áreas de contacto.....	99
Figura 5.13: Número de triángulos en colisión asociados a cada área de contacto.	99
Figura 5.14: Paralelización del método basado en rayos.	100
Figura 5.15: Transferencia de la información entre la GPU y la memoria principal.	102
Figura 5.16: Funcionamiento del z-buffer.....	103
Figura 5.17: Esquema del método sin utilizar shaders (a) y utilizando shaders (b).	105
Figura 5.18: Diagrama de flujo del método que utiliza el z-buffer.	106
Figura 5.19: Posicionamiento incorrecto de la cámara que impide dibujar los triángulos del objeto estático (a) y posicionamiento correcto de la cámara (b).....	107
Figura 5.20: Volumen de visión que no engloba el total del área de intersección (a) y volumen de visión acorde al tamaño del área de intersección (b).....	108
Figura 5.21: Penetración obtenida con tamaños de textura de 32x32, 64x64 y 128x128. En color negro se representa la penetración calculada como referencia.....	109
Figura 5.22: Penetración obtenida con tamaños de textura de 256x256, 512x512, 1024x1024 y 2048x2048. En color negro se representa la penetración calculada como referencia.	110
Figura 5.23: Tiempos de cálculo de la penetración con diferentes tamaños de textura.....	111
Figura 5.24: Tiempos de cálculo de la penetración con tamaños de texturas más pequeños.	111
Figura 5.25: El cálculo de la respuesta háptica se realiza en base a la posición real del objeto manipulado por el usuario (representado mediante sus triángulos), mientras que visualmente se representa el objeto móvil desacoplado cuya posición es calculada mediante el método del <i>god-object</i>	114
Figura 6.1: Modelo CAD del sistema (a) y el sistema real en funcionamiento (b).....	119
Figura 6.2: Esquema de la arquitectura del sistema REVIMA.	122
Figura 6.3: Modelo de 1.615.172 polígonos original (a) y simplificado un 40% (b).	123
Figura 6.4: Definición de un volumen de trabajo alrededor de un elemento.....	124
Figura 6.5: Mensajes intercambiados entre los módulos de REVIMA.	127
Figura 6.6: Entorno real en el que mediante un zoom se muestran zonas con discontinuidades geométricas y múltiples contactos.....	132
Figura 6.7: Doble contacto tipo 2: Penetraciones obtenidas en la simulación con los diferentes métodos y las variaciones que se producen en sus magnitudes.	134
Figura 6.8: Variación de la normal a medida que la esfera se acerca a la esquina.	135
Figura 6.9: Esquina convexa: Penetraciones y variaciones de la penetración a lo largo de la simulación para cada método.	136

Figura 6.10: Esquina cóncava: Variación de la normal cuando se recorre la esquina cóncava.	137
Figura 6.11: Esquina cóncava. Penetraciones obtenidas y variaciones entre <i>frames</i> consecutivos.....	137
Figura 6.12: Doble contacto tipo 1: Normales obtenidas a lo largo de la simulación en superficies redondeadas.....	138
Figura 6.13: Doble contacto tipo 1: Penetraciones y variaciones del módulo de penetración obtenidas a lo largo de la simulación para cada método.....	139
Figura 6.14: Doble contacto tipo 2: Variación de las normales en un hueco con discontinuidades geométricas.....	140
Figura 6.15: Doble contacto tipo 2: Penetraciones y variaciones del módulo de penetración obtenidas a lo largo de la simulación para cada método.....	140
Figura 6.16: Penetración obtenida con el z-buffer (a) y con el método de rayos (b).....	142
Figura 6.17: Zoom de las penetraciones obtenidas con el z-buffer (a) y con el método de rayos (b).....	143
Figura 6.18: Escenario teórico para testear el cálculo de los momentos: objetos laterales cilíndricos (a) y objetos laterales paralelepípedos (b).	145
Figura 6.19: Par resultante en cada uno de los ejes (T_x , T_y y T_z).....	146
Figura 6.20: Zoom de la zona con discontinuidad.	147
Figura 6.21: Variaciones en T_x a lo largo de la simulación	147
Figura 6.22: Tiempo de respuesta de colisión y coste computacional final (tiempo de detección de colisiones + tiempo de respuesta)	148
Figura 6.23: Número de triángulos en colisión común para todos los métodos.....	148
Figura 6.24: Simulación de tres giros de la abrazadera sobre el tubo central de la maqueta.	149
Figura 6.25: Simulación de los diferentes tipos de colisión con un entorno de objetos cilíndricos.	151
Figura 6.26: Zoom de las discontinuidades en el cálculo del momento en el eje x (T_x) para el experimento de objetos cilíndricos.....	152
Figura 6.27: Simulación de los diferentes tipos de colisión con un entorno de objetos paralelepípedos.....	154
Figura 6.28: Comparación del método basado en rayos con el método del <i>god-object</i>	155
Figura 6.29: Entorno de simulación de la extracción de una abrazadera a través de una tubería de una turbina de avión.....	156
Figura 6.30: Extracción de la abrazadera de una tubería.	157
Figura 6.31: Sistema utilizado para hacer el experimento.....	157
Figura 6.32: Escenario real de desmontaje para testear el cálculo de los momentos.....	159
Figura 6.33: Caso detallado de una colisión compleja en un entorno real	160
Figura 6.34: Actuación del algoritmo de control.	162
Figura 6.35: Actuación del algoritmo de control.	163
Figura 7.1: Secuencia de desensamblado de una pieza en un motor de avión.....	169

Figura 7.2: Descripción de los distintos módulos que componen el interfaz háptico multisensorial	171
Figura 7.3: Laberinto virtual empleado en los experimentos.....	172
Figura 7.4: Penetración media de colisión (mm) por cada usuario en cada modalidad, y diagrama de cajas con los datos (la línea dentro del rectángulo representa el valor de la mediana, y los asteriscos los valores atípicos).....	174
Figura 7.5: Tiempo (s) empleado por los participantes en completar la tarea en cada modalidad, y diagrama de cajas con los datos (la línea dentro del rectángulo representa el valor de la mediana, y los círculos los valores medios)	176
Figura 7.6: Rotación hand-roll.	180
Figura 7.7: Háptico e ilusión háptica.....	181
Figura 7.8: Movimiento libre en el giro hand-roll en el mundo real que se traduce al mundo virtual mediante una rotación de la herramienta virtual en este eje.	182
Figura 7.9: Estímulo táctil y visual proporcionado en el caso de la colisión con objetos del entorno virtual.	183
Figura 7.10: Ejemplo de interacción en espacio libre y colisión debido a rotaciones en hand-roll. De arriba abajo: Simulación del par ejercido por el usuario, rotación virtual resultante en hand-roll y estímulo táctil restituido al usuario.	184
Figura 7.11: Tiempo medio de reacción (s) para cada prueba, y valor medio y desviación estándar en cada modalidad y diagrama de cajas con los datos (las líneas dentro de las cajas representan las medianas y los círculos las medias).	188
Figura A. 1: PHANToM Premium 1.0.....	198
Figura A. 2: Palanca del CEIT de 3 gdl.....	199
Figura A. 3: Guía lineal.....	201
Figura A. 4: Modelo CAD del LHIFAM (a) y usuario manejando el LHIFAM con una maqueta virtual (b).....	202
Figura A. 5: Dimensiones del espacio de trabajo del LHIFAM.....	202
Figura A. 6: Diferentes reubicaciones del espacio de trabajo.....	203

LISTA DE TABLAS

Tabla 3.1: Resumen de los términos utilizados en el capítulo.....	49
Tabla 6.1: Modelos estáticos utilizados en las pruebas de fuerzas.....	132
Tabla 7.1: Valores de los parámetros utilizados en la modalidad de 2,5 gdl.	182
Tabla A. 1: Principales características técnicas del PHANTOM <i>Premium 1.0</i>	198
Tabla A. 2: Especificaciones de la palanca del CEIT de 3 gdl.	200
Tabla A. 3: Especificaciones de la guía lineal	201
Tabla A. 4: Especificaciones del LHIFAM.....	203

RESUMEN

Las aplicaciones de Realidad Virtual son cada vez más frecuentes en áreas como la industria o la medicina ya que pueden utilizarse como herramientas de trabajo o de aprendizaje. La necesidad actual de que estos sistemas sean cada vez más realistas obliga a introducir el sentido del tacto aumentando de forma considerable la precisión y eficiencia con la que pueden desarrollarse las tareas además de la inmersión de los usuarios en este tipo de entornos.

Los dispositivos hápticos permiten al usuario interactuar con objetos del entorno virtual a través del sentido del tacto percibiendo una respuesta de fuerza a cualquier colisión con el entorno. Sin embargo, esta tecnología háptica es todavía reciente lo que no permite obtener una respuesta estable y realista en casos complejos.

La presente Tesis investiga y ofrece soluciones en dos áreas fundamentales de la háptica: el cálculo de las fuerzas y pares de contacto como resultado de las acciones de los usuarios en entornos virtuales y la percepción multisensorial.

La detección y respuesta de colisión en tiempo real es una tarea complicada y esencial para la efectividad de estos sistemas hápticos. Este trabajo desarrolla un método de renderizado háptico capaz de calcular una respuesta háptica estable y agradable incluso en situaciones complejas tales como las tareas de desensamblado en entornos aeronáuticos.

Además, se analiza la eficacia de combinar los distintos canales sensoriales disponibles para mejorar la percepción global del sistema así como la inmersión final del usuario.

ABSTRACT

Virtual Reality applications are becoming very frequent in areas such as industry or medicine since they can be used as advanced working or learning tools. There is a need to enhance the realism of these systems which motivates to add the sense of touch increasing significantly the accuracy and efficiency to perform tasks in addition to the immersion of users in this type of environments.

Haptic devices allow users to interact with virtual objects through the sense of touch receiving force feedback as a response to collisions with the environment. However, haptic technology is still recent and as a result it is difficult to get a stable and realistic response in complex environments.

This thesis investigates and provides solutions in two haptic key areas: the calculation of contact forces and torques as a result of user actions in virtual environments and multisensory perception.

Real-time collision detection and response is a complicated and essential task for the effectiveness of these systems. This work develops a haptic rendering method able to compute an improved and stable haptic response even in complex situations such as disassembly operations in aeronautic environments.

Furthermore, it analyses the effectiveness of combining different sensory modalities available on the system to improve the overall perception of the system as well as user immersion.

NOTACIÓN

Siglas

AABB	Axis Aligned Bounding Box
ANOVA	ANalysis Of VAriance
CEIT	Centro de Estudios e Investigaciones Técnicas de Gipuzkoa
gdl	Grado de libertad
LHI _f AM	Large Haptic Interface for Aeronautics Maintainability
PHANToM	Personal HAptic iNTERface Mechanism
GPU	Graphical Processor Unit. Tarjeta gráfica

Renderizado háptico

b	Amortiguamiento virtual del modelo de fuerza de contacto con un objeto virtual
\mathbf{c}_g	Centro de inercia del objeto virtual controlado por el usuario
\mathbf{p}_c	Punto de contacto en una colisión entre objetos virtuales
\mathbf{F}	Fuerza aplicada al usuario
k	Rigidez virtual del modelo de fuerza de contacto con un objeto virtual
M	Lista de triángulos que definen al objeto móvil
M_c	Lista de triángulos en contacto del objeto móvil
\mathbf{n}	Vector normal de contacto
\mathbf{n}_{area}	Vector normal de un área de contacto
\mathbf{n}_{zona}	Vector normal de una zona de contacto
numTri	Número de triángulos contenidos en una lista de triángulos
numZonas	Número de zonas de contacto dentro de un área de contacto
d	Penetración de la herramienta virtual en el escenario
\mathbf{p}	Vértice cualquiera de un triángulo
\mathbf{q}	Punto de intersección entre un rayo y un triángulo
S	Lista de triángulos que definen al objeto estático
S_c	Lista de triángulos en contacto del objeto estático
\mathbf{T}	Par aplicado al usuario
$\{X Y Z\}$	Sistema de referencia fijo

CAPÍTULO 1

INTRODUCCIÓN

Desde hace unas décadas, los sistemas de Realidad Virtual se están implantando en áreas muy diversas como la medicina o la industria. Por ejemplo, pueden utilizarse como simuladores de entrenamiento para realizar operaciones quirúrgicas o como herramientas para validar prototipos sin la necesidad de construir maquetas reales. Sin embargo, muchos sistemas tienen la limitación de sólo transmitir información a través de dos canales: el visual y el auditivo. Este hecho hace que los sistemas de realidad virtual no sean adecuados para simular tareas en las que es necesario que el usuario interactúe físicamente con el entorno. Para solventar estas limitaciones, y en concreto añadir el sentido del tacto a los entornos virtuales, a comienzos de la década de los 90 se empezaron a desarrollar los dispositivos hápticos integrados en sistemas de Realidad Virtual.

El estudio y desarrollo de los dispositivos hápticos se conoce como *Haptics* (Háptica). El término *haptic* proviene del griego “*haptesthai*”, y hace referencia al tacto. Se refiere a la interacción manual con entornos, ya sea en forma de exploración (para obtener información) o manipulación (Srinivasan et al. 1997). La Háptica engloba campos muy distintos de la ingeniería, fundamentalmente el diseño mecánico, el control, la visualización 3D y desarrollo de algoritmos de cálculo geométrico.

Los dispositivos hápticos permiten al usuario tocar, sentir y manipular los objetos simulados en entornos virtuales a través de un dispositivo físico sensorizado. De hecho, Burdea (1999) establece que la respuesta háptica es crucial en aplicaciones de Realidad Virtual para poder manipular objetos de la escena ya que permite sentir propiedades físicas tales como forma, peso,

inercia o rugosidad de la superficie, incrementando de esta forma la inmersión del usuario.

Este capítulo explica algunos conceptos relacionados con los hápticos, así como los principales campos de aplicación de estos dispositivos.

1.1 ELEMENTOS DE UN SISTEMA HÁPTICO

La Figura 1.1 muestra los componentes de los que está compuesto un sistema háptico. El *usuario* utiliza el *dispositivo háptico* para manipular objetos o *herramientas virtuales* que le permiten llevar a cabo una tarea determinada en un *escenario*. El dispositivo háptico permite al usuario controlar el movimiento de la herramienta virtual y como respuesta a la interacción con el escenario, éste recibe un estímulo de fuerza y/o par, lo que permite simular el comportamiento de objetos reales en un mundo virtual.

En el caso concreto de esta Tesis, el escenario permanece estático de ahí que se le denomine *objeto estático* y a la herramienta virtual se le denominará *objeto móvil*. Se asumirá que ambos objetos son rígidos y que por lo tanto no sufren ningún tipo de deformación. Otra característica del entorno en el que se va a trabajar en esta Tesis consiste en que la herramienta virtual está definida por un número significativamente menor de triángulos en comparación con el objeto estático.

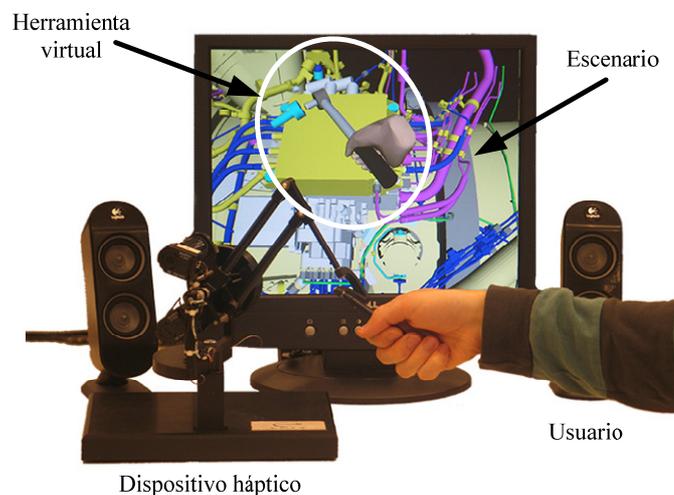


Figura 1.1: Elementos de un sistema háptico.

A los sistemas hápticos que además incluyen otras modalidades sensoriales como el canal auditivo se les denominan “*sistemas hápticos multisensoriales*”.

Una de las principales razones por las que la inclusión del tacto mejora la inmersión en el sistema se debe a que el flujo de información en este canal sensorial es bidireccional mientras que la información de las respuestas visuales y auditivas es unidireccional, es decir, del entorno virtual al usuario. Los dispositivos hápticos intercambian información en dos direcciones ya que por un lado, el usuario proporciona una entrada al entorno virtual que provoca cambios visuales y de desplazamiento, y como consecuencia se proporciona una salida al usuario mediante la actuación de los motores del dispositivo háptico. La Figura 1.2 muestra el flujo de la información por cada canal sensorial.

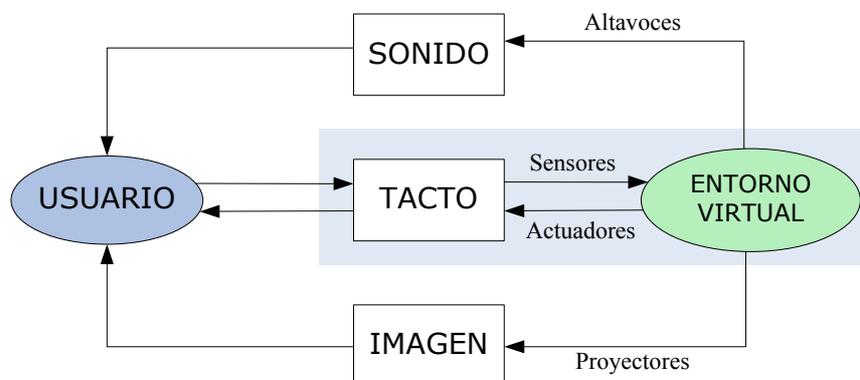


Figura 1.2: Sistema de Realidad Virtual con respuesta visual, auditiva y táctil.

El dispositivo háptico está dotado de sensores que permiten al sistema conocer la posición del usuario y de actuadores que restituyen la fuerza de contacto con el escenario virtual.

1.2 ARQUITECTURA DEL SISTEMA HÁPTICO

Al proceso por el que un entorno virtual es representado a un usuario mediante los estímulos sensoriales apropiados se le conoce como renderización, traducción de la palabra inglesa “*rendering*”. En su nivel más básico puede tratarse de una simple representación visual de un objeto, como por ejemplo un cubo. Si además es posible dotar al cubo de propiedades tales como texturas y/o sombras, el resultado se asemeja más a la percepción que obtendría un usuario en el mundo real. De la misma

manera, el renderizado háptico tiene como objetivo estimular el sentido del tacto de los usuarios con el objetivo de conseguir que las simulaciones virtuales sean más inmersivas y permitan un mayor grado de interactividad.

En las últimas décadas, los gráficos por computador han sufrido una auténtica revolución. De simples representaciones de líneas en 2D, se ha conseguido que muchas aplicaciones virtuales se asemejen a la realidad de manera sorprendente. De forma análoga, en el renderizado háptico, no es lo mismo sentir una fuerza que de alguna manera pueda informar acerca de un contacto entre objetos, a poder reproducir efectos reales como la fricción o la sensación de rugosidad. Si bien hoy por hoy, comparativamente, el renderizado háptico está en sus comienzos con respecto al renderizado gráfico, el desarrollo de este campo pretende revolucionar el concepto de interacción virtual existente hasta el momento. En el siguiente esquema (Figura 1.3), se puede observar una arquitectura básica de una aplicación virtual en la que interactúan estímulos hápticos, visuales y auditivos.

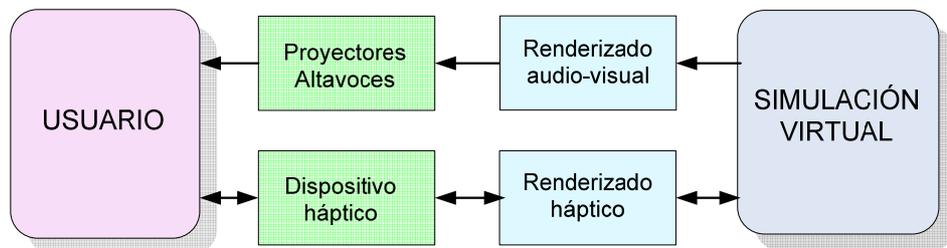


Figura 1.3: Arquitectura básica de un sistema háptico multisensorial.

En este tipo de sistemas, la colisión entre el objeto virtual controlado por el usuario y el escenario generará tres tipos de estímulos: una fuerza, un sonido, y la propia visualización de la escena. Cada uno de estos estímulos debe generarse a una frecuencia mínima a partir de la cual el ser humano es capaz de percibirla adecuadamente. Así, por ejemplo, se considera que se obtiene una visualización interactiva a partir de una frecuencia de 25 Hz percibiendo múltiples imágenes como una secuencia continua. En cambio, para el tacto, una frecuencia mínima para una buena sensación háptica está en torno a 1 kHz. El sistema auditivo es el más restrictivo al necesitar una frecuencia en torno a los 20 kHz para la generación de sonidos en tiempo real.

1.3 APLICACIONES

Actualmente los sistemas hápticos están siendo implantados en múltiples áreas como la medicina o la industria, y se espera que en los próximos años el uso de estos sistemas crezca de manera importante. Generalmente, estos sistemas hápticos se emplean principalmente en tareas de simulación y entrenamiento en las que el usuario interactúa con el escenario virtual a través de una herramienta como un bisturí o un destornillador.

En el campo de la medicina, los dispositivos hápticos se están extendiendo como herramientas muy útiles para poder realizar simulaciones de operaciones quirúrgicas (Figura 1.4a), permitiendo entrenar al personal médico. Incluso en el área veterinaria existe actualmente un dispositivo para entrenar a los estudiantes en la palpación del vientre de las vacas (Figura 1.1b).



(a)

(b)

Figura 1.4: Aplicaciones hápticas en medicina. Sistema de entrenamiento de VirtaMed para operaciones de cirugía mínimamente invasiva (a), y sistema de entrenamiento de Virtalis para la palpación de vientres de vacas (b).

La Figura 1.5 muestra el simulador virtual para operaciones de artroscopia de hombro desarrollado por la compañía GMV. Mediante dos dispositivos hápticos Omni, el usuario puede controlar el instrumental quirúrgico empleado en estos procesos. El sistema es un complemento ideal a técnicas tradicionales de formación, contribuyendo a facilitar la adquisición de destrezas básicas, reduciendo los tiempos de entrenamiento y mejorando considerablemente la curva de aprendizaje en estas complejas técnicas quirúrgicas.

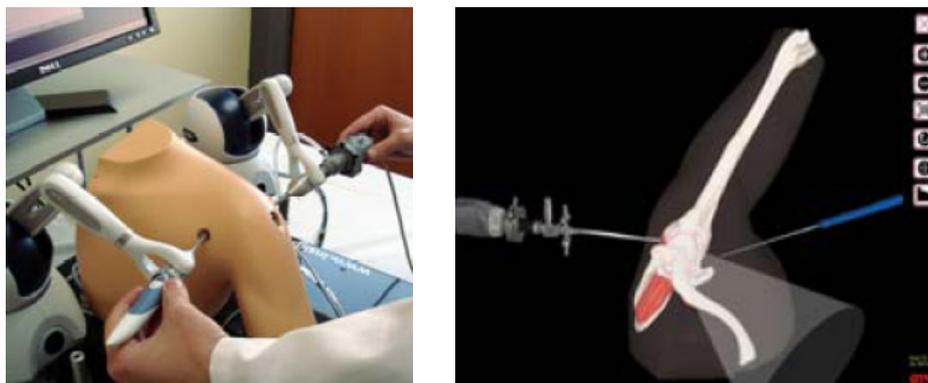


Figura 1.5: Sistema de entrenamiento de GMV para operaciones de artroscopia.

En el campo de la industria existen también diversas aplicaciones en las que los dispositivos hápticos son de gran utilidad como es el caso del prototipado virtual de motores aeronáuticos. Con maquetas tan complejas, es necesario garantizar en la etapa de diseño la accesibilidad a todos los componentes que lo forman. La validación de cada modelo requiere la construcción de maquetas físicas de elevado coste. Sin embargo, mediante el uso de dispositivos hápticos, esta validación puede realizarse con maquetas virtuales, lo que permite reducir el número de maquetas previas disminuyendo al mismo tiempo el coste, tiempo de diseño y permitiendo a su vez una detección precoz de problemas de diseño. Ejemplos de ello son los simuladores de las compañías Boeing o ITP (Figura 1.6). Este último es un proyecto desarrollado por el CEIT para la empresa ITP.

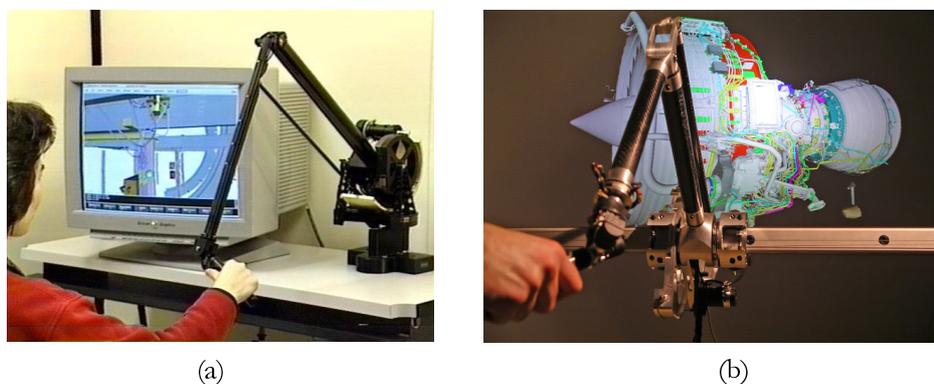


Figura 1.6: Simulador de Boeing (a) e ITP (b) para tareas de accesibilidad y mantenimiento en el sector aeronáutico.

Otro campo donde se está investigando el uso de los sistemas hápticos es en la teleoperación. El hecho de tener información táctil de lo que sucede en el extremo del robot esclavo es de gran ayuda en sistemas robóticos quirúrgicos o industriales. La Figura 1.7 muestra el sistema robótico *daVinci* de *Intuitive Surgical Inc.* para operaciones quirúrgicas mínimamente invasivas. En este caso el robot es teleoperado mediante un dispositivo háptico.



Figura 1.7: Sistema háptico de teleoperación daVinci.

En los últimos años también se están invirtiendo muchos recursos para el desarrollo de sistemas hápticos en el área de ocio y entretenimiento ya que la restitución de fuerzas aumenta considerablemente la sensación de inmersión del usuario en los juegos. En la Figura 1.8 se puede ver un simulador virtual de una bolera desarrollado por el Instituto de Ciencias Aplicadas de Nagasaki, en el que para simular el lanzamiento se utiliza un dispositivo de 5 grados de libertad (en adelante gdl) desarrollado por el propio instituto (Lawn et al. 1997).

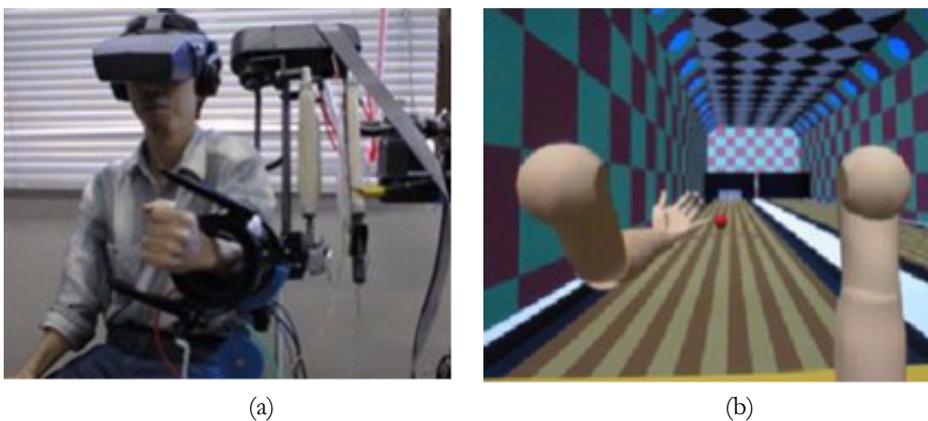


Figura 1.8: Simulador virtual de una bolera: dispositivo háptico de 5 gdl (a) y entorno virtual (b).

1.4 MOTIVACIÓN

El uso de sistemas hápticos es hoy por hoy una realidad en muchas áreas tal y como se han enumerado en el apartado anterior, y un campo con un futuro de gran potencial. Al tratarse todavía de una tecnología reciente, existen carencias que impiden que los estímulos hápticos generados sean tan adecuados y estables como fuese deseable, por lo que conseguirlo es uno de los grandes retos en este campo.

Durante los últimos años, se han desarrollado diversos métodos de renderizado háptico capaces de calcular las fuerzas debidas a colisiones entre objetos virtuales 3D, pero son menos los métodos que además son capaces de calcular los pares asociados a esas colisiones. Por otra parte, estos métodos son capaces de responder adecuadamente a contactos entre objetos virtuales sencillos como esferas o planos, pero cuando se trata de interacciones en entornos virtuales de geometrías complejas (o que existan discontinuidades geométricas), los métodos actuales todavía no son capaces de dar una respuesta adecuada.

Numerosos trabajos han demostrado que la inclusión de la restitución de fuerzas en un entorno virtual aumenta el realismo y la inmersión del usuario dentro del entorno (Burdea 1999). De hecho, este tipo de interacción se considera indispensable para poder llevar a cabo tareas de mantenimiento y accesibilidad.

Las tareas de accesibilidad consisten en buscar el camino más adecuado para poder acceder a un elemento de difícil acceso dentro de un sistema. En el caso de la industria y concretamente en tareas de mantenimiento en las que se usan modelos complejos, es necesario determinar en fase de diseño si los diferentes componentes que forman el modelo van a ser accesibles para ser desensamblados. Para detectar los posibles errores de diseño se construyen maquetas de madera de tamaño real, y con frecuencia se requiere la fabricación de nuevas maquetas con el coste que eso supone. Utilizar modelos virtuales y un sistema háptico para detectar estos fallos de diseño disminuye en gran medida estos costes. Este fue el principal objetivo del proyecto REVIMA (Desarrollo de una Herramienta de REalidad Virtual para la Simulación de procesos de montaje y MAntenimiento), sistema háptico desarrollado por el CEIT (Figura 1.9) en cuya implementación se vieron implicadas diferentes áreas de investigación como el diseño mecánico, teoría de control, gráficos por computador, geometría computacional e interacción persona-computador (Savall et al. 2002).

Una de las principales características de este sistema es su dispositivo háptico de gran tamaño concebido para la simulación de tareas de accesibilidad y mantenimiento en entornos aeronáuticos. Este dispositivo denominado LHIFAM (Large Haptic Interface for Aeronautic Maintainability) (Borro et al. 2004b) tiene 6 gdl aunque sólo 3 de ellos están actuados, por lo que la restitución háptica se limita a las fuerzas. Aunque este tipo de respuesta es adecuada para realizar ciertas tareas de accesibilidad, resulta insuficiente en la mayoría de los casos en los que además de las fuerzas es necesario obtener una restitución de pares. De hecho, diversos autores respaldan la idea de que la inclusión de los pares en la respuesta háptica incrementa el realismo en las aplicaciones mencionadas anteriormente (Cohen et al. 1999, Chen 1999).

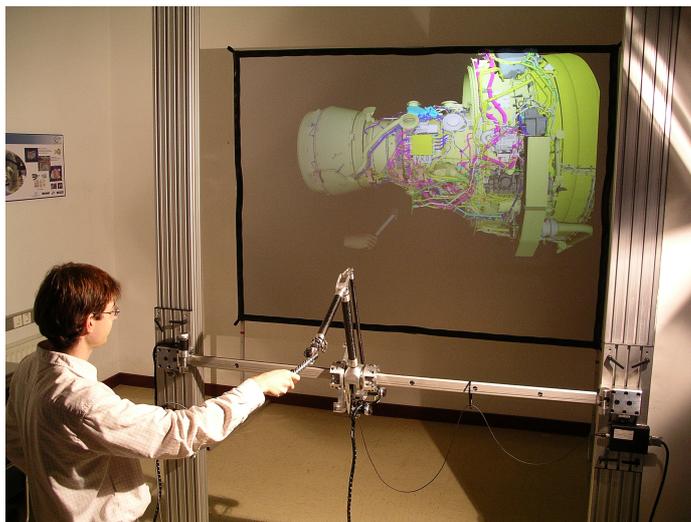


Figura 1.9: Sistema REVIMA.

El objetivo final sería conseguir ampliar este sistema a 6 gdl, lo que implica realizar cambios tanto a nivel hardware como software. Para solucionar las restricciones debidas al hardware actualmente se está desarrollando una muñeca háptica específica para este dispositivo que permitirá ampliar los gdl del sistema.

Pero además de los cambios necesarios en el diseño mecánico, hay que realizar cambios a nivel de software desarrollando un método de renderizado capaz de calcular una respuesta háptica estable y continua tanto en fuerzas como en pares.

1.5 OBJETIVOS

El objetivo principal de este trabajo es generar una respuesta háptica tanto en fuerzas como en pares estable y agradable al usuario de forma que se mejore la interacción con estos sistemas y la inmersión del usuario. De esta forma, se mejorará también la integración de estos sistemas en entornos de prototipado virtual y en otras aplicaciones. Hay que destacar que dentro del renderizado háptico, este trabajo se centra en la detección y respuesta de colisiones. Es decir, calcular la información necesaria de fuerzas y pares para que el control las aplique al dispositivo háptico. Todo lo relacionado con el lazo de control pertenece a otro trabajo que se realiza en paralelo a esta Tesis.

Los casos de especial interés en este trabajo serán aquellos que impliquen colisiones complejas con el entorno como discontinuidades geométricas o contactos múltiples. En estas situaciones se dará prioridad a la obtención de una respuesta continua, suave y por lo tanto agradable para el usuario que a una respuesta realista en términos físicos que implicaría seguramente inestabilidades hápticas debido a esas discontinuidades. Así, además de proporcionar una respuesta háptica adecuada, se garantizará la seguridad del usuario ante posibles inestabilidades del sistema, evitando que éste reciba niveles de fuerza elevados de forma inesperada.

Como actualmente no se dispone de un dispositivo háptico de 6 gdl, se estudiará el problema de la restitución de fuerzas y momentos de forma desacoplada utilizando dos dispositivos hápticos diferentes: uno con 3 gdl traslacionales actuados y otro con 3 gdl actuados en rotación. Como el objetivo perseguido es conseguir un sistema completo de 6 gdl, también se realizarán pruebas con un sistema de 4 gdl (3 de rotación y 1 de traslación), que aunque no reproduzca un sistema completo puede ayudar a validar la solución general propuesta.

Asimismo, se analizarán dos estrategias multisensoriales que combinan de forma inteligente el uso de diferentes canales sensoriales para mejorar la inmersión del usuario en los entornos virtuales y suplir las carencias tanto mecánicas como de renderizado de los sistemas hápticos actuales. Diversos autores señalan que la percepción háptica del usuario puede verse afectada por información proveniente de estos medios, con lo que el estudio de la manera en que pueden combinarse los distintos canales sensoriales disponibles puede abrir otra línea de investigación muy interesante.

1.6 ESTRUCTURA DE LA MEMORIA

Esta memoria está organizada en 8 capítulos, que van cubriendo los objetivos planteados en el apartado anterior.

En el capítulo 2 se realiza una revisión bibliográfica de los trabajos que abordan el problema del cálculo de una respuesta de colisión en los sistemas hápticos.

El capítulo 3 analiza el problema de la detección de colisiones entre dos objetos. Es necesario calcular la información geométrica adecuada para poder utilizarla en el cálculo de las fuerzas y momentos a restituir tras producirse la colisión con el entorno.

En el capítulo 4 se describen los algoritmos para calcular las fuerzas y los momentos a partir de la información geométrica calculada en el capítulo anterior. Esta fase también es referida en la bibliografía como el cálculo de la respuesta de colisión. Se distinguen tres factores que van a determinar que la respuesta háptica calculada sea estable. Los dos primeros, la dirección y el punto de contacto, se analizan en este capítulo. El problema de la penetración se analiza en el capítulo 5, donde se describen una serie de algoritmos propuestos.

Una vez calculados todos los factores necesarios, en el capítulo 6 se realiza una comparativa con los métodos propuestos y se muestran los resultados obtenidos en un ejemplo real de desensamblado.

El capítulo 7 analiza dos alternativas multisensoriales para incrementar el realismo y la inmersión en los sistemas hápticos.

Por último, en el capítulo 8 se presentan las conclusiones derivadas de este trabajo y las futuras líneas de investigación propuestas para la continuación de esta investigación.

CAPÍTULO 2

ESTADO DEL ARTE

Una síntesis de este capítulo se ha presentado en:

Borro, D., Hernantes, J., García-Alonso, A., and Matey, L., "Collision problem: Characteristics for a Taxonomy", Proceedings of the 9th International Conference Information Visualisation (IV'05), pp. 410-415. London, England. July 6-8, 2005.

Al proceso por el que un entorno virtual simulado es representado al usuario mediante los estímulos sensoriales apropiados se le conoce como renderización, que es una adaptación al castellano de la palabra inglesa "rendering". De la misma forma, el renderizado háptico (*haptic rendering*) tiene como objetivo estimular el sentido del tacto de los usuarios con el propósito de conseguir que las simulaciones virtuales sean más realistas, además de permitir un mayor grado de interactividad.

Los algoritmos de renderizado háptico son los encargados de calcular las fuerzas necesarias para percibir una interacción realista entre la representación virtual del dispositivo háptico y los objetos virtuales que constituyen la escena, y de asegurarse de que esta fuerza es transmitida correctamente al usuario mediante el dispositivo háptico (Salisbury et al. 1995b).

Tres son los componentes principales de los algoritmos de renderizado háptico: detección de colisiones, cálculo de la respuesta a dicha colisión y el módulo de control (Figura 2.1).

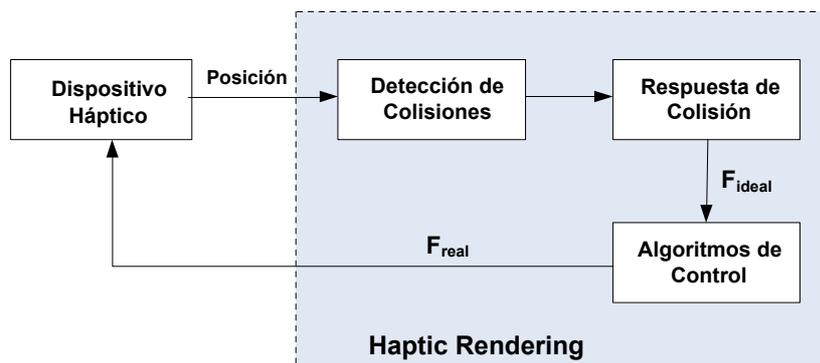


Figura 2.1: Principales módulos de un algoritmo de renderizado háptico.

El módulo de colisiones toma la posición del dispositivo háptico y la información del entorno virtual para detectar si hay colisiones. La respuesta de colisión toma la información de colisión y calcula la fuerza ideal de interacción F_{ideal} entre la herramienta virtual y el entorno. Por último, los algoritmos de control devuelven al usuario una fuerza F_{real} que aproxima la fuerza ideal a las capacidades del dispositivo háptico.

En los siguientes apartados se describen los diferentes módulos y las distintas técnicas existentes en la bibliografía utilizadas para resolver los problemas que se plantean. También se realiza una descripción del tipo de entornos virtuales con los que se va a trabajar.

El tipo de modelos que se utilizan en esta Tesis representan objetos rígidos, por lo que la revisión bibliográfica que se realiza está enfocada a resolver el problema de la detección y el cálculo de una respuesta adecuada para este tipo de objetos. El análisis por ejemplo de técnicas orientadas a objetos deformables queda fuera del alcance de este trabajo.

Los objetos que componen una escena pueden estar definidos mediante diferentes tipos de representaciones que a su vez condicionan las técnicas a utilizar para calcular una respuesta háptica tras una colisión. Otro factor que condiciona la complejidad y la precisión de la interacción con el entorno así como los grados de libertad de la respuesta háptica apropiada, es la representación del objeto móvil utilizada para el cálculo de dicha respuesta. En los siguientes apartados se analizan estos dos factores.

2.1 DETECCIÓN DE COLISIONES

La fidelidad de las simulaciones depende entre otros factores de la semejanza entre el comportamiento de los objetos reales y los virtuales. En el mundo real, los objetos sólidos no pueden penetrar entre sí, ya que existen fuerzas físicas que lo impiden. Sin embargo, a menos que estas fuerzas sean impuestas explícitamente, en el mundo virtual los objetos son libres de penetrar unos con otros. El objetivo de la detección de colisiones es informar cuándo se detecta algún contacto geométrico entre dos sólidos y determinar cómo es ese contacto.

Existen numerosos trabajos sobre el problema de la detección de colisiones, ya que aparece en una amplia gama de situaciones y aplicaciones: robótica, gráficos por computador, modelado, videojuegos o entornos de simulación. El problema de colisiones es diferente en cada una de estas aplicaciones por lo que se utilizan diferentes técnicas en cada caso. Hay que destacar las taxonomías aportadas en (Hubbard 1996, Jiménez et al. 2001, Lin et al. 1998, Lin et al. 1996, O'Sullivan et al. 2001) sobre los distintos campos y problemas de detección de colisiones.

Sin embargo, determinar únicamente si dos objetos virtuales colisionan, no es suficiente para calcular posteriormente una respuesta física a esta colisión. Para ello, es necesario calcular información adicional como la distancia de penetración, los puntos de contacto o las normales, que conlleva gran coste computacional. En (Lin et al. 2004) se realiza un útil estado del arte sobre los algoritmos de detección de colisiones utilizados dentro del área del renderizado háptico.

Partiendo del hecho de que en el cálculo de colisiones, el mayor porcentaje del tiempo se lo lleva la comparación de polígonos y objetos que realmente no se encuentran en colisión, las diferentes técnicas existentes en la bibliografía envuelven los objetos en estructuras que permiten rechazar rápidamente pares de objetos a testear.

En un trabajo ya publicado (Borro et al. 2005), realizamos un amplio estudio sobre los métodos existentes, clasificándolos en cuatro grandes grupos: métodos basados en volúmenes contenedores, métodos basados en subdivisión espacial, métodos basados en la jerarquía de volúmenes y por último, los métodos de proximidad. La presente Tesis trabaja con modelos poligonales y rígidos por lo que la clasificación se centrará en métodos que aborden este tipo de problemas.

2.1.1 VOLÚMENES CONTENEDORES

En este tipo de métodos los objetos de la escena se encierran en volúmenes y gracias a su geometría más simple, la detección de colisiones entre esos volúmenes es más fácil y eficiente. Entre los volúmenes más utilizados se encuentran los AABBs, OBBs y las esferas. La Figura 2.2 muestra un ejemplo de cada uno de estos volúmenes contenedores.

Los AABBs (*Axis-Alignment Bounding Box*) se construyen considerando las mínimas y máximas coordenadas del objeto que encierran (\mathbf{p}_{\min} y \mathbf{p}_{\max}) y son utilizados en las librerías de detección de colisiones I-COLLIDE (Cohen et al. 1995) y V-COLLIDE (Hudson et al. 1997).

Por su parte, Gottschalk et al. (1996) utilizan las cajas orientadas al objeto u OBBs (*Oriented Bounding Box*) que son cajas rectangulares con una orientación arbitraria en el espacio 3D que permiten ajustar mejor el tamaño del volumen contenedor, mediante los ocho puntos que lo definen (\mathbf{p}_i).

A diferencia de los dos volúmenes anteriores, las esferas son independientes de la orientación del objeto y son fáciles de calcular aunque no son válidas si lo que se requiere es precisión en los cálculos (Ritter 1990). Se definen en función de un punto (\mathbf{p}_c) y de un radio (r).

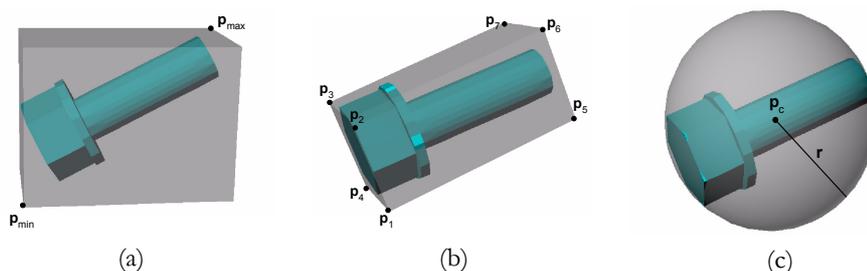


Figura 2.2: Volumen encerrado por un AABB (a), un OBB (b) y una esfera (c) junto con los datos necesarios para su definición.

2.1.2 MÉTODOS DE SUBDIVISIÓN ESPACIAL

La subdivisión espacial proporciona una valiosa información sobre la localización geométrica de un objeto, acelerando la detección de posibles intersecciones entre los diferentes objetos de la escena. Es también muy útil para algoritmos que no necesitan utilizar la geometría exacta para el cálculo de colisiones o puede utilizarse como una primera aproximación.

Los métodos de partición espacial generan una estructura de datos auxiliar a la geometría que define los objetos, de forma que permiten centrar la atención en la zona o zonas donde realmente los objetos se encuentran muy cercanos entre sí para posteriormente calcular de forma exacta la intersección entre estas zonas.

Las técnicas de subdivisión espacial dividen el espacio en volúmenes más pequeños y mantienen una lista de los objetos contenidos en cada volumen. Existen varios tipos de subdivisión espacial, pero dos de los más utilizados son:

Estructura de voxels: En este tipo de partición el espacio es dividido en un conjunto de celdas idénticas unidas entre sí, disjuntas y ordenadas en una malla tridimensional fija y regular (Foley et al. 1990). Estas celdas llamadas voxels, son iguales entre sí y pueden ser cubos (Held et al. 1995, McNeely et al. 1999) o bien paralelepípedos (García-Alonso et al. 1994).

La principal ventaja de la estructura de voxels (Figura 2.3a) es su acceso directo a un voxel aleatorio ya que se realiza en un tiempo constante ($O(1)$). Sin embargo, las principales desventajas de este método de subdivisión espacial son su alto consumo de memoria cuando el nivel de voxelización empieza a ser elevado y la dificultad de determinar el tamaño de la celda.

Estructura de octrees: El volumen de trabajo es subdividido recursivamente en ocho cubos utilizando los planos alineados a los ejes globales (Ayala et al. 1985, Hayward 1986). El proceso continúa de forma recursiva hasta alcanzar la resolución requerida, creando una estructura en forma de árbol (Figura 2.3b).

Comparando con la subdivisión en voxels, el uso de octrees reduce considerablemente el consumo de memoria ya que la partición de cada nodo únicamente se llevará a cabo en caso de ser necesaria (es decir, cuando haya geometría dentro del nodo) con el correspondiente ahorro de almacenamiento que supone el no seguir particionando. Sin embargo, presenta una clara desventaja en el acceso a sus nodos con respecto a los métodos de subdivisión uniforme debido a su estructura de tipo árbol. Los algoritmos de búsqueda utilizados tienen complejidad logarítmica $O(\log n)$ en el peor de los casos, siendo n el número de nodos hoja.

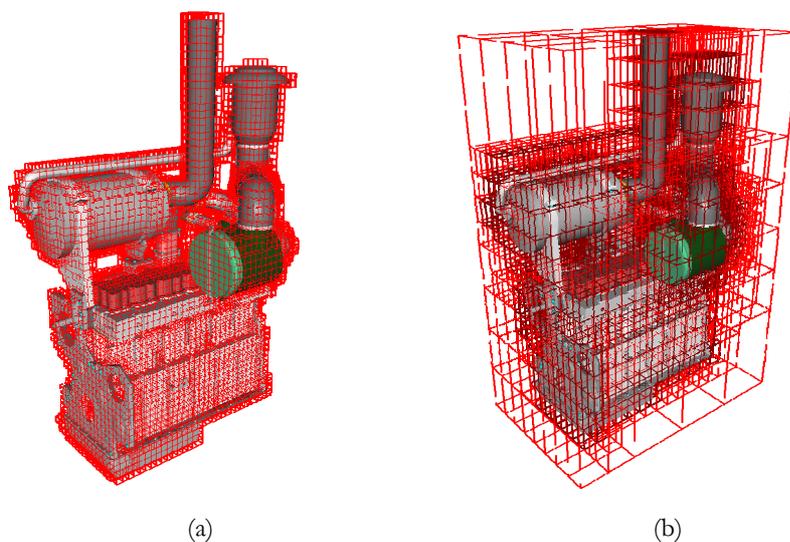


Figura 2.3: Subdivisión en voxels (a) y subdivisión utilizando una jerarquía de octrees (b).

Borro (2003) realiza un completo análisis de las dos técnicas de subdivisión espacial planteadas comprobando que la partición espacial basada en voxels, además de su acceso directo y constante, favorece tareas como la inserción o borrado de objetos del volumen de trabajo. La implementación de este tipo de operaciones utilizando los octrees requeriría volver a calcular la estructura en forma de árbol de los nodos, ya que existirán nodos que será necesario volver a particionar o que por el contrario se han quedado vacíos por lo que no es necesario llevar a cabo una partición.

Los voxels y octrees son los métodos más comunes dentro de la subdivisión espacial, sin embargo existen otras soluciones que utilizan descomposiciones espaciales recursivas diferentes a los octrees. Algunos ejemplos son los BSP (*Binary Space Partitioning*) que separa el espacio mediante planos para formar celdas convexas (Naylor et al. 1990), los *k-d Trees* que actúan como los octrees pero subdividiendo en dos la región a lo largo de uno de los ejes de coordenadas (Van Den Bergen 1999a, Zachmann 1997) o el trabajo de Jiménez (2006) basado en tetra-conos denominada *Tetra-Tree*. El cálculo de esta estructura es más rápido que por ejemplo la estructura de octrees, además de no necesitar excesivo espacio de almacenamiento para representarlo.

2.1.3 MÉTODOS DE JERARQUÍA DE VOLÚMENES

En este tipo de métodos se particionan los mismos objetos (o polígonos) agrupándolos y usando sus volúmenes contenedores hasta formar un árbol. Si el volumen contenedor de un nodo no intersecta con un objeto, tampoco lo harán sus nodos hijos por lo que se puede saltar en la jerarquía. Mientras dos nodos se intersecten habrá que ir descendiendo por el árbol hasta llegar a los nodos hoja donde se resuelve el test entre polígonos.

En base a los volúmenes descritos en el apartado 2.1.1, tres de las jerarquías más utilizadas son: los árboles de cajas alineadas o AABBTrees (Van Den Bergen 1997), árboles de cajas orientadas o OBBTrees (Gottschalk et al. 1996, Hudson et al. 1997) y el árbol de esferas (O'Sullivan et al. 2001).

H-COLLIDE desarrollado por Gregory et al. (1999) es una librería de colisiones optimizada para su uso en aplicaciones de renderizado háptico. Para ello utiliza una combinación de diferentes técnicas como la descomposición espacial de la escena mediante un mallado uniforme de celdas dentro de las cuales utiliza OBBTrees, además de sacar provecho a la coherencia temporal.

2.1.4 MÉTODOS DE PROXIMIDAD

Siguiendo con la idea de la subdivisión espacial en la que se detectan las colisiones entre los objetos cercanos entre sí, los métodos de proximidad se basan también en la coherencia espacial pero utilizan métodos completamente distintos.

El método más utilizado se basa en las Regiones Voronoi que contienen mapas de vecindad con los puntos del espacio que están más cerca de un objeto dado que del resto (Lin 1993, Ponamgi et al. 1997). Estas técnicas son utilizadas para desarrollar las librerías I-COLLIDE (Cohen et al. 1995) y V-CLIP (Mirtich 1998).

Otros trabajos calculan la distancia entre poliedros para saber si dos de ellos intersectan o no (si la distancia es mayor que cero no están colisionando). Para calcular la distancia eficientemente, estos algoritmos se basan en la diferencia de Minkowski entre dos poliedros convexos (Cameron 1997, Cameron et al. 1986, Van Den Bergen 1999b).

2.1.5 OTROS MÉTODOS

Los métodos referenciados en los apartados anteriores utilizan la CPU para realizar todos sus cálculos. Sin embargo, en los últimos años se ha dado un paso importante a la hora de aliviar la carga computacional de la CPU trasladando tareas de cálculo a la tarjeta gráfica. Concretamente, se han desarrollado diversos métodos de detección de colisiones que aprovechan las nuevas capacidades de las tarjetas gráficas para optimizar los cálculos. Para conseguirlo, sacan partido a la alta capacidad de paralelización de la GPU aumentando enormemente su capacidad de cálculo permitiendo procesar varios elementos al mismo tiempo (Grand Le 2007, Harada 2007, Kipfer 2007).

Este tipo de métodos son eficientes para aplicaciones que simulan el comportamiento de los sólidos rígidos, en las cuales se ven implicados numerosos objetos.

2.2 RESPUESTA DE COLISIÓN

La simulación de fenómenos que ocurren en el mundo real mediante la Realidad Virtual plantea numerosos problemas. Uno de los más notorios es que los objetos virtuales carecen de propiedades inherentes a los objetos reales como la imposibilidad de que dos sólidos ocupen un mismo espacio en el mismo instante de tiempo por existir fuerzas físicas que lo impiden. Puesto que la fidelidad de las simulaciones depende entre otros factores de la semejanza entre el comportamiento de los objetos reales y los virtuales, es necesario añadir reglas físicas de forma explícita en el mundo virtual para que los objetos no sean libres de penetrar unos con otros.

Los algoritmos de respuesta de colisión son los encargados de calcular una respuesta física tras detectar la colisión entre dos o más objetos contenidos en el entorno virtual. La respuesta a devolver tras una colisión es específica para el dominio de aplicación (O'Sullivan et al. 2001) y esta respuesta podría ir desde la destrucción de los objetos en un juego hasta el cálculo de las posteriores trayectorias una vez ocurrida la colisión como ocurre en aplicaciones de simulaciones físicas. En el caso del uso de dispositivos de restitución de esfuerzos, la respuesta vendrá dada por el cálculo de una fuerza a restituir realista y estable. Para ello, los algoritmos de respuesta utilizan las posiciones de los objetos de la escena, la posición de la herramienta virtual y la información calculada por el módulo de colisiones para calcular un buen feedback, ya sea de fuerzas y/o pares, y una vez

calculados se envían al módulo de control para que puedan ser aplicados al dispositivo háptico. Trabajos recientes han realizado un estudio bibliográfico sobre las técnicas existentes en esta área de aplicación (Laycock et al. 2007, Salisbury et al. 2004).

Entre los distintos métodos de renderizado háptico existentes en la bibliografía existen dos filosofías muy distintas de concebir la respuesta de colisión entre objetos virtuales: el *renderizado directo* (*direct rendering*) y el *acoplamiento virtual* (*virtual coupling*).

Los métodos de renderizado directo son aquellos en los que la posición y/o orientación del háptico y el objeto virtual controlado por éste coinciden en todo momento, tanto en el movimiento libre como cuando se produce una colisión. En este caso, la respuesta de colisiones es proporcional a la interpenetración de la herramienta virtual en los objetos del escenario (Figura 2.4a). El inconveniente de este tipo de filosofía es que si la rigidez del entorno no es lo suficientemente alta, se puede observar que el objeto virtual penetra cierta distancia en el escenario, lo que puede restar realismo a la interacción.

En los métodos de acoplamiento virtual (Colgate et al. 1995), la posición de la herramienta virtual coincide también con la posición real del háptico en los movimientos libres. Sin embargo, cuando se produce una colisión, la posición de la herramienta virtual se desacopla de la posición real del háptico, de tal manera que ésta quede en todo momento en la superficie del escenario con el que ha colisionado. En este caso, la información de interpenetración entre los objetos se utiliza para gobernar la dinámica de la herramienta y la fuerza devuelta al usuario es proporcional a la distancia entre la posición ideal de la herramienta y la posición real del háptico (Figura 2.4b). A esta unión virtual entre ambas posiciones se le denomina acoplamiento virtual, y suele ser en general un acoplamiento elástico o viscoelástico.

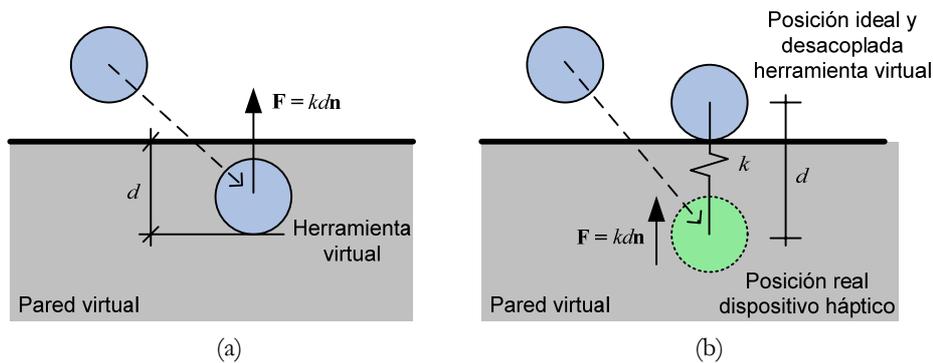


Figura 2.4: En los métodos de renderizado directo, la posición de la herramienta virtual y la posición real del háptico coinciden siempre (a). En los métodos de acoplamiento virtual, la posición de la herramienta virtual se desacopla de la posición real del háptico de tal manera que permanezca en la superficie del objeto con el que ha colisionado (b).

Por lo tanto, a partir del uso de una u otra técnica, la información de colisión se utiliza de manera distinta tal y como se expondrá en los siguientes subapartados.

2.2.1 REPRESENTACIÓN DEL OBJETO MÓVIL

Los métodos utilizados para el cálculo de las fuerzas de contacto dependen de la forma y el diseño de la herramienta con la que se interactúa en el entorno virtual (Basdogan et al. 2002). Siguiendo este criterio, las técnicas pueden clasificarse en métodos basados en un punto, en rayo o en objetos 3D (Figura 2.5).

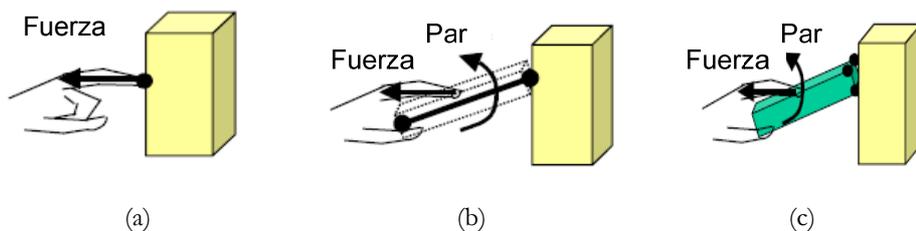


Figura 2.5: Diferentes representaciones para la herramienta virtual: basada en un punto (a), basado en un rayo (b) y objetos 3D (c) (Basdogan and Srinivasan 2002).

La interacción basada en un punto (Figura 2.5a) consigue simplificar considerablemente el problema de la representación del objeto con el que se interactúa en el entorno y la complejidad del algoritmo. De esta forma se

consiguen altas frecuencias permitiendo al usuario sentir un amplio abanico de interacciones.

Sin embargo, este tipo de interacción presenta una limitación importante al no ser capaz de devolver pares. Esto se debe a que al representar la herramienta virtual mediante un punto no se tiene en cuenta su orientación. A pesar de las limitaciones que plantean este tipo de métodos, se consideran válidos para aplicaciones sencillas tales como la pintura y escultura donde el usuario puede percibir una fuerza similar a la que percibiría al explorar los objetos con el extremo o punta de un pincel.

Con el objetivo de superar las limitaciones de los métodos basados en un punto, se desarrollan los métodos basados en un rayo en los que la herramienta virtual se representa mediante una línea finita, cuya orientación se tiene en cuenta a la hora de calcular las fuerzas y los pares.

Las técnicas basadas en un rayo se pueden considerar como una aproximación a una herramienta alargada (Figura 2.5b). Sin embargo puede que no sea suficiente para muchas aplicaciones cuya simulación requiere de mayor precisión en el cálculo de las fuerzas y pares, de modo que es imprescindible utilizar toda la información geométrica del objeto móvil sin recurrir a ningún tipo de simplificación (Figura 2.5c). De esta forma se consigue proporcionar al usuario la habilidad suficiente para sentir, explorar y manipular los objetos dentro del entorno virtual.

Éste es el caso de simulaciones médicas, la teleoperación o el prototipado virtual que simulan tareas de mantenibilidad en las que se necesita conocer con precisión el acceso de la herramienta a ciertos emplazamientos, necesitando para ello una interacción 3D objeto-objeto.

2.2.2 ESTRATEGIAS DE RESPUESTA

A la hora de simular el comportamiento de dos cuerpos al chocarse, la dificultad reside en el cálculo de las fuerzas y pares que deben aplicarse para prevenir que los dos objetos penetren. Además, debe garantizarse que dicha respuesta sea estable para que la simulación sea realista y agradable para el usuario.

En la bibliografía actual existen tres variantes diferentes para resolver el problema del renderizado háptico: métodos basados en penalizaciones, en restricciones y en impulsos.

Los trabajos que se referencian en cada uno de estos grupos aportan soluciones al problema en cuestión pero mayoritariamente utilizan representaciones del objeto móvil basadas en un punto o en un rayo. Aunque estos métodos presentan limitaciones en la respuesta y no son válidos para resolver el problema de la presente Tesis, se ha considerado interesante su estudio para analizar la evolución que se ha producido desde los métodos pioneros que ofrecían una respuesta de 3 gdl a los métodos actuales de 6 gdl, analizados en el apartado 2.2.3.

2.2.2.1 MÉTODOS DE PENALIZACIÓN

Cuando dos objetos colisionan, aunque sean rígidos, se producen pequeñas deformaciones en el área en la que se produce dicha colisión. La filosofía que siguen estos métodos está basada en modelar estas deformaciones mediante muelles en los diferentes puntos de contacto (Moore et al. 1988). La magnitud de la fuerza de repulsión resultante dependerá de la interpenetración producida entre ambos objetos.

De este modo, para poder calcular las fuerzas y los pares (Ecuaciones (2.1) y (2.2)), los métodos basados en penalización utilizan un punto de contacto \mathbf{p}_C , una normal de contacto o dirección de la fuerza \mathbf{n} y la distancia de penetración d , que mide la intersección entre los objetos.

$$\mathbf{F} = f(d)\mathbf{n} \quad (2.1)$$

$$\mathbf{T} = (\mathbf{p}_C - \mathbf{c}_g) \times \mathbf{F} \quad (2.2)$$

Donde \mathbf{c}_g es el centro de masas del objeto y la función $f(d)$ representa el modelo de contacto que puede ser representado mediante un muelle simple, añadirle una componente viscosa o incluso utilizar una función más compleja. Incluso para reducir el coste de la detección de colisiones y disminuir la posible penetración entre objetos, puede añadirse una fuerza repulsiva que empiece a interactuar cuando el objeto se encuentre a una distancia menor a la de un umbral (Salcudean et al. 1994).

Una de las ventajas que presentan los métodos de penalización es que el modelo de fuerzas es local a cada contacto de modo que las fuerzas se calculan en base a posiciones sin tener en cuenta el movimiento del objeto. Además, estos métodos requieren que el control realice un número menor de cálculos permitiendo que el lazo rápido pueda actualizar datos a una frecuencia de 1kHz.

Sin embargo, estos métodos también presentan algunos inconvenientes. La rigidez necesaria del sistema debe ser alta para evitar la penetración entre los objetos lo que puede provocar problemas de estabilidad. Además, la respuesta de estos métodos está claramente condicionada por las discontinuidades inherentes a la geometría que definen los modelos lo que puede provocar discontinuidades en la respuesta háptica.

Numerosos trabajos de la bibliografía han utilizado esta técnica basada en la penalización para el cálculo de una respuesta háptica estable. Los primeros algoritmos desarrollados en el área del renderizado háptico utilizan la representación de la herramienta virtual basada en un punto para simular el contacto con paredes virtuales. A este punto se le denomina HIP (*Haptic Interface Point*) y es utilizado por los algoritmos de detección de colisiones para determinar si existe o no contacto. Para ello se prueba si el HIP se encuentra dentro o fuera del objeto virtual. El módulo de la fuerza final a restituir se calcula como la distancia mínima necesaria para que el objeto deje de estar en colisión.

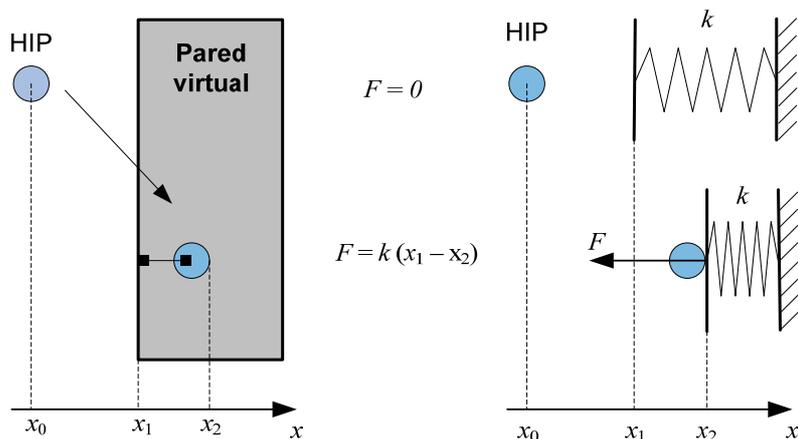


Figura 2.6: Ejemplo de colisión entre un punto y una pared virtual

Massie et al.(1994) desarrollaron una técnica que subdividía el volumen del objeto en subvolúmenes o regiones asociando cada uno de ellos a una superficie. De esta forma, en caso de existir colisión, la dirección de la fuerza resultante es normal a la superficie en colisión.

Sin embargo, tal y como se describen en (Zilles et al. 1995), esta técnica presenta algunas limitaciones a la hora de calcular la dirección de la fuerza resultante. Los principales inconvenientes son los siguientes. En

primer lugar, pueden producirse discontinuidades en la fuerza de respuesta cuando se atraviesan volúmenes internos, devolviendo la fuerza perpendicular a dicho volumen. En este caso, la fuerza resultante provocará que el objeto sea expulsado hacia la segunda superficie (Figura 2.7a).

Los objetos pequeños o finos pueden tener un volumen interno insuficiente para generar las fuerzas de restitución necesarias para prevenir que la herramienta virtual atravesase el objeto, lo que se conoce como efecto *pop-thru* (Figura 2.7b). Por último, para objetos más complejos formados por objetos simples, es difícil determinar a qué superficie se asocian los subvolúmenes del objeto (Figura 2.7c).

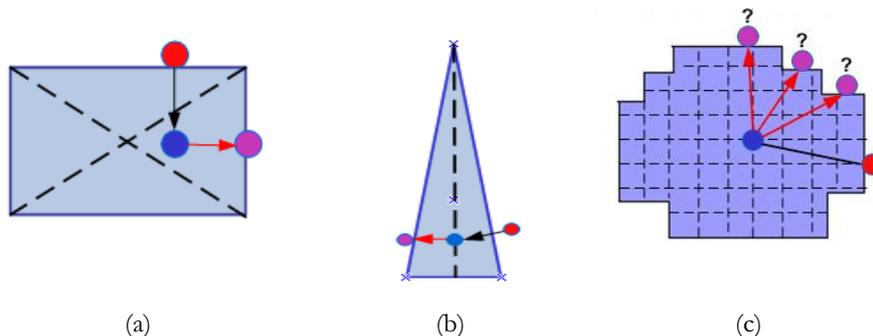


Figura 2.7: Limitaciones de los métodos basados en un punto. Discontinuidades en la fuerza (a). Pop-Thru. (b) y Falta de localización (c). La esfera azul representa la posición real de la herramienta virtual, la esfera roja la posición del objeto en la superficie y la esfera morada la posición incorrecta resultante. Las líneas discontinuas representan los subvolúmenes internos de los objetos.

Para simplificar la interacción entre dos objetos virtuales, trabajos posteriores (Adachi et al. 1995, Mark et al. 1996) reemplazan la superficie que define los objetos por planos intermedios. En cada *frame*, se calcula una aproximación de la superficie mediante un plano tangente en el punto de colisión, de modo que la fuerza de respuesta se calcula en base a este plano tangente a la superficie en el punto de contacto. La dirección de la fuerza de respuesta coincide con la normal del plano tangente generado y el módulo se calcula en base a la distancia de penetración.

El problema de este método es la frecuencia a la cual se calcula el plano. Aunque las fuerzas de respuesta se actualicen a una frecuencia de 1 kHz, el plano lo hace a una frecuencia menor. Esta diferencia entre las frecuencias puede provocar discontinuidades en la fuerza a restituir en el caso en el que la representación virtual del háptico realice un

desplazamiento considerable antes de que el nuevo plano tangente sea actualizado.

Algunas veces la representación mediante mallas de polígonos no es suficiente para representar objetos curvos y para estos casos se utilizan las superficies. De esta forma, Thompson II (1997) utiliza un escenario representado mediante superficies NURBS (Non-Uniform Rational B-Splines). En primer lugar, se utilizan cajas contenedoras para determinar si el punto que representa al dispositivo (HIP) se encuentra en colisión. Si existe colisión, el HIP se proyecta en la superficie y la distancia entre el HIP y el punto proyectado proporciona la magnitud de la fuerza a devolver.

De nuevo Thompson propone una extensión del algoritmo anterior para que sea compatible con objetos virtuales definidos mediante trimmed-NURBS, que permiten definir un rango más amplio de escenarios (Thompson II et al. 1999).

Otros trabajos sin embargo, saben aprovechar al mismo tiempo las ventajas que aportan los modelos geométricos y las superficies implícitas (Kim et al. 2002a). Concretamente utilizan los modelos geométricos para representar la escena y las funciones implícitas para el cálculo de las fuerzas a restituir, representando la superficie con valores de potenciales en una malla 3D uniforme que proporcionan el valor de la penetración. La normal se calcula mediante el gradiente de la función implícita. Gracias a las propiedades de las superficies implícitas, se consigue un contacto suave y sin discontinuidades.

A diferencia de trabajos anteriores, este método calcula la posición de un punto de contacto virtual restringido a la superficie del objeto, de forma que se permite al usuario sentir la geometría real del objeto. Además del cálculo de la fuerza a restituir, el algoritmo simula de forma realista propiedades de las superficies como la fricción, la rigidez y las texturas.

Existen métodos más recientes que siguen esta filosofía de penalización y proporcionan una respuesta háptica de 6 gdl (Kim et al. 2003, McNeely et al. 1999, Otaduy et al. 2003). Estos trabajos serán analizados en el apartado 2.2.3.

2.2.2.2 MÉTODOS BASADOS EN RESTRICCIONES

A diferencia de los métodos de penalización, los métodos basados en restricciones no utilizan las fuerzas de interpenetración entre objetos rígidos para calcular la respuesta de colisión. Estos métodos utilizan técnicas de

acoplamiento virtual, y restringen el objeto virtual desacoplado en la superficie de los obstáculos, gobernando su dinámica mediante determinadas restricciones.

El primer método de respuesta de colisión basado en restricciones fue desarrollado por Zilles y Salisbury (1995), y bautizado como método del *god-object*. Además de utilizar el punto de interacción con el entorno (HIP), se utiliza al mismo tiempo otro punto denominado *god-object* que puede definirse como una representación idealizada de la posición del háptico restringida a la superficie del modelo. Su posición se corresponde a la que tendría el HIP si la rigidez del entorno virtual fuese infinita. Lo novedoso de su método residía en la manera en la que restringían el movimiento del *god-object* en la superficie de los objetos virtuales contra los que colisionaba. En cada instante de tiempo, el algoritmo calcula la localización del *god-object* como el punto que minimiza la distancia entre dicho objeto virtual y la posición del háptico, sujeto a unas restricciones impuestas por la geometría del objeto con el que se ha colisionado.

Para calcular la posición del *god-object*, los autores minimizan la energía de un muelle virtual de rigidez unidad que une las posiciones del háptico y del *god-object*. La Ecuación (2.3) muestra dicha expresión, donde x_G, y_G, z_G son las coordenadas desconocidas de la posición del *god-object*, y x_H, y_H, z_H son las coordenadas del háptico en el instante actual:

$$Q = \frac{1}{2}(x_G - x_H)^2 + \frac{1}{2}(y_G - y_H)^2 + \frac{1}{2}(z_G - z_H)^2 \quad (2.3)$$

Las restricciones al movimiento se añaden en forma de planos sobre los que el *god-object* puede moverse, pero no penetrar:

$$A_n x_G + B_n y_G + C_n z_G - D_n = 0 \quad (2.4)$$

Se deben incluir tantas ecuaciones de restricción como superficies en contacto haya entre el *god-object* y el obstáculo virtual.

A partir de las Ecuaciones (2.3) y (2.4), se pueden utilizar los multiplicadores de Lagrange, tal que la nueva posición del *god-object* se obtiene minimizando L en la Ecuación (2.5). En esta ecuación la variable i representa el número de superficies distintas en contacto.

$$L = \frac{1}{2}(x_G - x_H)^2 + \frac{1}{2}(y_G - y_H)^2 + \frac{1}{2}(z_G - z_H)^2 + \sum_{i=1}^n \lambda_i (A_i x_G + B_i y_G + C_i z_G - D_i) \quad (2.5)$$

Por último, la fuerza que se restituye al usuario será proporcional a la elongación del muelle virtual de acoplamiento, siendo el parámetro k la constante de rigidez de dicho muelle:

$$\mathbf{F} = k(\mathbf{x}_G - \mathbf{x}_H) \quad (2.6)$$

La Figura 2.8 muestra un ejemplo del cálculo de la posición \mathbf{x}_G del *god-object* resultante de una penetración del háptico \mathbf{x}_H en una superficie plana, y la fuerza restituida al usuario.

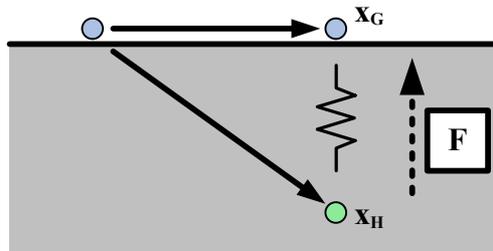


Figura 2.8: Cálculo de la posición del *god-object*.

Sin embargo, el uso de un punto como representación del dispositivo háptico restringe el tipo de superficies virtuales que pueden simularse de una manera efectiva. Un claro ejemplo son las superficies generadas por polígonos muy pequeños, en los que el punto que representa al *god-object* puede caer en una grieta o imperfecciones del mallado provocadas por errores matemáticos de redondeo.

En un trabajo posterior, Ruspini et al. (1997) propusieron un método similar al de Zilles y Salisbury al que llamaron “*virtual proxy*”, y en el que el *god-object* (un punto) era sustituido por un objeto esférico cuyo radio debe ser lo suficientemente grande para evitar las imperfecciones del mallado. Además, los autores añadieron otras propiedades al método, como la capacidad de restituir fricción en el movimiento del objeto esférico. Trabajos posteriores también han seguido esta misma filosofía (Niemeyer et al. 2004, Walker et al. 2003).

En general, los métodos basados en restricciones funcionan muy bien cuando existen pocas superficies de contacto, por ejemplo una pared. A diferencia de los métodos de penalización, no es necesario calcular la interpenetración entre objetos en cada instante, y las fuerzas y pares restituidos al usuario dependen de las posiciones del objeto virtual desacoplado y el dispositivo háptico, que son calculadas en el módulo de control. El módulo de colisiones calcula la normal de la superficie de contacto y el punto de contacto, que en el caso de una única superficie no varían significativamente. Sin embargo, al recorrer objetos redondeados, cada triángulo del objeto se convierte prácticamente en un plano de restricción. En estos casos la información varía permanentemente, y la sensación táctil con estos métodos es que la restitución de fuerzas va con retraso respecto al movimiento cinestético realizado por el usuario. Tampoco es muy buena la sensación táctil en esquinas convexas, o superficies en las que la dirección de la normal cambia abruptamente.

En general, uno de los mayores problemas de este método y de los métodos de penalización anteriormente analizados, radica en que no son válidos para ser utilizados en aplicaciones en las que es necesario realizar giros o tocar varios objetos simultáneamente. Para solucionar este problema, Ho et al. (2000) propusieron sustituir el punto que representaba el *god-object* por una línea o rayo, mediante el que eran capaces de restituir pares de contacto al usuario. A este tipo de métodos en los que el objeto virtual se representa mediante una línea finita en vez de por un punto, se les denomina métodos basados en un rayo. La aplicación de los métodos basados en un rayo en aplicaciones como la medicina tiene ciertas ventajas con respecto a los métodos basados en un punto. Un claro ejemplo es el caso de la cirugía mínimamente invasiva en el que el cirujano introduce finas y largas herramientas en el cuerpo del paciente, en cuyos extremos se fijan pequeños utensilios para la manipulación de órganos y tejidos. Durante la operación, el médico accede a la zona dañada mediante estas herramientas que le ayudan a sentir las fuerzas y pares al tocar o separar los tejidos. Si la herramienta estuviese modelada mediante un punto, sería imposible detectar las colisiones laterales que se producen con los órganos y se podría atravesar cualquier órgano o tejido que no estuviese en contacto con la punta de dicha herramienta. Varios trabajos han constatado un incremento notable del realismo en simulaciones de cirugía laparoscópica (Basdogan et al. 2004, Basdogan et al. 1998).

2.2.2.3 MÉTODOS BASADOS EN IMPULSOS

Los algoritmos de renderizado basados en impulsos son los menos utilizados en la bibliografía, y como su nombre indica, restituyen fuerzas en forma de impulsos en los eventos de colisión (Mirtich et al. 1994). Las colisiones continuas son tratadas como series de pequeñas colisiones, en cada una de las cuales se aplican los impulsos para evitar la interpenetración entre objetos. Los métodos basados en impulsos producen resultados visualmente aceptables, pero hápticamente poco convincentes (Chang et al. 1997). Por ello, pueden utilizarse en combinación con los métodos de penalización (Constantinescu et al. 2005).

Tal y como indicó Mirtich, este tipo de métodos están más orientados para simulaciones en las que se producen colisiones entre numerosos objetos simultáneamente. Sin embargo, esta Tesis se centra en calcular una respuesta de colisión entre un escenario y una herramienta virtual con la que el usuario realiza una tarea establecida o intenta encontrar un posible camino para acceder a un determinado elemento de la escena. Por este motivo, este tipo de entornos no son los más adecuados para este tipo de métodos además de considerarse que no aportarían una solución realista al problema propuesto.

2.2.3 RESPUESTA DE 6 GDL

Los sistemas hápticos se utilizan cada vez más como simuladores de entrenamiento en aplicaciones tan diferentes como la industria o la medicina. Dichos campos de aplicación exigen cada vez un mayor grado de realismo y precisión que no es alcanzable por los métodos basados en un punto o en rayos descritos anteriormente. Por este motivo, los métodos más recientes utilizan toda la información geométrica del objeto móvil para garantizar una respuesta estable, precisa y de 6 gdl. Debido a la complejidad del problema y a su reciente implantación en sistemas hápticos, el número de trabajos existentes en la bibliografía es reducido.

Los métodos que se describen en los siguientes subapartados se han clasificado en base a la representación utilizada para definir el objeto móvil: polígonos, voxels o superficies.

El área de los gráficos por computador ha propuesto e investigado una amplia variedad de técnicas para representar objetos 3D. Una de las representaciones más utilizadas consiste en usar geometría (triángulos o polígonos) para definir la superficie del objeto (Figura 2.9a). Concretamente

el modelo geométrico más utilizado en la bibliografía del cálculo de la respuesta de colisión y el utilizado en esta Tesis, es el B-Rep o representación frontera mediante facetas planas en el que el objeto es definido por las superficies que delimitan el volumen de trabajo. La B-Rep basada en caras planas es una representación muy frecuente para el intercambio de información entre diferentes aplicaciones de CAD debido a la gran cantidad de formatos que soporta esta representación. Otra ventaja es el hecho de que la mayoría de las tarjetas gráficas ofrecen aceleraciones hardware si se utilizan polígonos como representación geométrica.

El segundo tipo de representación es el basado en superficies que permite definir objetos curvos con gran naturalidad eliminando las limitaciones de los modelos poligonales (Figura 2.9b). Dos de las superficies más utilizadas son las superficies implícitas y NURBS.

Por último, la representación mediante volúmenes consiste en descomponer o subdividir el sólido en una colección de entidades geométricas más simples como pueden ser cubos, paralelepípedos, etc. Las representaciones más usadas dentro de este grupo son los voxels que particionan el espacio de forma uniforme (Figura 2.9c), y los octrees donde la subdivisión se lleva a cabo jerárquicamente.

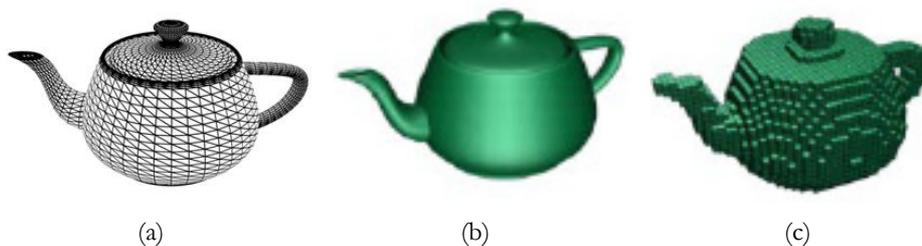


Figura 2.9: Modelo de una tetera representado mediante una malla de triángulos (a), superficies (b) y voxels (c).

2.2.3.1 MODELOS POLIGONALES

Los métodos que se describen a continuación utilizan representaciones poligonales para definir los objetos virtuales. Generalmente, se utilizan mallados de triángulos y en base a los triángulos en colisión detectados se calcula la información necesaria para garantizar una respuesta estable.

Gregory et al. (2000b) por ejemplo, descompone en preproceso los modelos de polígonos utilizados en el entorno virtual en primitivas convexas. El cálculo de colisiones está basado en el método desarrollado en

(Lin et al. 1991), algoritmo que saca provecho de la coherencia temporal y la localidad espacial para predecir los contactos. Para disminuir la frecuencia para el cálculo de la penetración, los autores utilizan una técnica predictiva en la que se establece que existe colisión cuando la distancia entre los dos objetos es menor que una tolerancia preestablecida. Se lleva a cabo también una interpolación de las fuerzas y los pares para evitar saltos bruscos en las fuerzas de contacto que puedan introducir discontinuidades en *frames* sucesivos. Sin embargo, este método no calcula una buena estimación de la penetración cuando se trata de objetos no convexos, provocando problemas de estabilidad en dichos casos.

Para evitar estas discontinuidades, el trabajo propuesto por Kim et al. (2003) utiliza una técnica basada en clusters para el cálculo de los contactos y fuerzas. Su algoritmo únicamente trabaja con modelos poligonales convexos, por lo que al igual que en el caso anterior todos los objetos se descomponen en elementos convexos tras una etapa de preproceso. Como resultado de esta descomposición, se obtiene un número elevado de puntos de contacto que puede originar problemas derivados de un valor de rigidez elevado. Para evitarlo, estos puntos de contacto se agrupan en clusters en base a su proximidad espacial. Para cada uno de estos clusters se calcula su posición, la distancia y la dirección de la fuerza realizando una media ponderada en base a la penetración de cada uno de los puntos que forman parte de ese contacto. Finalmente, se obtienen las fuerzas y los pares resultantes sumando las fuerzas y pares calculados para cada uno de los clusters identificados.

Sin embargo, este método presenta problemas de discontinuidad en la respuesta provocados por la geometría de los modelos. La posible falta de continuidad se debe a que los puntos de contacto asociados a cada cluster pueden aparecer y desaparecer en *frames* consecutivos. Un claro ejemplo es la simulación de la rotación de un cubo sobre una superficie (Figura 2.10), donde la discontinuidad en el punto de contacto provoca que el objeto oscile continuamente.

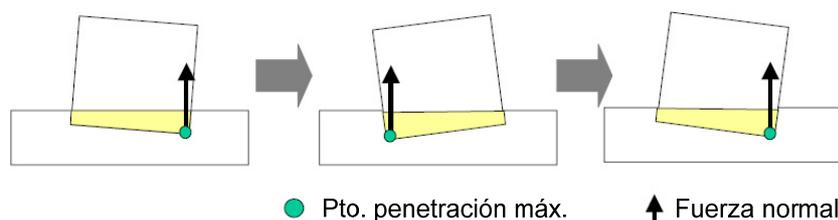


Figura 2.10: Ejemplo de discontinuidad en los pares.

Para evitar estos problemas, Hasegawa et al. (2004) proponen distribuir un modelo de muelle y amortiguamiento en todo el área del contacto. De esta forma se consigue un cambio continuo en los puntos de aplicación y por lo tanto en las normales de contacto (Figura 2.11). Sin embargo, esta aproximación es sólo aplicable a objetos muy sencillos y convexos.

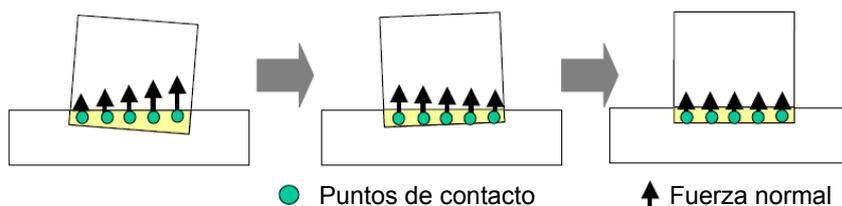


Figura 2.11: El área de contacto se define mediante varios puntos de contacto.

Por su parte, Otaduy et al. (2003) llevan a cabo una simplificación del modelo geométrico dando mayor prioridad a la velocidad que a la precisión en el cálculo de la respuesta. Esto les permite trabajar con maquetas de complejidad considerable con decenas de miles de polígonos, manteniendo la frecuencia de muestreo necesaria en este tipo de aplicaciones.

El método propuesto tiene una etapa de preproceso en la que descompone los objetos en elementos convexos y calcula una jerarquía multi-resolución en la que almacena diferentes aproximaciones del modelo original utilizando para ello diferentes niveles de detalle que los autores denominan CLOD (*Contact Level Of Detail*). Esto les permite seleccionar de forma adaptativa la resolución adecuada a cada zona de contacto, reduciendo el coste computacional del cálculo y la respuesta a dicha colisión.

Una vez detectados los contactos entre el objeto móvil y los CLODs que representan la escena, se calculan un conjunto de contactos representativos obtenidos mediante una técnica de clusters basada en (Jain et al. 1999) a partir de los cuales se calculan las fuerzas y momentos a restituir sobre el objeto virtual.

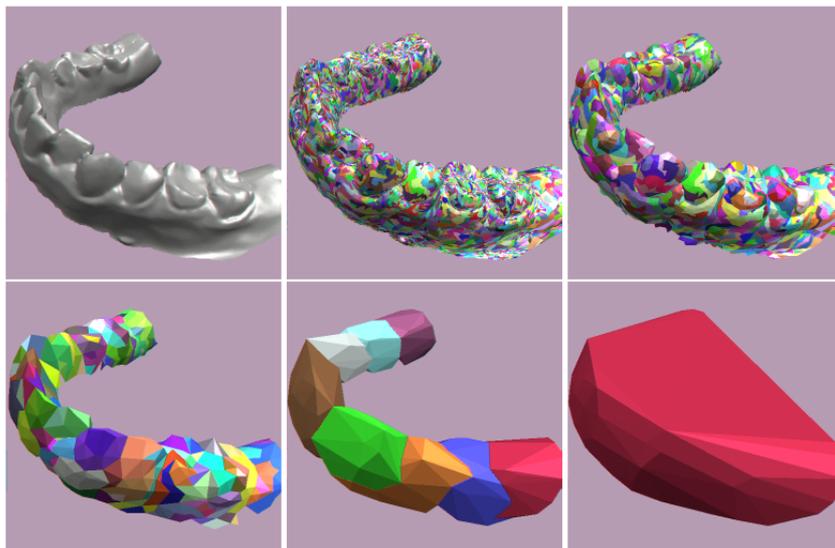


Figura 2.12: CLODs de una dentadura. En la parte superior izquierda se muestra el modelo original. La parte inferior derecha muestra el modelo simplificado, mientras que el resto representan diferentes resoluciones del mismo modelo.

La estabilidad del método también se ve mejorada con respecto a algoritmos anteriores puesto que consigue reducir la inestabilidad provocada por el acoplamiento virtual mediante la integración implícita de las ecuaciones del sólido rígido que definen su movimiento (Otaduy et al. 2005). Una información más detallada sobre este trabajo puede consultarse en (Otaduy 2004).

En (Johnson et al. 2003) se sigue una filosofía diferente para calcular la respuesta en un sistema de 6 gdl. En vez de separar los objetos una vez de que han interpenetrado, se previene la colisión utilizando para ello una envolvente para cada objeto que interactúa en el entorno. De esta forma se consigue evitar el cálculo de las penetraciones que resulta generalmente muy costoso. Las fuerzas resultantes se calculan en base a los puntos más cercanos entre ambos objetos y las distancias mínimas calculadas mediante

la técnica SNCH (*Spatialised Normal Cone Hierarchies*) desarrollada en (Johnson et al. 2001).

Y más recientemente, Ortega et al. (2006) proponen la ampliación del método original del *god-object* de 3 gdl (fuerzas) hasta los 6 gdl (fuerzas y pares). A continuación se describe brevemente en qué consiste dicha ampliación. Sean \mathbf{a}^u y \mathbf{a}^c los vectores que representan en cada iteración el movimiento (posición y orientación) libre y restringido respectivamente para el *god-object*. El movimiento libre será aquel que mueva al *god-object* desde la posición anterior a la posición del háptico, y el restringido el que queremos calcular (Figura 2.13a). La Ecuación (2.7) muestra sus expresiones, donde \mathbf{x} representa el vector traslación y \mathbf{r} el vector orientación en notación eje-ángulo.

$$\begin{aligned}\mathbf{a}^u &= (\mathbf{x}^u, \boldsymbol{\alpha}^u) = (\mathbf{x}_H - \mathbf{x}_{G_ant}, \mathbf{r}_H - \mathbf{r}_{G_ant}) \\ \mathbf{a}^c &= (\mathbf{x}^c, \boldsymbol{\alpha}^c) = (\mathbf{x}_G - \mathbf{x}_{G_ant}, \mathbf{r}_G - \mathbf{r}_{G_ant})\end{aligned}\quad (2.7)$$

De manera similar al método original, la solución al problema vendrá de la minimización de la función (2.8):

$$Q = \frac{1}{2} \|\mathbf{a}^c - \mathbf{a}^u\|^2 \quad (2.8)$$

Para restringir tanto las posibles traslaciones como orientaciones que puedan hacer que el *god-object* penetre en las superficies de contacto, los autores proponen minimizar (2.8) teniendo en cuenta la ecuación de restricción definida por la expresión (2.9).

$$\mathbf{x}^u \mathbf{n}_i + \boldsymbol{\alpha}^u (\mathbf{GI}_i \wedge \mathbf{n}_i) \geq 0 \quad (2.9)$$

En esta ecuación, \mathbf{GI} es un vector que une el centro de inercia del objeto virtual con el punto de contacto de colisión y \mathbf{n} es el vector normal de cada superficie i en contacto con el *god-object*.

Por último, las fuerzas y pares que deben devolverse al usuario vienen dadas por la Ecuación (2.10), donde k es la constante de rigidez del acoplamiento virtual, y puede tener distinto valor para las fuerzas y los pares (Figura 2.13b).

$$\mathbf{F} = k(\mathbf{a}^c - \mathbf{a}^u) \quad (2.10)$$

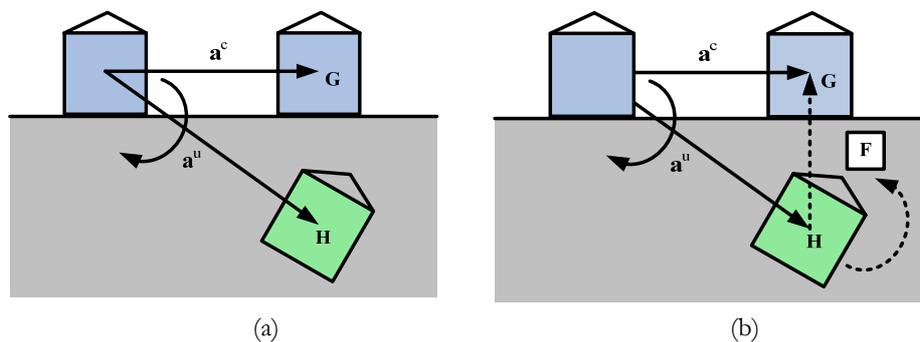


Figura 2.13: Cálculo de la posición G del *god-object* (a) y fuerzas y pares **F** restituidos al usuario (b).

2.2.3.2 MODELOS DEFINIDOS MEDIANTE VOXELS

Diversos trabajos han optado por subdividir el entorno virtual con una malla de voxels con la finalidad de simplificar el cálculo de las intersecciones como de las penetraciones entre dos objetos.

El trabajo desarrollado por McNeely et al. (1999) para la compañía Boeing, realiza tareas de mantenimiento en modelos masivos de motores aeronáuticos con un único objeto móvil o herramienta. En este trabajo se utilizan dos estructuras diferentes que reemplazan a las representaciones poligonales de los objetos (Figura 2.14). La primera denominada *Voxmap* es una estructura de partición espacial basada en voxels que sustituye al modelo geométrico de la maqueta virtual. Por otro lado, la estructura correspondiente al objeto móvil se denomina *Point Shell*, y consiste en una nube de puntos que definen la superficie del objeto. Cada uno de estos puntos tiene asociado una normal dirigida hacia el interior del objeto.

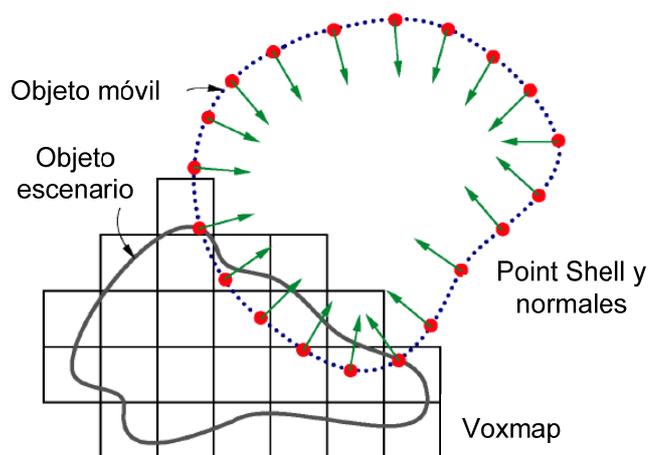


Figura 2.14: Voxmap de un objeto estático colisionando con el Point Shell de un objeto móvil.

El uso de estas dos estructuras de datos facilita el cálculo de la normal resultante y la penetración del objeto móvil con los objetos que tiene alrededor. La dirección normal de contacto \mathbf{n} es calculada como la suma vectorial normalizada de las normales asociadas a los puntos interiores. Por su parte, la penetración se calcula como la distancia existente entre el punto en colisión del objeto móvil al plano que contenga el punto central del voxel y sea perpendicular a la normal de contacto (Figura 2.15). Finalmente, la fuerza y el par resultante que debe aplicarse al objeto móvil se obtiene sumando las fuerzas obtenidas para cada intersección entre punto y voxel.

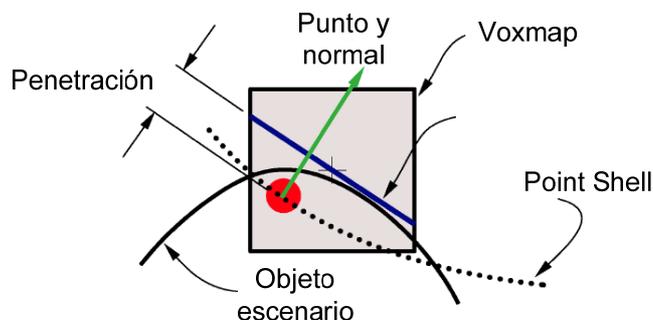


Figura 2.15: Cálculo de la penetración mediante los Voxmap-PointShell

El punto débil de esta estrategia es la precisión que posee. Al utilizar la representación de la superficie de los objetos basada en voxels, el método

no es capaz de calcular penetraciones profundas que pueden producirse en caso de que dos objetos colisionen a una velocidad alta. Otro ejemplo puede observarse en la Figura 2.16 en el que el punto del *Point Shell* está por encima del plano tangente y por lo tanto el método concluye que no existe penetración.

Para evitar estos casos, se aplican fuerzas antes de que se produzca la colisión reduciendo así la velocidad del contacto y por lo tanto, evitando de este modo penetraciones elevadas.

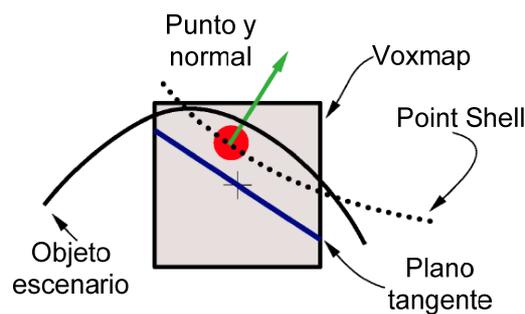


Figura 2.16: Error en el cálculo de la penetración mediante los Voxmap-PointShell.

Con respecto a la estabilidad del sistema, ésta puede verse afectada cuando existen múltiples puntos de contacto simultáneamente que provocan que la rigidez del sistema aumente de forma brusca produciendo discontinuidades en las fuerzas resultantes. Dichas discontinuidades se reducen limitando la rigidez y utilizando la técnica del acoplamiento virtual.

En trabajos posteriores se han desarrollado notables mejoras al algoritmo. En primer lugar, se consigue optimizar la precisión en el cálculo proyectando los puntos del Point Shell a los triángulos de la superficie del objeto mejorando tanto la precisión como la estabilidad en las fuerzas de contacto (Renz et al. 2001). También cambia el modelo de acoplamiento virtual utilizado en el trabajo anterior, permitiendo un correcto funcionamiento del algoritmo independientemente del dispositivo háptico utilizado.

En Wan et al. (2003), se plantea una nueva solución para el cálculo de la posición del objeto móvil con el objetivo de evitar las inestabilidades producidas por bruscos incrementos de la rigidez en la fuerza resultante. Se reemplaza el modelo dinámico empleado para calcular la posición y la orientación del objeto móvil basado en el método de integración explícito

de Newton-Euler, utilizado en McNeely et al. (1999). El nuevo algoritmo mejora la respuesta táctil planteando un equilibrio entre las fuerzas de colisión que actúan sobre el objeto móvil y las fuerzas externas aplicadas al acoplamiento virtual. Este nuevo método permite mantener la frecuencia de muestreo a 1kHz, y consigue mejorar la sensación de la fuerza y el par resultantes.

Las mejoras añadidas en el último trabajo publicado (McNeely et al. 2006), permiten la aplicación de la simulación háptica de 6 gdl en tareas reales de gran complejidad gracias al uso tanto de la coherencia espacial como de la temporal. Para ello los autores extienden la voxelización del objeto estático más allá de la superficie marcando los nuevos voxels con un valor que representa una estimación de la distancia de cada voxel a la superficie en unidades del tamaño de voxel establecido. También se calcula la distancia máxima que puede recorrer un punto en cada *frame*, de forma que se consiguen evitar los cambios bruscos en el movimiento reduciendo así la inestabilidad en el sistema.

Un punto de vista diferente del uso de los voxels para la detección de colisiones en sistemas hápticos es el propuesto en (Borro et al. 2004a). La subdivisión en voxels no reemplaza a la geometría que define los modelos sino que se utiliza para acelerar la detección de las intersecciones entre objetos.

2.2.3.3 MODELOS DEFINIDOS MEDIANTE SUPERFICIES

La representación implícita de una superficie externa S de un objeto se describe con la siguiente ecuación:

$$S = \{(x, y, z) \in \mathbb{R}^3 \mid f(x, y, z) = 0\} \quad (2.11)$$

donde \mathbf{f} es la función implícita también conocida como potencial, y (x, y, z) las coordenadas del punto en el espacio 3D.

Si el valor del potencial es 0, se sabe que el punto (x, y, z) pertenece a la superficie. En cambio, si el valor es positivo, entonces el punto es exterior a la superficie y en caso contrario, el punto se encuentra dentro de la superficie. Las normales a la superficie por su parte, son fáciles de calcular utilizando el gradiente de la función implícita.

El principal inconveniente del uso de este tipo de superficies es la dificultad de controlar su forma, por lo que las NURBS se presentan como

la mejor opción para definir objetos complejos. Este tipo de superficies permite un control local de la geometría de la curva modificando para ello sus puntos de control. Las superficies NURBS adoptan la siguiente expresión:

$$\mathbf{S}(\mathbf{u}, \mathbf{v}) = \frac{\sum_{i=0}^n \sum_{j=0}^m \mathbf{N}_{i,p}(u) \mathbf{N}_{j,q}(v) w_{i,j} \mathbf{P}_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m \mathbf{N}_{i,p}(u) \mathbf{N}_{j,q}(v) w_{i,j}} \quad (2.12)$$

Donde $w_{i,j}$ son los pesos asociados a los puntos de control $\mathbf{P}_{i,j}$ que definen la superficie. $\mathbf{N}_{i,p}(u)$ y $\mathbf{N}_{j,q}(v)$ son las funciones B-spline definidas sobre las secuencias de nodos u y v .

La representación de los objetos mediante superficies permite definir objetos curvos con gran naturalidad eliminando las limitaciones de los modelos poligonales. Existen técnicas en la bibliografía que utilizan esta representación en el cálculo del renderizado háptico aunque es un campo que no ha sido estudiado en profundidad. En esta Tesis, este tipo de representación se ha descartado ya que todos los modelos utilizados vienen representados mediante mallas de triángulos.

Uno de los problemas de los métodos que utilizan superficies consiste en calcular las distancias mínimas y los valores de penetración entre las diferentes superficies a una frecuencia adecuada para las aplicaciones hápticas. Turnbull et al. (1998) modifica el algoritmo GJK (Gilbert et al. 1988) para utilizar superficies convexas definidas mediante NURBS. De forma análoga, Lin et al. (1993) presentan un algoritmo para modelos curvos definidos mediante superficies Spline.

Nelson et al. (1999) desarrollan un método que calcula las mínimas distancias y valores apropiados de penetración entre las dos superficies. Finalmente, Patoglu et al. (2004) desarrollan un algoritmo para determinar las distancias entre superficies paramétricas convexas. Este método también es aplicable a objetos cóncavos ya que éstos pueden aproximarse mediante superficies paramétricas convexas, al igual que se descomponen objetos poligonales cóncavos en elementos convexas.

2.3 ALGORITMO DE CONTROL

Una vez que el módulo de respuesta de colisión ha calculado la fuerza ideal de interacción entre la herramienta y el entorno virtual, estas fuerzas deben ser aplicadas al usuario. Una de las tareas principales del módulo de control es adaptar estas fuerzas a las capacidades reales del dispositivo háptico antes de restituírselas al usuario. Son varios los factores que limitan la capacidad de un dispositivo háptico para ejercer las fuerzas deseadas. Por ejemplo, la fricción e inercia del dispositivo o el nivel de saturación de los actuadores.

El objetivo del lazo de control es garantizar una respuesta háptica estable al usuario. Para ello, este lazo afronta fundamentalmente dos problemas: la actuación en ausencia de información de contacto y la transición al nuevo contacto recibido. Estos problemas derivan del hecho de que el control tiene una frecuencia de muestreo mayor que la frecuencia del cálculo de las colisiones. Según los estudios realizados (Shimoga 1992), la mayoría de autores eligen una frecuencia de muestreo entre 500 Hz y 1 kHz para el módulo de control, ya que permiten mantener el sistema estable y cubren todos los efectos que se pueden simular en el tacto humano. Sin embargo, el módulo de colisiones no tiene una frecuencia tan alta y tampoco un periodo de muestreo fijo ya que depende de la complejidad geométrica de los modelos en colisión. Esta diferencia entre los diferentes periodos de muestreo obliga al control a dar una respuesta adecuada en aquellos periodos de muestreo en los que carezca de información de colisión (Adachi et al. 1995, Mark et al. 1996, Volino et al. 1997).

Otro factor importante a tener en cuenta es que los algoritmos de renderizado trabajan en tiempo discreto, mientras que el usuario trabaja en tiempo continuo. Esto implica que factores como el muestreo de la posición del usuario o el error de cuantización en los sensores puedan generar energía extra en el sistema en los eventos de colisión (Gillespie et al. 1996), provocando una respuesta inestable del dispositivo. Existen diversos trabajos en la bibliografía que han analizado los factores que influyen en la estabilidad de los interfaces hápticos (Colgate et al. 1994, Gil et al. 2007).

El lazo de control también es el encargado de añadir fricción al modelo y rigidez al entorno virtual para aumentar el realismo de la respuesta háptica.

CAPÍTULO 3

DETECCIÓN DE COLISIONES

Como ya se ha explicado en el capítulo 2, los algoritmos de renderizado háptico constan de tres módulos: detección de colisiones, respuesta a dicha colisión y el módulo de control. Este capítulo se centra en el primer módulo mencionado, el cálculo de las colisiones.

Los problemas de accesibilidad que se mencionan en esta Tesis pueden encontrarse en aplicaciones tan diferentes como los simuladores de entrenamiento en medicina o en aplicaciones industriales de mantenimiento. El primer paso para poder simular este tipo de tareas de forma adecuada es detectar si se produce una colisión con el entorno y calcular la información necesaria para obtener una respuesta háptica al evento de colisión.

Los modelos geométricos que se emplean en este trabajo están representados mediante B-Rep (*Boundary Representation*). Es decir, un objeto 3D está definido por una lista de triángulos que delimitan su superficie. Se ha tomado como punto de partida este modelo poliédrico ya que es el más utilizado en los sistemas CAD para exportar descripciones geométricas.

En este capítulo se describe el módulo de detección de colisiones desarrollado en (Borro 2003) así como las mejoras que se han introducido para obtener la información necesaria para calcular una respuesta háptica estable tras la colisión. Al utilizar modelos representados mediante mallas de triángulos, esta información vendrá dada por una lista de triángulos asociada a cada uno de los objetos en colisión.

3.1 DESCRIPCIÓN GENERAL DEL PROBLEMA

El objetivo de la detección de colisiones es informar automáticamente cuándo se detecta algún contacto geométrico con el fin de evitar que dos sólidos ocupen el mismo espacio en un mundo virtual.

El método más simple para detectar si existe colisión entre dos objetos definidos mediante una malla de triángulos consiste en testear cada triángulo del primer objeto con cada triángulo del segundo objeto. Sin embargo, este método de fuerza bruta requiere de un coste computacional muy elevado $O(n^2)$, ya que su complejidad aumenta de forma cuadrática con el número de triángulos con los que se define cada objeto. Este coste computacional es inaceptable en sistemas hápticos en los que se necesitan altas frecuencias de muestreo de la información.

Por lo tanto, para realizar una detección de colisiones eficiente para aplicaciones en tiempo real, es necesario acotar la zona de búsqueda a aquellas zonas donde realmente los objetos se encuentren cercanos, para posteriormente realizar un cálculo de colisiones exacto en dichas zonas. En esta Tesis se ha utilizado el módulo de colisiones desarrollado en el departamento de Mecánica Aplicada del CEIT (Borro 2003), en el que se utiliza una partición espacial basada en voxels para reducir la complejidad del problema. Se ha de remarcar que esta subdivisión no sustituye en ningún momento al modelo geométrico original del objeto estático, como es el caso de otros trabajos (McNeely et al. 1999).

Una vez acotada la zona, se calcula la intersección entre los dos objetos obteniendo las listas de triángulos que se intersectan para cada uno de ellos. Sin embargo, un cálculo de colisiones con un coste computacional reducido no es suficiente para garantizar una restitución háptica estable si no se calcula la información necesaria. El módulo de respuesta basa sus cálculos en la información enviada por el módulo de detección por lo que las listas de triángulos detectadas deben proporcionar una información detallada de la colisión producida.

En este trabajo se sigue la estrategia basada en los métodos de penalización que calculan la fuerza a restituir en base a una dirección y a la magnitud de interpenetración entre los dos objetos. Esta distancia es conocida como penetración y es uno de los factores que condicionan la estabilidad final de la respuesta.

En los apartados sucesivos se describe brevemente el módulo de colisiones original y las modificaciones que se han tenido que realizar para

adecuarlo a los algoritmos de respuesta de colisión planteados en sucesivos capítulos.

3.2 CÁLCULO DE COLISIONES

El esquema general del método desarrollado por Borro se muestra en la Figura 3.1. Como se puede observar, el cálculo de las colisiones se divide en dos etapas. La primera se ejecuta una única vez en preproceso, y es la encargada de generar la división espacial mientras que la segunda se ejecuta en cada *frame* para determinar los triángulos que se encuentran en intersección.

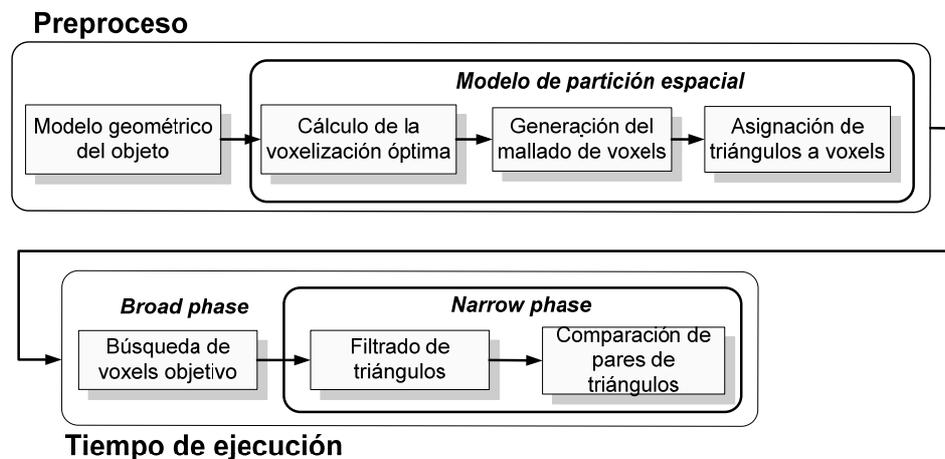


Figura 3.1: Esquema general del cálculo de colisiones.

En los dos siguientes apartados se explica cada uno de los bloques mostrados en el esquema del método.

3.2.1 ETAPA DE PREPROCESO

En esta etapa se genera la estructura de partición espacial basada en voxels a partir de la geometría original del modelo. Esta estructura de datos proporciona la información necesaria para acelerar la detección de colisiones entre los diferentes objetos que constituyen la escena reduciendo así la complejidad del problema.

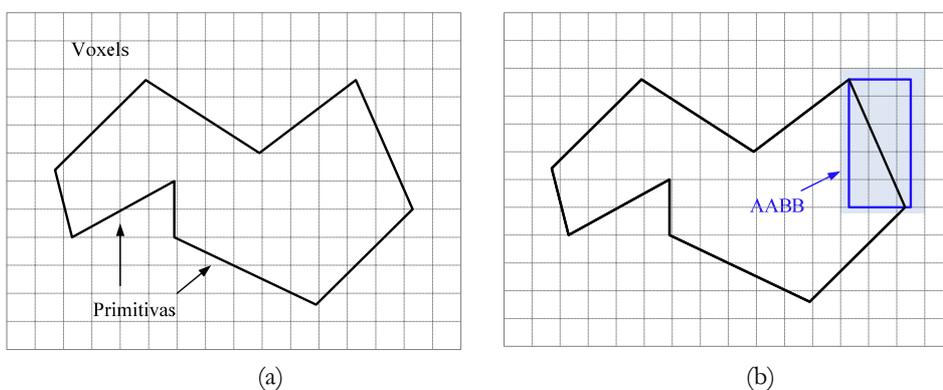
A diferencia de otros métodos, la partición espacial únicamente se lleva a cabo sobre la maqueta estática de la escena, utilizando como entrada su representación geométrica que en el caso de esta Tesis, se corresponde

con la representación B-Rep generada por un sistema CAD. Para el objeto móvil, se utiliza directamente su representación poligonal.

El primer paso para la creación de esta estructura de datos consiste en determinar el nivel de voxelizado óptimo para la maqueta estática. En su trabajo, Borro plantea en base únicamente a la geometría de la escena una solución analítica a este problema que garantiza un tiempo de ejecución mínimo (Borro et al. 2004a). Una vez calculado el tamaño del voxel con el que se divide la escena, se reserva la memoria necesaria para almacenar este mallado tras lo cual se procede a asignar a cada voxel los triángulos contenidos en él.

Para identificar qué voxels son intersectados por alguna primitiva del objeto, se podría comparar cada triángulo de la escena con cada celda de la malla de voxelizado. Sin embargo, esta solución tendría un elevado coste computacional. Para acelerar este cálculo e identificar en qué voxel está contenido, parcial o totalmente, cada uno de los polígonos pertenecientes a la frontera del objeto, el método analiza únicamente aquellos voxels que estén en colisión con el AABB del triángulo en curso.

Una vez detectada la intersección entre un triángulo y el voxel, se guarda el puntero del triángulo en curso en dicho voxel. De esta forma, cada voxel almacena una lista de punteros a los triángulos que lo intersectan. Las diferentes fases de este proceso se representan gráficamente mediante la Figura 3.2.



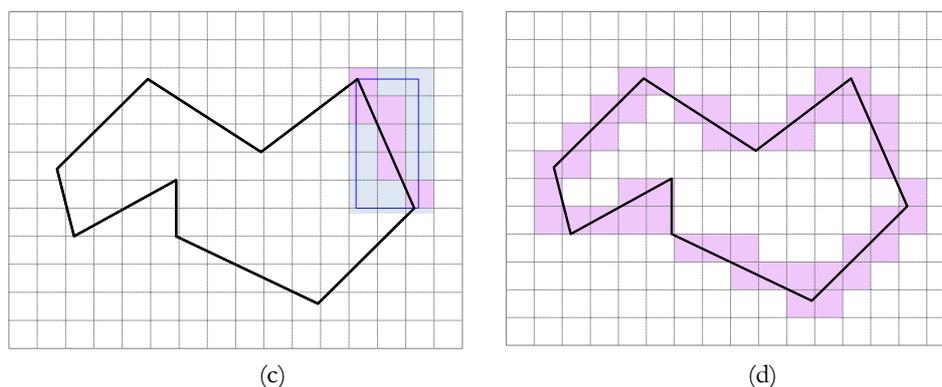


Figura 3.2: Representación 2D del proceso de voxelización. Reserva de memoria (a), selección de los voxels que intersectan con la caja contenedora de cada primitiva (b), cálculo de la intersección entre el triángulo y cada voxel seleccionado (c), los voxels marcados contendrán por lo menos una primitiva (d).

3.2.2 ETAPA EN TIEMPO DE EJECUCIÓN

La finalidad de esta etapa consiste en calcular en cada *frame* los elementos de la escena que se encuentran en colisión. Para ello, el método recorre la lista de triángulos del objeto móvil comparando cada uno de ellos con la malla de voxels del objeto estático definida en el apartado anterior. Esta etapa se divide en dos fases: *broad phase* y *narrow phase*.

El objetivo de la *broad phase* es la de encontrar aquellos objetos con probabilidades de colisión, descartando todas las zonas de trabajo sin interés. Por cada triángulo del objeto móvil se testea qué conjunto de voxels lo intersectan, de forma que se consigue acotar el problema de la detección de la colisión a un conjunto reducido de voxels. A este conjunto reducido se le llamará voxels objetivo (V_o).

A partir del triángulo móvil t utilizado en la *broad phase* y el conjunto de voxels objetivo detectados, la *narrow phase* es la encargada de detectar los triángulos del objeto estático contenidos en V_o que intersectan con t . Para acelerar el proceso, en primer lugar se analizan si los AABBs de los triángulos estáticos contenidos en V_o intersectan realmente con t . Si es así, finalmente se calcula la intersección de los triángulos contenidos en V_o con t mediante el método de Möller (1997). Una vez analizados todos los triángulos contenidos en V_o , se vuelve al inicio del ciclo para calcular otro conjunto de voxels objetivo para el siguiente triángulo móvil. La Figura 3.3 muestra de forma esquemática los pasos que se siguen en esta fase.

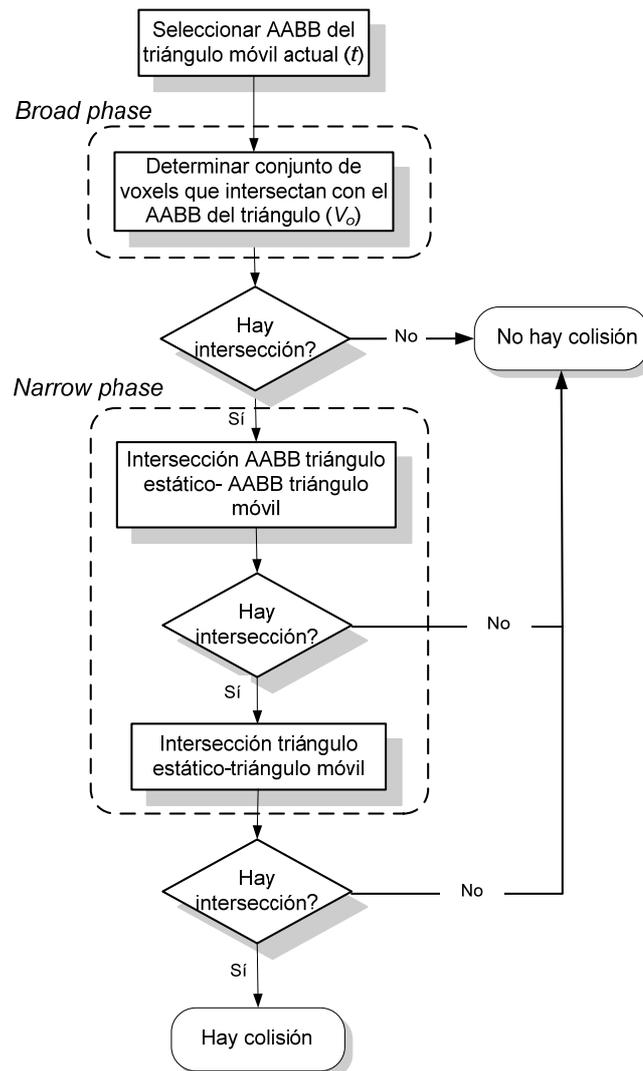


Figura 3.3: Esquema del cálculo de colisiones en tiempo de ejecución.

Esta etapa proporciona como salida las dos listas de triángulos en colisión asociadas al objeto estático y móvil respectivamente.

3.3 MODIFICACIONES REALIZADAS

El método diseñado e implementado por Borro obtiene excelentes resultados en la detección de colisiones en entornos muy densos, con maquetas definidas por millones de triángulos. Las listas de triángulos en

intersección calculadas por este módulo permiten devolver al usuario una respuesta visual de forma casi inmediata y además estas listas se utilizan para calcular una respuesta háptica en forma de fuerza adecuada en situaciones de baja complejidad.

Sin embargo, este método presenta algunas limitaciones para poder ser utilizado en aplicaciones en las que además de sentir fuerzas es necesario obtener una restitución de pares. Principalmente, las limitaciones se deben a que las estructuras de datos utilizadas están orientadas al cálculo de fuerzas. Además, la información calculada en situaciones de discontinuidad y contactos múltiples resulta insuficiente para el cálculo de una respuesta háptica agradable y estable para dichas situaciones. Por este motivo, ha sido necesario modificar el método original y calcular triángulos adicionales a las listas obtenidas por el módulo de colisiones.

En los siguientes apartados se estudia la forma de determinar esta información adicional necesaria para calcular una respuesta adecuada a colisiones de mayor complejidad. La Tabla 3.1 muestra un resumen de los conceptos que serán utilizados en el resto de este capítulo.

Nomenclatura	Significado
Triángulos en colisión estáticos	Triángulos del objeto estático que se encuentran en intersección con triángulos del objeto móvil. Dos triángulos se intersectan cuando pasan el test de Möller.
Triángulos en colisión móviles	Triángulos del objeto móvil que se encuentran en intersección con triángulos del objeto estático. Dos triángulos se intersectan cuando pasan el test de Möller.
Triángulos frontera estáticos	Triángulos del objeto estático contenidos en los voxels frontera pero que no están en colisión.
Triángulos frontera móviles	Triángulos del objeto móvil contenidos en voxels frontera del objeto estático pero que no son triángulos en colisión ni triángulos internos.
Triángulos internos móviles	Triángulos del objeto móvil contenidos en voxels internos del objeto estático. Estos triángulos están en el interior del objeto estático pero no están en colisión con su descripción poligonal.

Tabla 3.1: Resumen de los términos utilizados en el capítulo.

3.3.1 DETECCIÓN DE TRIÁNGULOS INTERNOS MÓVILES

Las listas de triángulos en colisión aportan una información limitada de la situación real de la colisión. Por este motivo, aquí se presenta un nuevo punto de vista. Cuando un objeto penetra en otro, la intersección entre ambos queda definida por un volumen de intersección delimitado por las superficies de los dos objetos. Este volumen es el utilizado por los métodos analizados en el capítulo 5 para calcular una respuesta háptica al evento de colisión.

Los triángulos en colisión detectados por el módulo de colisiones son insuficientes para delimitar este volumen ya que en la mayoría de los casos no consideran los triángulos que quedan contenidos en el interior de la maqueta estática. La Figura 3.4a muestra un ejemplo de este tipo de situaciones en las que se muestran únicamente los triángulos en intersección para cada uno de los objetos.

Como se puede observar, estos triángulos no aportan información suficiente para calcular un valor correcto de penetración, obteniendo un valor inferior al real. Por lo tanto, para un cálculo correcto de la penetración es necesario detectar aquellos triángulos que definen el objeto móvil y que se encuentran en el interior del objeto estático. En adelante, a estos triángulos del objeto móvil contenidos en el interior del objeto estático se les denominará *triángulos internos móviles*. En la Figura 3.4b se muestran estos triángulos.

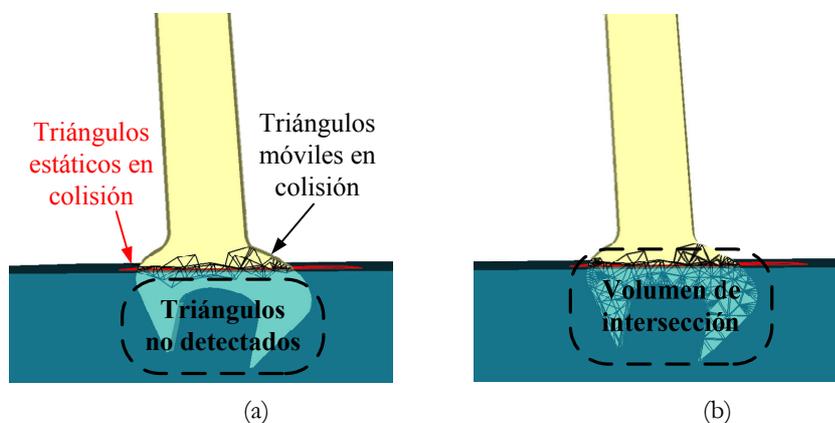


Figura 3.4: Herramienta virtual que colisiona con un objeto. En color rojo se representan los triángulos en colisión asociados al objeto estático y en negro los triángulos en colisión del objeto móvil (a). Se añaden los triángulos del objeto móvil interiores al objeto estático (b).

Ante este problema, se plantea una solución alternativa que consiste en utilizar una clasificación de voxels que facilite la detección de triángulos internos de forma correcta.

Esta clasificación se lleva a cabo en la etapa de preproceso. En esta fase, además de la asignación de los triángulos de la maqueta estática a los voxels a los que intersectan, se realiza también un proceso de clasificación de cada una de estas celdas que facilitará el cálculo de la respuesta de colisión. Esta clasificación distingue tres tipos diferentes de celdas: *externas, frontera e internas*.

El proceso de clasificación comienza clasificando todas las celdas como internas y una vez calculada la intersección entre los voxels y los triángulos del objeto estático, estas celdas se clasifican como frontera. Finalmente, es necesario realizar una última pasada para poder identificar las celdas exteriores.

La Figura 3.5 muestra un ejemplo de los pasos que se realizan para obtener la clasificación del voxelizado.

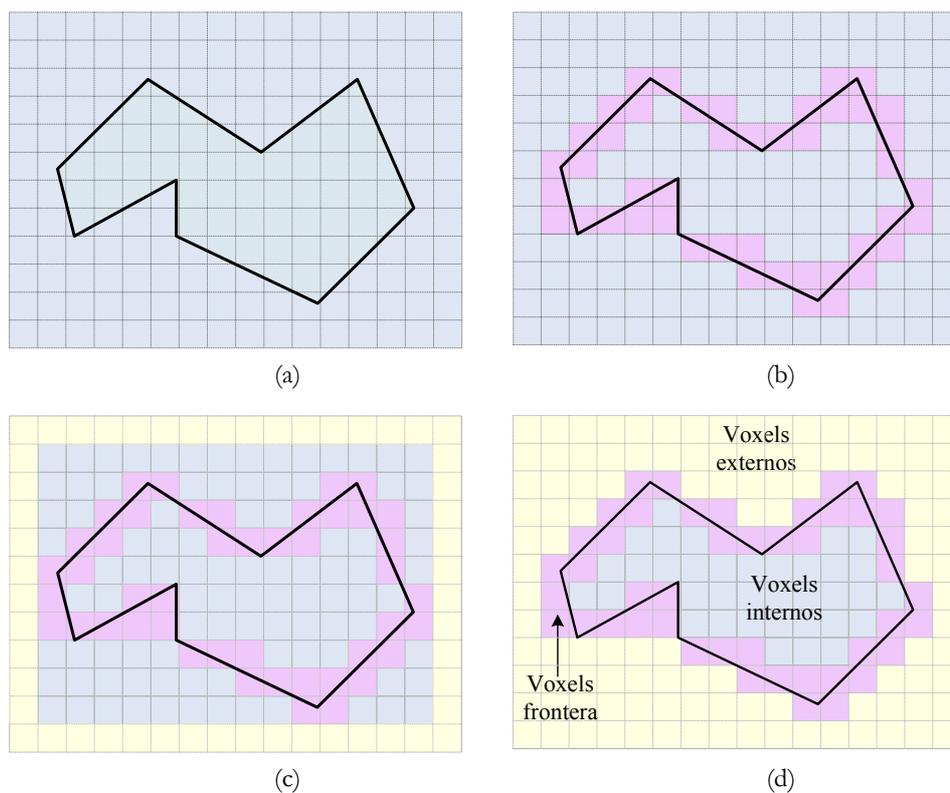


Figura 3.5: Clasificación de voxels: en azul los voxel internos, en amarillo los externos y finalmente, en rosa los voxels frontera. Inicialmente, todos los voxels se clasifican como internos (a), aquellos voxels que contengan triángulos se clasifican como frontera (b). Se inicia el proceso para encontrar los voxels exteriores (c) y finalmente, la clasificación obtenida (d).

La Figura 3.6 muestra un ejemplo 3D de la clasificación obtenida para un modelo real.

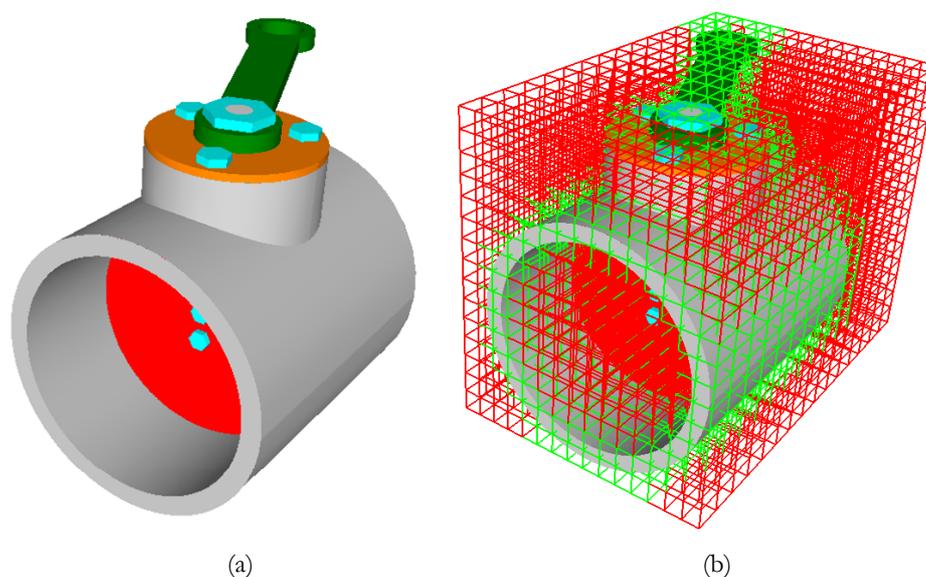


Figura 3.6: Ejemplo 3D de la clasificación del voxelizado: modelo (a) y voxelizado generado (b). En rojo se representan los voxels clasificados como externos y en verde los clasificados como frontera. Las celdas internas no se ven por encontrarse en el interior del modelo.

La clasificación de los voxels realizada facilita la detección de los grupos de triángulos de la Tabla 3.1 considerando internos aquellos triángulos del objeto móvil que se encuentren dentro de las celdas del objeto estático clasificadas como internas. Además, el coste computacional añadido por esta clasificación no penaliza el tiempo de respuesta del método ya que es una tarea que se lleva a cabo una única vez y en preproceso.

Por lo tanto, gracias a esta clasificación el módulo de colisiones es capaz de detectar todos los triángulos que permiten definir con mayor precisión el volumen de intersección necesario para el cálculo de la penetración.

La Figura 3.7 muestra gráficamente la influencia de la detección de los triángulos internos en el valor final de la penetración. Esta figura representa una posición en la que dos objetos se encuentran en colisión. Si no se tienen en cuenta los triángulos internos se obtendría una penetración menor a la real (*dIncorrecta*) ya que este valor se calcularía únicamente con los triángulos móviles que se encuentran en intersección con los triángulos del objeto estático. Como se puede observar, los triángulos internos detectados gracias

a la clasificación de las celdas, permiten calcular un valor más preciso de la magnitud de la penetración producida en cada instante de tiempo ($d_{Correcta}$).

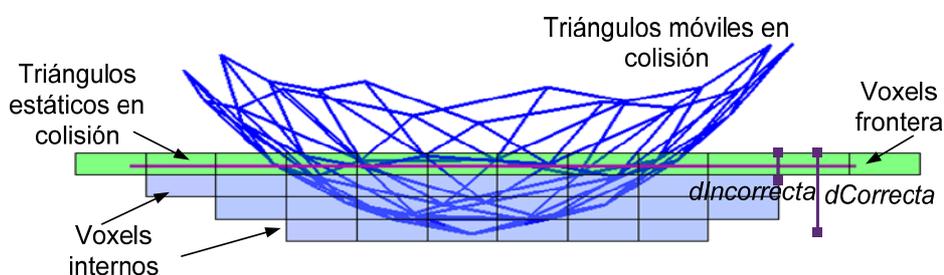


Figura 3.7: Ejemplo 3D de la penetración entre dos objetos. En color azul se han representado los triángulos en colisión del objeto móvil, en verde los voxels frontera y en azul los voxels internos.

La influencia de estos triángulos en el cálculo de la penetración a lo largo del tiempo también se ha analizado experimentalmente. La Figura 3.8 muestra una comparación entre los valores de penetración obtenidos utilizando únicamente los triángulos en colisión detectados por el módulo original de colisiones y por otro lado, utilizando además los triángulos internos. Como se puede observar en la gráfica, la detección de estos triángulos internos permite que la penetración calculada sea más suave evitando cambios bruscos entre *frames* consecutivos, lo que proporciona al usuario una mejor sensación en la respuesta háptica.

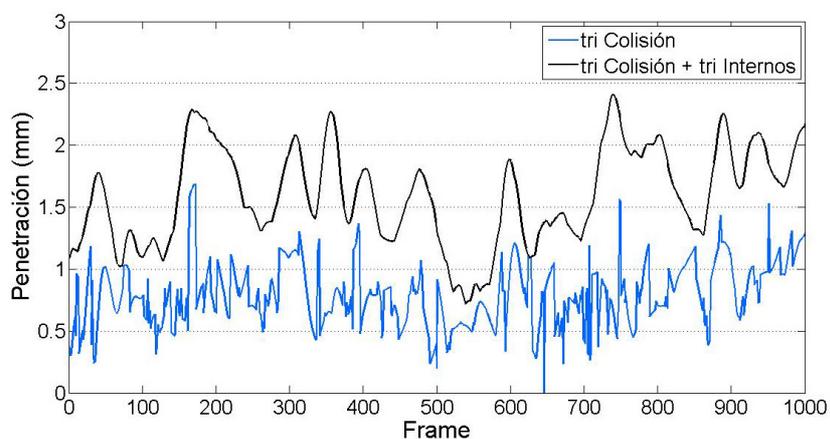


Figura 3.8: Comparación de valores de penetración obtenidos sin y con la detección de los triángulos internos móviles.

En la mayoría de las gráficas mostradas en este trabajo, en el eje X se representan los *frames* del módulo de colisiones (no confundir con el *frame rate* del visual). La descripción del funcionamiento y de la frecuencia de cada módulo se explica en el apartado 6.1.

3.3.2 DETECCIÓN DE TRIÁNGULOS FRONTERA MÓVILES

En el apartado anterior, se ha mostrado la influencia de los triángulos internos en el cálculo de la penetración. Sin embargo, existen casos en los que la detección de estos triángulos no es suficiente para garantizar la continuidad de la penetración calculada a lo largo de una simulación con un dispositivo háptico.

Según la Tabla 3.1, a los triángulos del objeto móvil que se encuentran dentro de los voxels frontera pero que no intersectan con los triángulos de la maqueta virtual, se les denominará *triángulos frontera móviles* (Figura 3.9). Estos triángulos no se consideran triángulos en colisión por no estar en intersección con triángulos del objeto estático y tampoco se consideran triángulos interiores por estar contenidos en voxels frontera y no en voxels internos.

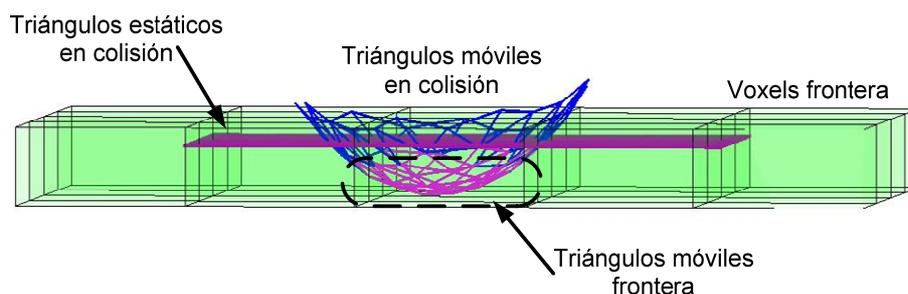


Figura 3.9: Ejemplo en el que se muestran los triángulos denominados frontera.

El hecho de no detectar estos triángulos frontera puede influir también en el cálculo de la penetración, ya que puede llevar a variaciones bruscas de su magnitud tal y como se muestra en las siguientes figuras.

La Figura 3.10 muestra dos instantes de tiempo consecutivos en los que la herramienta virtual penetra en la maqueta estática. En el instante t , el objeto móvil intersecta únicamente con voxels tipo frontera. En este caso, no todos los triángulos contenidos en estas celdas frontera son detectados, de modo que los triángulos utilizados para el cálculo de la penetración son sólo aquellos en estricta intersección con los triángulos del objeto estático.

El valor de la penetración calculada, como se puede comprobar en la figura, sería aproximadamente de 2 mm.

En el instante $t+1$, la herramienta penetra a nivel de celdas interiores por lo que los triángulos contenidos en estos voxels se detectarían como triángulos internos móviles. Estos triángulos se utilizarían en el cálculo de la penetración obteniendo un valor de aproximadamente 5 mm. Esta variación de penetración entre *frames* consecutivos resulta excesiva para garantizar al usuario una respuesta háptica continua y suave. Al utilizar los triángulos frontera para el cálculo de la penetración, esta variación brusca se reduce a 1 mm garantizando una mayor continuidad de la respuesta.

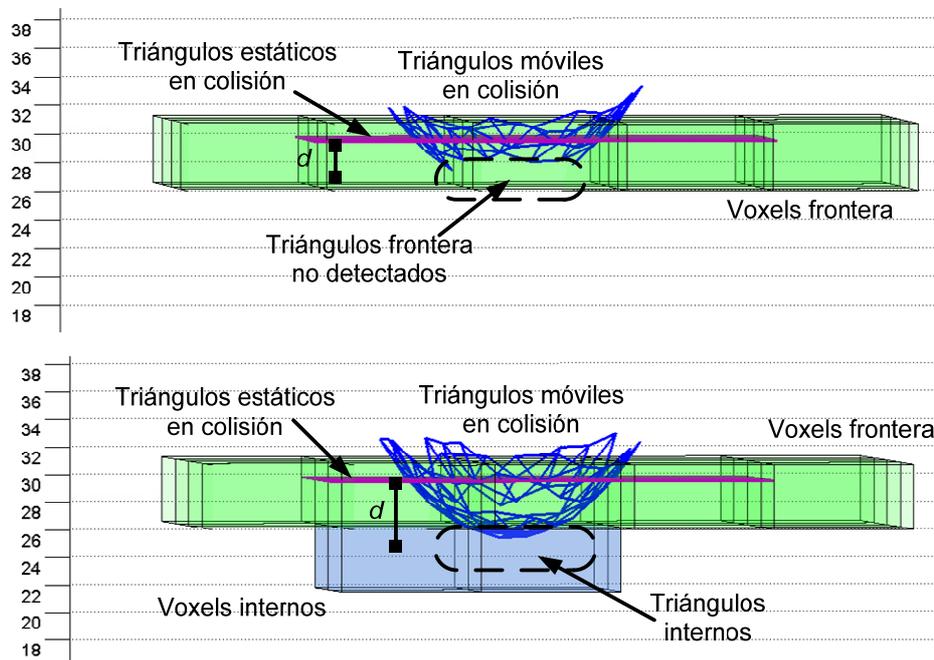


Figura 3.10: Colisión en el instante t (a) y en el instante $t+1$ (b). En el eje Y se representa el nivel de penetración en mm.

La detección de los triángulos internos está condicionada por el tamaño de voxelizado utilizado en la partición espacial del modelo estático. Si el tamaño de celda escogido es similar al tamaño de los triángulos del objeto móvil en colisión, el cálculo de los triángulos internos se realiza correctamente. Pero no sucede lo mismo cuando el tamaño de celda es considerablemente superior al tamaño de los triángulos del objeto móvil.

Para demostrar la influencia del tamaño de la voxelización en la detección de los triángulos frontera, y como consecuencia, en el cálculo de la penetración, se ha realizado un experimento analizando tres tamaños diferentes de celda a lo largo de un path predefinido que se utiliza para los tres casos. Para cada uno de estos tres tamaños, se ha calculado la penetración obtenida sin detectar los triángulos frontera móviles pero considerando los triángulos internos móviles.

La Figura 3.11 muestra que a medida que el tamaño de celda disminuye, las penetraciones obtenidas son más continuas aún sin utilizar los triángulos frontera. Esto se debe principalmente a que los triángulos frontera dejan de tener influencia a medida que el tamaño de la celda disminuye, ya que estos triángulos no detectados inicialmente, serán detectados como triángulos interiores con celdas de menor tamaño. Los tamaños de celda utilizados en el experimento son: 18,8, 7,5 y 4,2 mm respectivamente utilizando voxels cúbicos en la partición espacial.

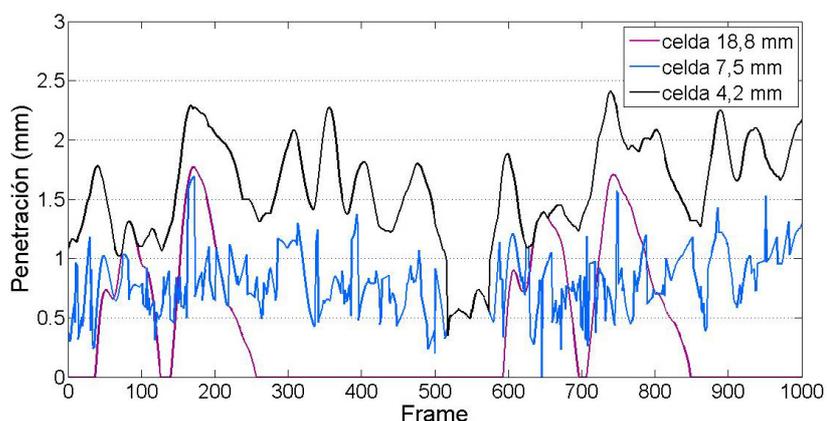


Figura 3.11: Influencia del tamaño del voxelizado en el cálculo de la penetración.

3.3.3 CÁLCULO DE LOS TRIÁNGULOS FRONTERA ESTÁTICOS

Cuando se detecta una colisión en el entorno, la subdivisión espacial del objeto estático permite reducir la búsqueda de los voxels en colisión a un conjunto reducido de celdas que contienen los triángulos del objeto estático que definen esa área de contacto (V_0).

A partir de este V_0 se detectan los triángulos en colisión del objeto estático. Sin embargo, dependiendo del nivel de penetración de la herramienta en el entorno, existen triángulos del objeto estático que aún

estando dentro de los V_0 no intersectan con los triángulos del objeto móvil. Generalmente, estos triángulos son los que se encuentran en la parte central del área en colisión delimitada por los triángulos estáticos que sí han sido identificados como triángulos en colisión. En la Tabla 3.1, a estos triángulos se les ha denominado *triángulos estáticos frontera*.

La Figura 3.12 muestra un ejemplo de la zona de la superficie en intersección en la que no se han detectado los triángulos centrales del objeto estático.

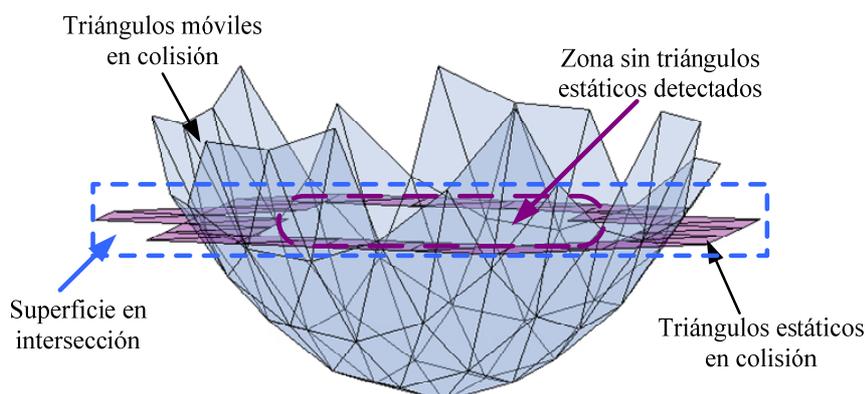


Figura 3.12: Ejemplo de dos objetos en colisión. Los triángulos del objeto estático de la zona central del área de intersección no son detectados.

De nuevo, este problema viene derivado por la representación frontera utilizada en los modelos. Los triángulos frontera estáticos no son detectados por estar ambos modelos representados únicamente por su contorno. En esta zona central, se produce una penetración tal que los triángulos que definen el contorno de la herramienta no intersectan con los triángulos del objeto estático que definen esa zona central del área de contacto.

Como se puede observar en la Figura 3.13, la detección de estos triángulos frontera estáticos tiene gran influencia en el cálculo de una penetración adecuada para la restitución de esfuerzos, ya que estos triángulos junto con los triángulos internos móviles detectados, son los que delimitan el volumen de intersección. Sin estos triángulos, los métodos de respuesta de colisión descritos en el capítulo 5 detectarían una penetración menor a la que realmente se ha producido en el momento de la colisión (*dIncorrecta*). Por el contrario, la penetración obtenida considerando los

triángulos estáticos frontera se acerca más a la penetración real de la colisión ($d_{Correcta}$).

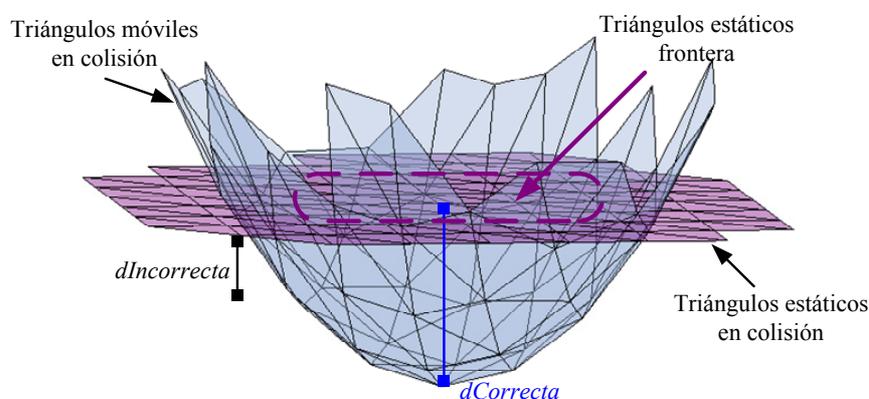


Figura 3.13: Ejemplo de la influencia de los triángulos frontera estáticos en el cálculo de la penetración.

La Figura 3.14 muestra una comparación de los resultados experimentales obtenidos con y sin la detección de los triángulos frontera estáticos. En ambos experimentos el path utilizado ha sido el mismo. De los resultados se concluye que la detección de estos triángulos ayuda a conseguir un valor de penetración más suave.

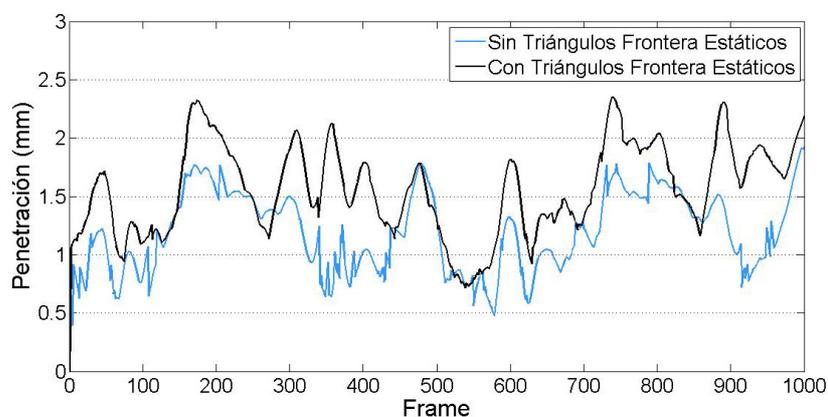


Figura 3.14: Comparación de valores obtenidos en el cálculo de la penetración con y sin la detección de los triángulos frontera estáticos.

La detección de estos triángulos estáticos frontera no es una tarea trivial por lo que se han analizado diferentes soluciones para su

identificación. La primera de ellas consiste en considerar como triángulos en colisión del objeto estático todos aquellos que se encuentren dentro de los voxels objetivo que definen el área en colisión aunque no intersecten con los triángulos del objeto móvil. Esto implica que no sería necesario realizar la *narrow phase* del módulo de colisiones con el consiguiente ahorro del coste computacional asociado a la fase de detección de colisiones.

La segunda opción planteada utiliza dos listas de triángulos en el proceso de detección de colisiones. En una de las listas se almacenan todos los triángulos estáticos que intersectan con los triángulos móviles (*ITriEst*). En la segunda lista, se guardan todos los triángulos del objeto estático contenidos en los voxels objetivo que engloban el área en colisión pero que no han sido detectados como triángulos en colisión (*triángulos estáticos frontera*).

Utilizando las coordenadas mínima y máxima de los triángulos contenidos en *ITriEst*, se delimitan las dimensiones de la superficie en contacto. En base a estas medidas se define una caja contenedora (AABB) y se añaden a la lista de triángulos en colisión del objeto estático aquellos triángulos de la lista adicional que se encuentren dentro de esta caja. Esta segunda opción no optimiza el módulo de colisiones pero en cambio limita el número de triángulos estáticos frontera que se añaden a la lista final de triángulos estáticos en colisión.

Para escoger la opción que más se ajuste a las restricciones de tiempo propias de las aplicaciones de renderizado háptico, se analiza tanto el coste computacional de la detección de las colisiones como el tiempo necesario para calcular la respuesta a la colisión. El número de triángulos en colisión utilizados en cada caso condicionará el tiempo de respuesta por lo que también se muestran gráficamente.

La Figura 3.15 muestra que a pesar de que en la primera opción no se realiza la *narrow phase*, el ahorro computacional es mínimo con respecto a la segunda opción, a pesar del incremento de triángulos estáticos en colisión detectados para cada caso (Figura 3.16).

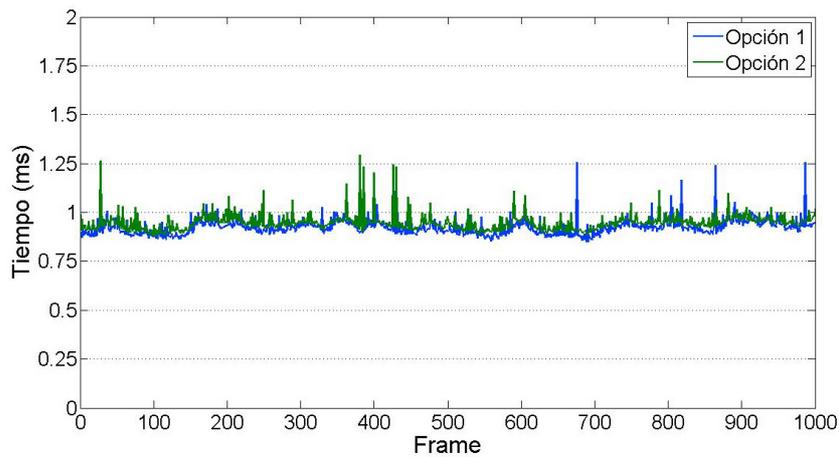


Figura 3.15: Comparación del tiempo de detección de colisión para cada una de las opciones.

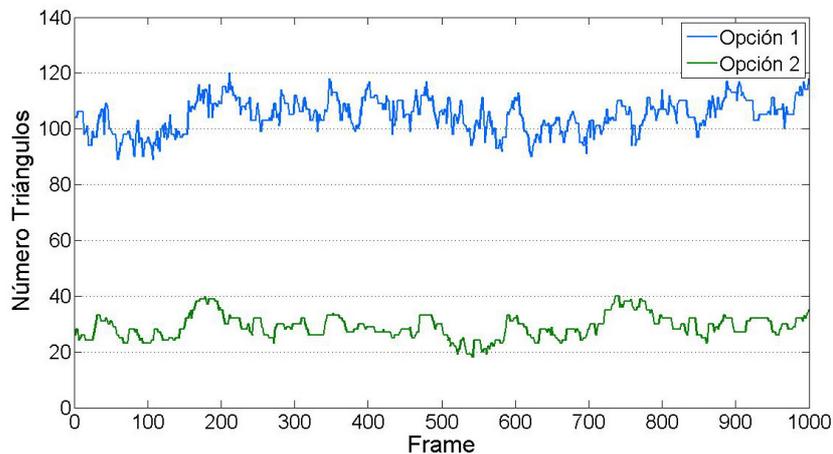
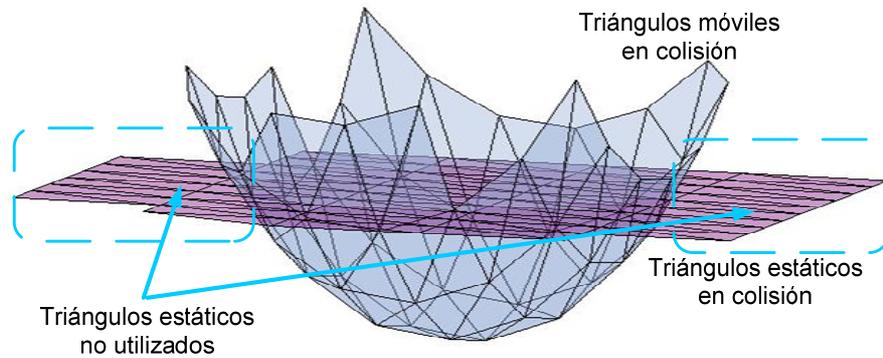


Figura 3.16: Número de triángulos estáticos en contacto para cada una de las opciones

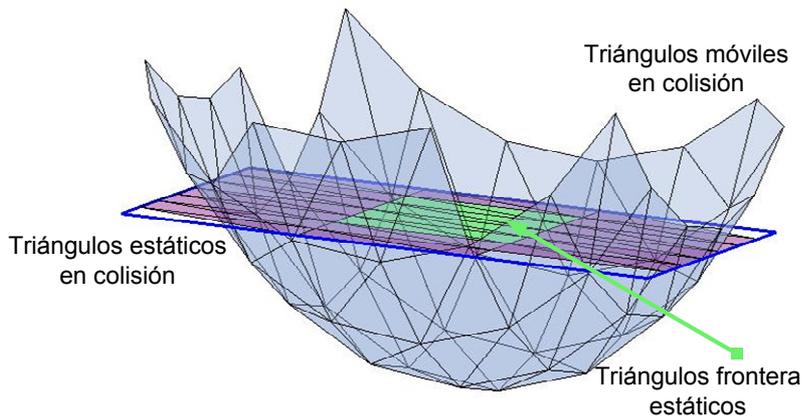
En la Figura 3.17 se aprecia la diferencia significativa entre los triángulos estáticos detectados para cada una de las opciones. En el caso de la primera opción, se consideran triángulos en contacto todos los triángulos que se encuentran dentro de los V_ω , lo que implica que se tienen en consideración un mayor número de triángulos. Pero tal y como se observa en la Figura 3.17a, muchos de estos triángulos no son útiles para definir el volumen de intersección.

En cambio la segunda opción (Figura 3.17b) incluye únicamente aquellos triángulos que se encuentran dentro de un volumen definido por

las coordenadas mínima y máxima de los triángulos en colisión del objeto estático detectados por el módulo de colisiones. Esta superficie está representada mediante el cuadro azul. Como se puede comprobar, el número de triángulos que se tienen en cuenta disminuye considerablemente.



(a)



(b)

Figura 3.17: Triángulos estáticos detectados con la opción 1 (a) y con la opción 2 (b).

3.4 CONSIDERACIONES FINALES

El cálculo de la respuesta desarrollado en esta Tesis sigue la filosofía de los métodos basados en penalización, por lo que es necesario calcular un valor de penetración adecuado que permita separar dos objetos en colisión hasta

que dejen de intersectarse. Esta distancia se calcula en base a los triángulos detectados por el módulo de detección de colisiones.

En este capítulo se han mostrado las modificaciones realizadas al método original de detección de colisiones para obtener la información necesaria para los algoritmos de respuesta expuestos en el capítulo 4 y capítulo 5.

Como se ha explicado a lo largo de este capítulo, la detección de los triángulos en intersección de cada uno de los objetos en colisión no es suficiente para garantizar que el usuario reciba una respuesta háptica continua a lo largo de una simulación. Por lo tanto, además de los triángulos en intersección, es necesario considerar triángulos adicionales de cada uno de los objetos. Para el caso del objeto estático, han de tenerse en cuenta los triángulos frontera estáticos y para el caso del objeto móvil, es necesario añadir los triángulos frontera móviles y los triángulos internos móviles. A este conjunto de triángulos (pertenecientes a cada objeto) utilizado en el cálculo de la respuesta, en adelante se les denominará *triángulos en contacto*.

La definición formal de este concepto se puede expresar de la siguiente forma. Dado un triángulo t , un conjunto de triángulos TRI y voxels de tipo frontera ($V_{frontera}$) e interiores ($V_{interior}$) obtenidos a partir de la partición espacial realizada, se definen las siguientes funciones booleanas:

colisión (t , TRI) que devuelve true si y sólo si $\exists t' \in TRI \mid t$ y t' intersectan,
frontera (t , $V_{frontera}$) que devuelve true si $t \subset V_{frontera}$ y finalmente la función
interno (t , $V_{interior}$) que devuelve true si $t \subset V_{internos}$.

Si S y M son los conjuntos de triángulos que definen los objetos estático y móvil respectivamente, el módulo de colisiones obtiene como salida dos conjuntos de triángulos S_c y M_c que se definen como:

$$\begin{aligned} S_c &= \{t \in S, \text{colisión}(t, M) = \text{true} \cup \text{frontera}(t, S) = \text{true}\} \\ M_c &= \{t \in M, \text{colisión}(t, S) = \text{true} \cup \text{internos}(t, S) = \text{true} \cup \text{frontera}(t, S) = \text{true}\} \end{aligned} \quad (3.1)$$

En los sucesivos capítulos, se utilizarán estos conjuntos, S_c y M_c para representar el tipo de información que se envía al módulo de respuesta de colisión.

CAPÍTULO 4

PROBLEMA DE LA RESPUESTA DE COLISIÓN

La incorporación de los dispositivos hápticos a los entornos virtuales permite aumentar el realismo de estos entornos además de proporcionar un mayor grado de interactividad. Gracias a estos dispositivos, el usuario puede obtener una sensación física ante la colisión entre dos objetos además de sentir pesos y rugosidades diferentes para cada objeto simulando el mundo real.

En el caso concreto de las tareas de accesibilidad, el usuario podría determinar el camino adecuado para acceder a un determinado elemento con el único estímulo de la vista pero el realismo de la aplicación aumenta considerablemente si además es capaz de sentir una fuerza y/o par que le impida penetrar en los objetos virtuales.

Para calcular una respuesta háptica a la colisión, se utiliza la información calculada por el módulo de colisiones explicado en el capítulo anterior. Este módulo es el encargado de comprobar constantemente si los diferentes objetos que componen la escena colisionan entre sí y de calcular las listas de triángulos que están implicadas en dicha colisión. A partir de esta información, el módulo de respuesta de colisión será el encargado de evitar que los objetos interpenetren calculando de forma explícita una serie de fuerzas y/o pares que se restituyen al usuario.

En este capítulo, se analiza el problema del cálculo de la respuesta háptica de forma general describiendo los diferentes parámetros a determinar y la solución aportada utilizando la información geométrica

calculada por el módulo de colisiones. La solución desarrollada se basa en estrategias de penalización descritas en el capítulo 2.

Además se analiza la importancia que tiene en la percepción del usuario factores como la dirección o el punto de contacto utilizados para calcular la respuesta háptica final.

4.1 RESPUESTA DE COLISIÓN

El cálculo final de las fuerzas y momentos a restituir es una tarea que se lleva a cabo conjuntamente entre el módulo de respuesta de colisión y el módulo de control a partir de los datos obtenidos por el módulo de detección de colisiones.

El módulo de detección de colisiones detecta todos los triángulos necesarios para un cálculo correcto de la respuesta a dicha colisión, concretamente, se obtienen dos listas de triángulos en contacto tal y como se ha explicado en el capítulo 3: S_c contiene los triángulos del objeto estático implicados en la colisión y M_c los del objeto móvil.

A partir de esta información geométrica, el módulo de respuesta de colisión es el encargado de calcular la fuerza y/o momento ideales (\mathbf{F}_{ideal} y \mathbf{T}_{ideal}) que el usuario debería sentir. Finalmente, esta información es enviada al módulo de control que modificará esta información para ajustarla a las características propias del dispositivo háptico utilizado (\mathbf{F} y \mathbf{T}). La frecuencia del módulo de control (1 kHz) es superior a la del módulo de colisiones, por lo que el lazo de control debe además utilizar técnicas de interpolación y predicción que le permitan seguir calculando una respuesta háptica estable ante la falta de información del lazo de colisiones. La Figura 4.1 muestra un esquema del flujo de información entre estos tres módulos.

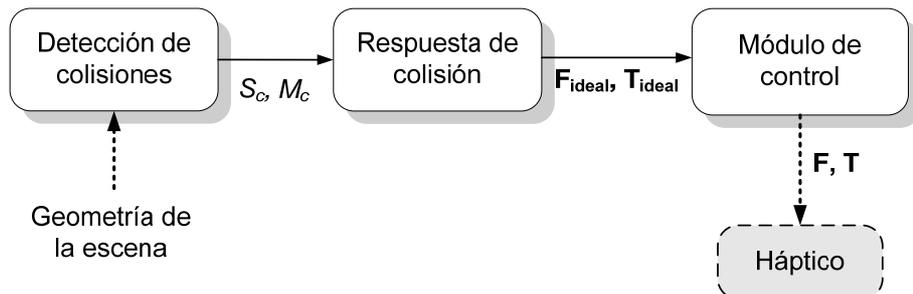


Figura 4.1: Flujo de información entre los módulos de detección, respuesta de colisiones y control.

Las expresiones más sencillas para expresar la fuerza a restituir son el modelo elástico de un muelle simple definido por la ley de Hooke (Ecuación (4.1)) o el modelo viscoelástico definido en la Ecuación (4.2) en el que se introduce amortiguamiento.

$$\mathbf{F} = k d \mathbf{n} \quad (4.1)$$

$$\mathbf{F} = k d \mathbf{n} + b \mathbf{v} \quad (4.2)$$

donde k es la constante de rigidez del sistema, d es la penetración, \mathbf{n} es la dirección de la fuerza o normal de contacto, b es la componente viscosa y \mathbf{v} la velocidad de aumento de la penetración.

En caso de restituir un momento (\mathbf{T}), la expresión viene dada por la Ecuación (4.3) donde \mathbf{r} es el vector formado por el punto donde se aplica el par normalmente el centro de masas del objeto (\mathbf{c}_g) y el punto de contacto en el que se calcula la colisión (\mathbf{p}_c).

$$\mathbf{T} = \mathbf{r} \times \mathbf{F} \quad (4.3)$$

En los siguientes apartados se analizan dos de los parámetros necesarios para determinar una respuesta háptica: la dirección y el punto de contacto. El problema del cálculo de la penetración se analizará en el siguiente capítulo. Es necesario recalcar que las pautas seguidas en este trabajo tienen como prioridad proporcionar al usuario una respuesta agradable y continua anteponiendo estos objetivos al coste computacional.

4.2 CÁLCULO DEL NÚMERO DE CONTACTOS IMPLICADOS EN LA COLISIÓN

Cuando se produce una colisión con el entorno, la herramienta virtual puede colisionar simultáneamente con diferentes zonas del objeto estático, hecho que ocurre con mayor probabilidad cuando se trabaja con escenarios complejos. Cada una de estas zonas de intersección está definida por los triángulos en contacto de cada uno de los objetos detectados por el módulo de colisiones. A estas zonas se les denominará *áreas de contacto*.

En caso de detectar más de un área de contacto, cada contacto se analiza de forma independiente obteniendo unos resultados parciales para cada área. Estos resultados parciales serán después utilizados conjuntamente para calcular una respuesta global a la colisión (Ecuaciones (4.4) y (4.5)).

$$\mathbf{F}_{\text{final}} = \sum_{i=1}^m \mathbf{F}_i = \sum_{i=1}^m \mathbf{n}_i d_i \quad (4.4)$$

$$\mathbf{T}_{\text{final}} = \sum_{i=1}^m \mathbf{r}_i \times \mathbf{F}_i \quad (4.5)$$

La obtención de una respuesta coherente a la colisión producida con el entorno depende en gran medida de determinar la información que define a cada una de las áreas. Por lo tanto, la primera fase del método es la encargada de determinar el número de áreas de contacto que definen la colisión. Esta tarea se realiza en cada *frame* por lo que su coste debe ser reducido para no influir negativamente en el tiempo final de cálculo.

4.2.1 CÁLCULO DE LAS ÁREAS DE CONTACTO

La separación de la información de colisión en diferentes contactos plantea un problema ya que las listas de triángulos en contacto de cada uno de los objetos (S_c y M_c) que recibe el módulo de respuesta de colisión no se encuentran ordenadas bajo ningún criterio. S_c y M_c son dos listas globales que contienen todos los triángulos en contacto sin ningún indicador del área a la que pertenecen.

Como punto de partida para la identificación de las áreas de contacto existentes en la colisión, se toman los triángulos contenidos en S_c . La ventaja de escoger estos triángulos es que, como su propio nombre indica, este objeto no se mueve por lo que no es necesario recalcular las nuevas coordenadas después de un desplazamiento y como consecuencia es más sencillo plantear optimizaciones en preproceso.

La asignación de estos triángulos a su correspondiente área de contacto se realiza analizando las adyacencias geométricas. De esta forma, se considera que los triángulos en contacto estáticos forman un área de contacto cuando son adyacentes o vecinos¹. El proceso por el cual se identifican las adyacencias de los triángulos tiene un coste computacional elevado si se realiza en tiempo de ejecución. Este coste se debe a que por cada triángulo en contacto habría que testear las adyacencias con el resto de

¹ Se consideran triángulos vecinos a aquellos que comparten una arista. De esta forma, un triángulo tiene como máximo tres triángulos vecinos.

triángulos de la lista, obteniendo un algoritmo de orden $O(n^2)$ siendo n el número de triángulos en contacto del objeto estático.

Con el objetivo de reducir el número de cálculos en tiempo de ejecución, el cálculo de las adyacencias se realiza en preproceso. En esta etapa se almacena para cada uno de los triángulos que definen el modelo estático sus triángulos vecinos. Posteriormente y ya en tiempo de ejecución, el módulo de colisiones detecta los triángulos del objeto estático que delimitan el área o áreas donde se ha producido la colisión. Para poder identificar el número de áreas en colisión y asignar cada triángulo a su área correspondiente, se tienen en cuenta las adyacencias determinadas en la etapa de preproceso. En la definición de vecindad utilizada, se ha establecido que un triángulo tiene como máximo tres triángulos vecinos, por lo que el coste computacional del cálculo de las áreas es lineal $O(3n)$ en vez de cuadrático.

Se ha realizado un análisis tanto del coste computacional añadido como del incremento de memoria necesario para el cálculo de las adyacencias geométricas utilizando modelos definidos mediante un número diferente de polígonos. Los resultados (Figura 4.2 y Figura 4.3) muestran que el incremento de memoria es mínimo y aunque el coste computacional se ve incrementado, este proceso se lleva a cabo en la etapa de preproceso, por lo que no repercute en el rendimiento final del método.

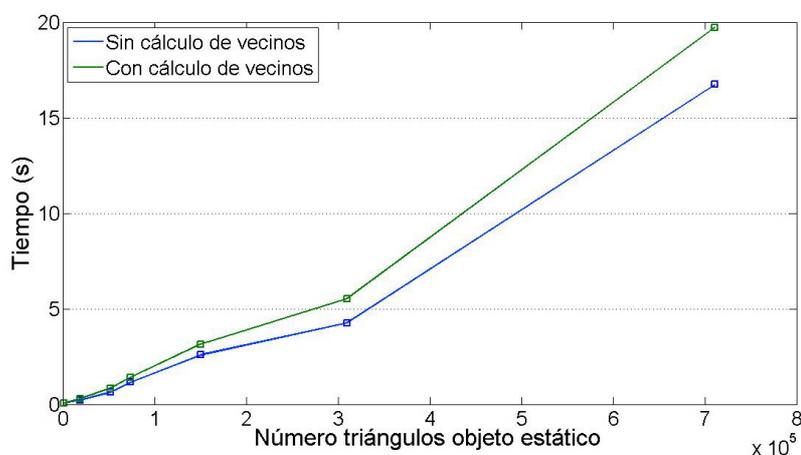


Figura 4.2: Comparativa del coste computacional del cálculo de triángulos vecinos

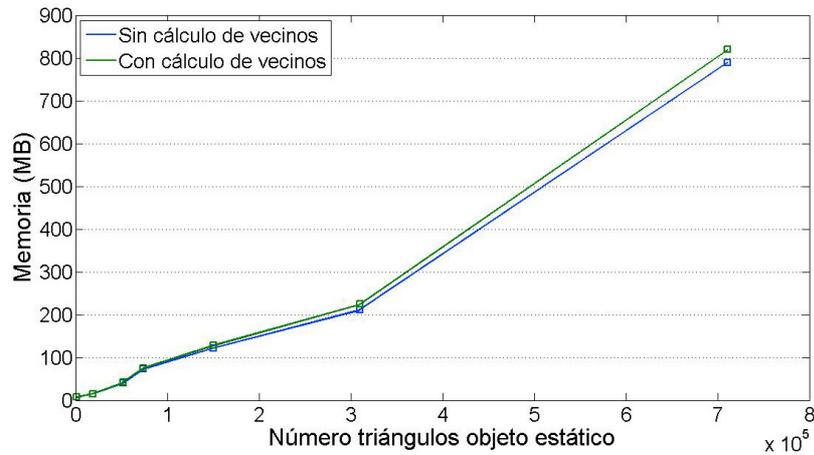


Figura 4.3 Comparativa de la memoria consumida en el cálculo de triángulos vecinos

Una vez que los triángulos estáticos en contacto se separan en las diferentes áreas de contacto, es necesario hacer lo mismo con los triángulos en contacto de la herramienta virtual. Para llevar a cabo esta tarea, es necesario determinar de forma aproximada las dimensiones de cada área identificada para limitar la búsqueda de los triángulos móviles.

La mínima y máxima coordenada de los triángulos estáticos asociados a cada área de contacto delimitan una caja contenedora que la engloba. La longitud de esta caja (l) se utiliza para definir una esfera de radio $l/2$ que se posiciona en el punto medio de esta caja contenedora. Esta esfera permite separar de forma rápida y eficiente los triángulos del objeto móvil asociados a cada área de contacto considerando que cada triángulo del objeto móvil se asignará al área cuya esfera asociada lo contenga.

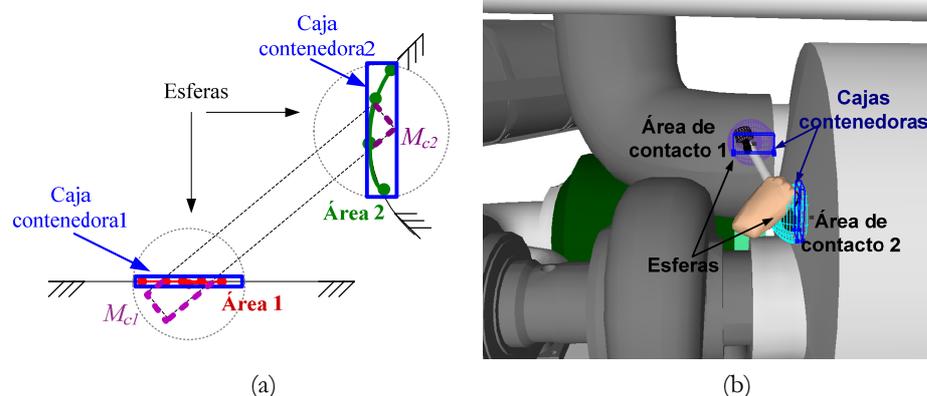


Figura 4.4: Separación por áreas de los triángulos en contacto del objeto estático y móvil. Representación 2D (a) y 3D (b).

4.2.2 CÁLCULO DE LAS ZONAS DE CONTACTO

La dificultad para calcular una respuesta háptica continua y suave a una colisión se ve incrementada en zonas de la superficie en las que aparecen discontinuidades de tipo C^1 . Este tipo de discontinuidades pueden tratarse como casos especiales pero para ello es necesario detectar cuándo se producen, lo que en principio no resulta trivial a partir de la información de la que se dispone (\mathcal{S}).

La división o separación de los contactos en áreas proporciona la información del número de superficies diferentes que se encuentran en colisión. Sin embargo, esta subdivisión no proporciona información suficiente acerca de la complejidad de la superficie en contacto por lo que se desconoce si se trata de una esquina, de una superficie plana o curva, si la superficie presenta diferentes discontinuidades, etc.

Conocer las características de las diferentes áreas de contacto proporciona información adicional que permite calcular en cada caso y de una forma más adecuada la respuesta a dicha colisión. Por lo tanto, dentro de cada área de contacto se lleva a cabo un análisis de la geometría asociada definiendo diferentes zonas por cada área a las que se denominará *zonas de contacto*. Estas nuevas zonas se crean en función del ángulo que forman entre sí los vectores normales de los triángulos asociados al área a subdividir. Si este ángulo supera un valor umbral predeterminado, entonces se creará una nueva zona de contacto formada por los triángulos que cumplan esta condición. El ángulo umbral es un parámetro cuyo valor puede ser

modificado, de forma que puede escogerse el grado de precisión en el que se dividen las diferentes zonas de contacto. En principio, el número de zonas en las que se subdivide un área de contacto dependerá del número de discontinuidades de tipo C^1 existentes en esa área de contacto.

En la Figura 4.5 se muestra un ejemplo de la división en zonas de un área de contacto. Las líneas azules representan las facetas que se encuentran en contacto con sus respectivas normales. Tras la primera fase del método, estos dos triángulos pertenecen a la misma área de contacto al tratarse de triángulos adyacentes y el área de contacto en cuestión estaría formada por una única zona de contacto.

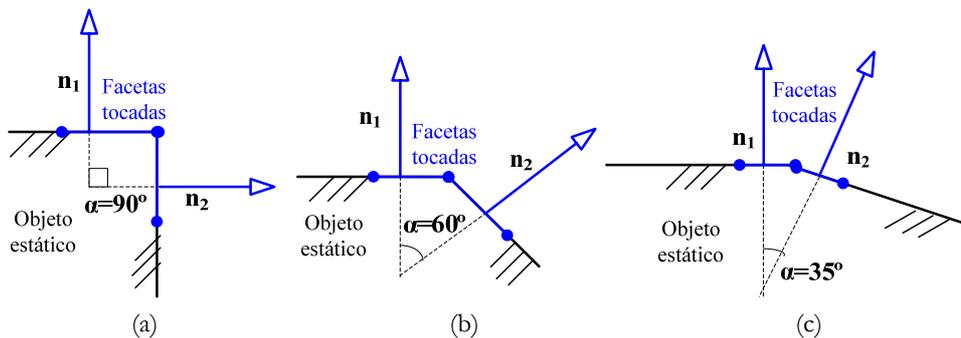


Figura 4.5: Ejemplo 2D de la división en varias zonas de contacto.

Para la subdivisión de cada área de contacto, se establece por ejemplo un valor de 40° para el ángulo umbral. En la Figura 4.5a y Figura 4.5b, el ángulo que forman las normales de los triángulos (α) es mayor que el ángulo umbral fijado de forma que se pasa de tener una zona de contacto compuesta por dos triángulos a tener dos zonas de contacto con un triángulo cada una. En el último caso (Figura 4.5c), el valor del ángulo no supera el valor preestablecido, por lo que se mantendrá una única zona de contacto.

En la Figura 4.6, se muestra con un ejemplo 2D y 3D la subdivisión de las áreas y zonas de contacto que se lleva a cabo. Inicialmente, se detectan dos áreas de contacto en función de la adyacencia de los triángulos, y cada una de estas áreas está formada por una única zona de contacto. Posteriormente, se realiza la subdivisión de cada una de las áreas de contacto identificadas. Para la primera de estas áreas (Área de contacto 1) sólo existe una zona de contacto ya que las normales de los triángulos asociadas a ella son iguales. La segunda área detectada (Área de contacto 2), presenta una discontinuidad geométrica por lo que se lleva a cabo una

subdivisión obteniendo dos zonas de contacto representadas en color rojo y verde.

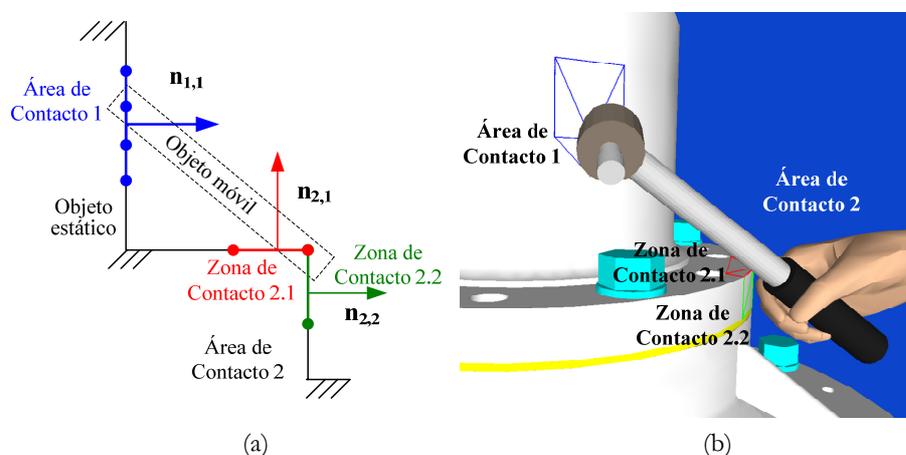


Figura 4.6: Ejemplo 2D (a) y 3D (b) de la subdivisión en áreas y zonas de contacto.

Una vez detectadas las diferentes áreas y haber subdividido cada una de ellas en diferentes zonas de contacto, se calcula una fuerza de restitución para cada una de estas áreas. Esta fuerza viene dada por la Ecuación (4.1) o (4.2) según el modelo de contacto utilizado, para lo que es necesario calcular una normal y una penetración resultante para cada una de las áreas. Si la respuesta final deseada es un par, la complejidad del cálculo aumenta ya que además de los dos parámetros que definen la fuerza (\mathbf{n} y d) es necesario calcular un punto de contacto (\mathbf{p}_c) en el cual se aplicará la fuerza resultante.

En los siguientes apartados se analizarán los pasos que se siguen para calcular dos de los tres parámetros necesarios para calcular las fuerzas y pares finales: normal y punto de contacto.

4.3 CÁLCULO DE LAS NORMALES DE CONTACTO PARA CADA ÁREA

La dirección tanto de las fuerzas parciales como de la respuesta final juega un papel importante para garantizar una respuesta estable. Los cambios bruscos en las normales de las superficies como ocurre en las esquinas o aristas pueden provocar discontinuidades en la fuerza resultante. Para evitar estos cambios bruscos de normales sería necesario utilizar un número elevado de polígonos redondeando dichas superficies. Otra solución es aplicar estrategias de interpolación de normales similares a las utilizadas para

generar la iluminación de las escenas (Gouraud 1971, Phong 1975). A estas técnicas de sombreado que suavizan las normales aplicadas al cálculo de una respuesta háptica se les denomina *force shading*.

Concretamente, Morgenbesser et al. (1996) fueron los primeros en demostrar la sensibilidad de los usuarios ante cambios muy bruscos de la dirección de la fuerza restituida en el sistema. Utilizando una técnica similar a la de Phong, Salisbury et al. (1995a) demostraron que la percepción de suavidad de un objeto no depende tanto del mallado que lo define sino de las normales que se definan para ese modelo. Ruspini et al. (2001) también utilizan las técnicas de *force shading* en su trabajo.

Además de los cambios bruscos en las direcciones de las normales, las aplicaciones que utilizan la estrategia de la penalización para el cálculo de la fuerza resultante se ven influenciadas por el parámetro de la penetración. Cualquier cambio brusco en alguno de estos dos parámetros puede originar discontinuidades en la fuerza resultante que provocan un comportamiento inestable del sistema que influye de forma negativa en la percepción final del usuario. Gregory et al. (2000b) detectan este problema y lo solucionan evitando que las diferencias entre las fuerzas actuales y las de un *frame* anterior superen cierto valor pre-establecido.

En este apartado se explica la técnica utilizada para suavizar los cambios bruscos que se producen en la dirección de la normal debidas a diferentes discontinuidades geométricas. Partiendo de los triángulos asignados a cada una de las áreas de contacto, se distinguen dos posibles casos: la herramienta está tocando una única zona (Figura 4.7a) o por el contrario está colisionando con varias zonas de contacto simultáneamente (Figura 4.7b).

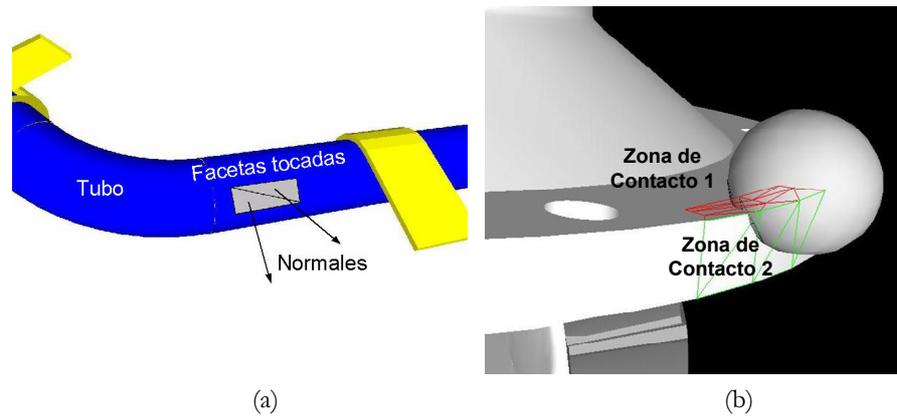


Figura 4.7: Colisión de un objeto móvil con un objeto estático en un área de contacto: el área de contacto se divide en una zona de contacto (a), el área se divide en dos zonas de contacto representadas en color rojo y verde (b).

4.3.1 CASO DE UNA ZONA DE CONTACTO

En este caso la herramienta u objeto móvil está tocando una única zona de contacto. El cálculo de la normal resultante se calcula teniendo en cuenta la información de la superficie estática, es decir, como la suma vectorial de los m vectores normales asociados a los triángulos asignados a esa zona tras lo cual se normaliza la solución, tal y como se muestra en la Ecuación (4.6).

$$\mathbf{n}_{\text{área}} = \frac{\sum_{i=1}^m \mathbf{n}_i}{\left| \sum_{i=1}^m \mathbf{n}_i \right|} \quad (4.6)$$

Puede aparecer un problema si la suma de las normales es nula. Este hecho puede ocurrir si la herramienta corta transversalmente un objeto cilíndrico. El típico ejemplo de este tipo de problema se puede observar en la Figura 4.8 que representa una colisión contra unos tubos. Estos elementos pueden ser muy largos y finos por lo que la herramienta puede llegar a atravesarlos con facilidad.

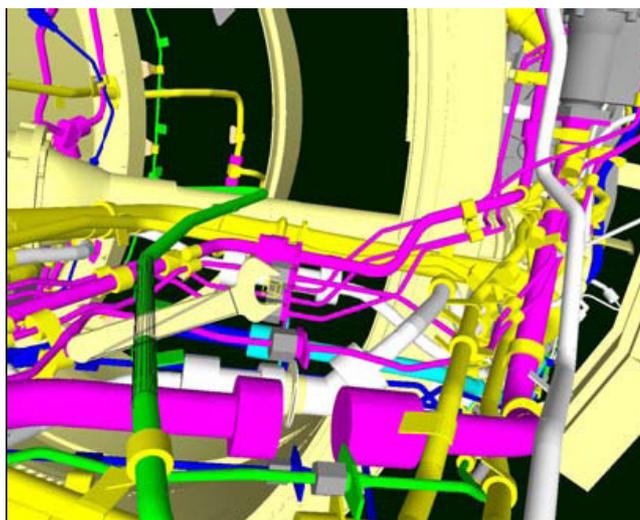


Figura 4.8: Colisión de la herramienta atravesando un tubo.

Para solucionar este problema, el módulo de respuesta de colisiones implementa una técnica de *Back-face Culling* aplicada a las colisiones (Vanecek 1994). Si una faceta estática que está en colisión tiene una normal en la dirección de la trayectoria de la herramienta, entonces esa faceta no se tiene en cuenta para el cálculo de la normal de contacto. Más formalmente, la faceta i es rechazada si $\mathbf{v} \cdot \mathbf{n}_i > 0$ donde \mathbf{v} es el vector velocidad del objeto móvil y \mathbf{n}_i es la normal de la faceta i . De esta manera también se incrementa la eficiencia del cálculo ya que se rechazan cálculos innecesarios para el resultado.

4.3.2 CASO DE VARIAS ZONAS DE CONTACTO

Si los objetos utilizados son complejos es muy probable que la herramienta colisione con áreas de la maqueta estática con discontinuidades C^1 . En este caso, una misma área de contacto se dividirá en más de una zona de contacto. Para resolver estos casos de la forma más adecuada, se han estudiado diferentes alternativas de respuesta en cuanto a realismo, percepción, continuidad y estabilidad.

Los dos parámetros que definen una fuerza son la normal de contacto y la penetración. Cualquier cambio brusco en alguno de estos dos parámetros puede provocar discontinuidades en la fuerza resultante y un comportamiento inestable del dispositivo que influirá negativamente en la percepción final del usuario.

Para resolver de la forma más adecuada posible el problema del cambio de la normal, se han analizado diferentes soluciones. La primera aproximación utilizada consiste en calcular la normal resultante del área como la suma vectorial de las normales de los triángulos estáticos asignados a cada una de las zonas en las que se ha dividido el área de contacto a tratar (véase la Ecuación (4.7)), donde $numZonas$ es el número de zonas en las que se divide el área en colisión y $numTriZona$ representa el número total de triángulos asociados a una determinada zona de contacto.

$$\mathbf{n}_{\text{área}} = \frac{\sum_{j=1}^{numZonas} \mathbf{n}_{\text{zona } j}}{\left| \sum_{j=1}^{numZonas} \mathbf{n}_{\text{zona } j} \right|}, \text{ donde } \mathbf{n}_{\text{zona } j} = \sum_{i=1}^{numTriZona} \mathbf{n}_i \quad (4.7)$$

La segunda aproximación consiste en utilizar las normales de los triángulos en contacto del objeto móvil asociados al área de contacto a tratar (Ecuación (4.8)). Al igual que en el caso anterior, se calcula una suma vectorial que posteriormente es normalizada. Al utilizar las normales de los triángulos móviles en el cálculo de la normal resultante, ésta debe invertirse para que se oponga al movimiento del objeto móvil.

$$\mathbf{n}_{\text{área}} = \frac{-\sum_{t=1}^{numTri} \mathbf{n}_t}{\left| \sum_{t=1}^{numTri} \mathbf{n}_t \right|}, \text{ donde } t \in M_{ci} \quad (4.8)$$

donde M_{ci} es un subconjunto de M_c que contiene los triángulos del objeto móvil asociados a una determinada área de contacto y $numTri$ el número de triángulos móviles que pertenecen a dicha área de contacto.

Para comprobar el comportamiento real de cada una de las aproximaciones planteadas para el caso de una discontinuidad, se ha realizado una simulación de una esfera de radio 50 mm que se desliza sobre un plano hasta llegar a una esquina convexa. Para ello se ha definido un camino ideal a lo largo del tiempo con una penetración constante de 10 mm. Estas simulaciones permiten analizar la dirección de la fuerza de restitución calculada, a partir de la cual se puede deducir si la respuesta devuelta al usuario es estable.

La Figura 4.9a muestra los resultados obtenidos con la primera aproximación en la que se devuelve una normal de 45° cuando se detecta la esquina. Este cambio de dirección es muy brusco lo que hace indicar que la respuesta que recibe el usuario también lo será.

En la segunda aproximación (Figura 4.9b) se puede comprobar que el cambio de dirección en la fuerza resultante es progresivo obteniendo una sensación suave que permite al usuario recorrer la esquina sin ser expulsado obteniendo una sensación de redondeo en la discontinuidad.

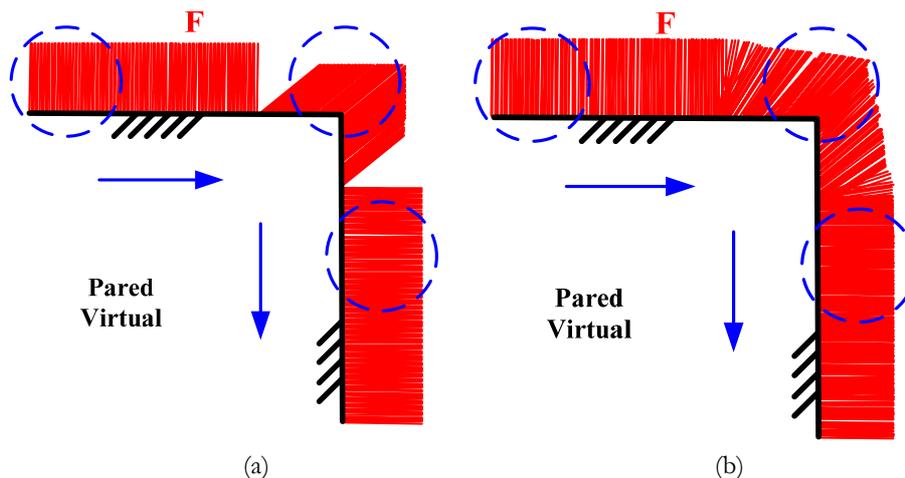


Figura 4.9: Primera aproximación (a) y segunda aproximación (b) para el cálculo de la dirección de la fuerza resultante en caso de existir una discontinuidad en la superficie.

La percepción de la respuesta háptica no sólo depende de las sensaciones físicas que se obtengan, sino también de factores psicológicos. De hecho, se han realizado numerosos estudios en los que se ha analizado la influencia de diferentes factores como la visión o el sonido en la percepción e inmersión de los sistemas hápticos. En el capítulo 7 se analizan algunos de estos trabajos.

Siguiendo esta misma línea de razonamiento, se ha planteado la posibilidad de orientar el cálculo de la restitución de esfuerzos para conseguir una respuesta con la que el usuario se sienta cómodo aunque no se reproduzca fielmente lo que ocurre en la realidad (por ejemplo la simulación de la esquina “redondeada” fruto de la segunda aproximación en la Figura 4.9b).

4.4 CÁLCULO DEL PUNTO DE CONTACTO

En el cálculo de los momentos, hay que distinguir dos conceptos: punto de aplicación y punto de contacto. Estos dos puntos son los que definen el vector necesario para calcular el par producido por cada contacto.

El punto de aplicación es el punto de la herramienta virtual en el cual se aplican las fuerzas o momentos como respuesta a la colisión con el entorno. En este caso, se ha considerado que el punto de aplicación sea el centro de masas de la herramienta virtual, de forma que los giros se realizarán en base a este punto.

Además, en el cálculo de momentos es necesario calcular un punto de contacto para cada una de las áreas de contacto implicadas en la colisión. La elección incorrecta de este punto de contacto puede provocar rotaciones no deseadas e inestabilidades en la respuesta. Kim (2003) proponen utilizar como punto de contacto aquel punto del objeto móvil que más haya penetrado en la maqueta virtual. Sin embargo, esta solución presenta algunos inconvenientes ya que este punto varía en función de la rotación del objeto provocando que el objeto oscile continuamente (Figura 4.10).

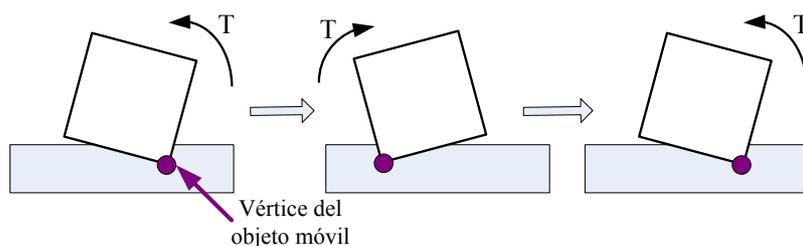


Figura 4.10: La variación en el punto de contacto del objeto móvil provoca que el objeto oscile.

Para evitar este problema una primera aproximación consiste en escoger un punto cualquiera de S_c con la premisa de que éstos sufren menores variaciones a lo largo del tiempo. Sin embargo, este punto no puede elegirse al azar ya que su posición puede condicionar la respuesta que sentirá el usuario. La Figura 4.11 muestra un ejemplo de dos respuestas de colisión diferentes según el punto de contacto elegido. Si se escoge un punto extremo del área en colisión del objeto estático, el vector formado por el centro de gravedad y este punto (\mathbf{r}) darían como resultado una fuerza y un momento tal que provocarían que el objeto girase (Figura 4.11a). En caso de escoger un punto de la zona central del área de contacto, la

respuesta final calculada sería únicamente una fuerza. El momento sería nulo por ser el vector \mathbf{r} paralelo a la fuerza resultante (Figura 4.11b).

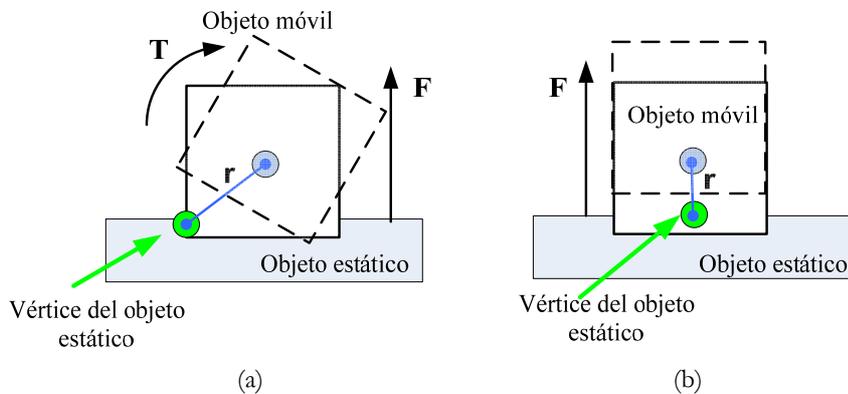


Figura 4.11: Respuesta háptica distinta en función del punto de contacto seleccionado: momento y fuerza (a), sólo una fuerza (b).

Para evitar dichas situaciones, la solución planteada consiste en escoger un punto de contacto representativo para cada una de las áreas de contacto identificadas en una colisión. Este punto se calcula como la media de los puntos medios de los voxels frontera asociados a cada área de contacto (Figura 4.12). La solución propuesta elimina posibles rotaciones no deseadas como las que se acaban de comentar.

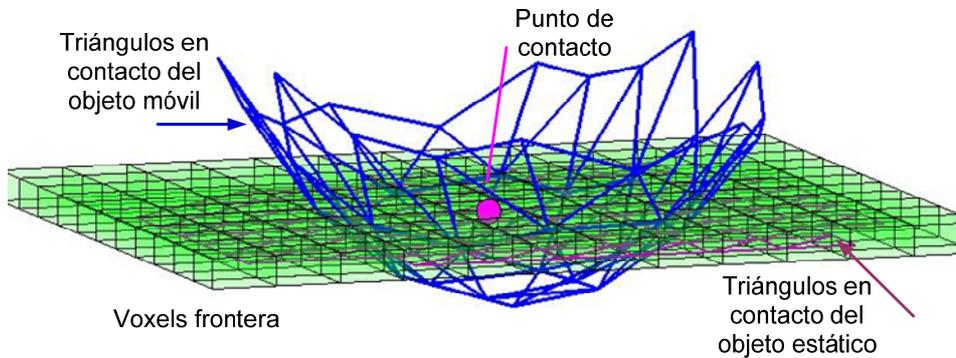


Figura 4.12: Punto de contacto calculado a partir de los voxels que contienen a los triángulos en contacto del objeto estático.

4.5 ÚLTIMAS CONSIDERACIONES

Una respuesta háptica estable, ya sea una fuerza y/o momento, se ve influenciada principalmente por tres factores: dirección de las fuerzas, penetración y puntos de contacto. En este capítulo se han abordado las problemáticas de la dirección y de los puntos de contacto mientras que el problema de la penetración se analizará en el siguiente capítulo.

La clasificación de los triángulos estáticos en contacto (\mathcal{S}) en áreas de contacto permite calcular una respuesta háptica agradable y estable para el usuario en casos con colisiones múltiples. La subdivisión de estas áreas a su vez en zonas permite detectar la existencia de discontinuidades geométricas y tratar estas situaciones de forma especial. Trabajos anteriores han demostrado la sensibilidad de los usuarios ante estos cambios bruscos de dirección que llegan a provocar inestabilidades en la respuesta háptica final. La solución propuesta obtiene un cambio suave de dirección en situaciones complejas mejorando así la percepción final del usuario y aumentando su inmersión en el entorno.

Para el cálculo de los momentos, se ha analizado el factor del punto de contacto y los problemas que implica un cálculo erróneo de este punto. Finalmente, se propone una solución que evita la aparición de problemas debidos a una mala elección de este punto.

CAPÍTULO 5

MÉTODOS PARA EL CÁLCULO DE LA PENETRACIÓN

Parte de este capítulo se ha presentado en:

Hernantes, J., Borro, D., Matey, L., and García-Alonso, A., "Analysis of Collision Detection in Dense Geometry Sets", Proceedings of the 10th International Fall Workshop Vision, Modeling, and Visualization (VMV 2005), pp. 99-106. Erlangen, Germany. November 16-18, 2005.

Hernantes, J., Díaz, I., Álvarez, H., and Borro, D., "Stable Collision Response Method in Complex Interactions", Poster Proceedings of the Third Workshop in Virtual Reality Interactions, and Physical Simulations (VRIPHYS'06), pp. 123-124. Móstoles, Madrid, Spain. November 6-7, 2006.

Hernantes, J., Borro, D., and García-Alonso, A., "Stable Haptic Response for Complex Interactions", Proceedings of the 2nd International Conference on Computer Graphics Theory and Applications (GRAPP'07), pp. 146-153. Barcelona, Spain. March 8-11, 2007.

En un sistema háptico, es importante garantizar una respuesta estable al usuario de cara a la seguridad. Para ello, es necesario evitar variaciones bruscas en la fuerza y/o par restituidos, hecho que puede incomodar al usuario provocando una percepción negativa del entorno.

En el capítulo anterior se han analizado diferentes factores que afectan a dicha estabilidad y como consecuencia a la percepción del usuario, como la dirección de la fuerza o el punto de contacto utilizado en el cálculo del momento. En este capítulo se analiza un tercer factor, la penetración.

Se denomina penetración a la distancia necesaria para separar dos objetos que se intersectan. Existen numerosos métodos en la bibliografía tal y como se enumeran a continuación, que calculan un valor exacto de la penetración producida. Sin embargo, algunas de estas soluciones no son válidas para sistemas hápticos debido a su elevado coste computacional. El objetivo principal es llegar a una solución de compromiso entre la precisión del cálculo y el coste computacional asociado.

En este capítulo se plantean cuatro métodos diferentes para el cálculo de la penetración. Para cada uno de ellos se analizan sus posibles ventajas e inconvenientes para ser aplicados a sistemas hápticos. La elección del método definitivo a utilizar se realizará en el siguiente capítulo en base a los experimentos realizados y teniendo en cuenta las variaciones en la magnitud de la penetración y los costes computacionales.

5.1 ESTADO DEL ARTE DE MÉTODOS DE PENETRACIÓN

El cálculo de distancias entre objetos geométricos es un problema que aparece usualmente en áreas como la robótica, path planning y los gráficos por computador. Una de las principales formas de calcularlo es utilizando distancia Euclídea. Sin embargo, los primeros métodos de la bibliografía consideran esta distancia nula cuando dos objetos se intersectan, de forma que no proporcionan ninguna información acerca de la intensidad real de la penetración.

Dados dos objetos A y B en intersección, la penetración entre estos dos objetos se define como la mínima distancia necesaria para separarlos en cualquier dirección. El principal objetivo de la distancia de penetración es cuantificar la profundidad de la intersección. Buckley et al. (1985) y Cameron et al. (1986) fueron los primeros en proponer el uso de una distancia negativa para medir la traslación necesaria para que los objetos no tengan puntos interiores en común.

Una de las métricas más utilizadas para el cálculo de la penetración son las sumas de Minkowski (Dobkin et al. 1993, Gilbert et al. 1988), sin embargo su cálculo exacto puede ser excesivo para aplicaciones en tiempo

real como el renderizado háptico. Por este motivo, Cameron (1997) propone una mejora del algoritmo propuesto por Gilbert denominado GJK, utilizando la coherencia temporal y las propiedades de convexidad de los polígonos. Trabajos posteriores también han añadido mejoras al algoritmo GJK (Choi et al. 2006, Van Den Bergen 2001).

Kim et al. (2002b) desarrollan un método incremental que calcula las superficies de las sumas de Minkowski mediante mapas de Gauss. Sin embargo, su elevado coste computacional lo hace incompatible para técnicas de renderizado háptico. Kim et al. (2002c) también presentan otra técnica en la que descompone los objetos en elementos convexos y utiliza jerarquías de volúmenes contenedores para aproximar recursivamente la estimación realizada del valor de la penetración.

Debido a la elevada frecuencia requerida en los sistemas hápticos, el coste computacional del cálculo de la penetración debe ser reducido. Así, McNeely et al. (1999) utiliza la representación del objeto basada en voxels aunque su precisión del método está limitada al tamaño de voxel utilizado. Otaduy (2004) utiliza una técnica basada en clusters mientras que Redon et al. (2005) proponen un método rápido que calcula un valor aproximado de la penetración utilizando el hardware gráfico. Sin embargo, este trabajo además de realizar dos lecturas de textura, determina valores erróneos de la penetración al no garantizar un cálculo apropiado del OBB que contiene las curvas de intersección que delimitan la parte del objeto móvil en intersección con el objeto estático (Figura 5.1).

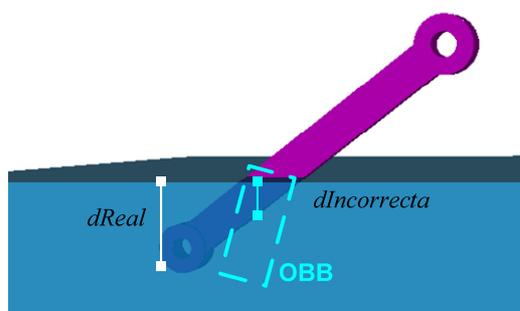


Figura 5.1: Error de cálculo del método de Redon. Se ha representado la penetración calculada ($d_{Incorrecta}$) y la penetración real (d_{Real}).

Sin embargo estos trabajos únicamente tienen en cuenta el movimiento traslacional del objeto para calcular el valor de la penetración. En aplicaciones como el renderizado háptico de 6 gdl además de la

traslación es necesario tener en cuenta el movimiento rotacional del objeto para calcular las fuerzas y momentos resultantes. Zhang et al. (2006) resuelven el problema definiendo una nueva métrica en el espacio de configuración (*C-space*) aunque su coste es todavía elevado para ser aplicado al renderizado háptico.

5.2 DISTANCIA PUNTO-PLANO: 1ª APROXIMACIÓN

Este método es una evolución basada en la primera aproximación del trabajo doctoral propuesto por Borro (2003), en el que para cada área de contacto, se calcula un plano de contacto (Π) perpendicular a la normal asociada a esa área ($\mathbf{n}_{\text{área}}$) y que pasa por el vértice de las facetas estáticas más alejado en la dirección de $\mathbf{n}_{\text{área}}$. La Ecuación (5.1) muestra el caso general para más de una área de contacto, donde $\mathbf{n}_{\text{área } i}$ representa la normal de contacto resultante en cada área, \mathbf{p}_{Ci} el punto de contacto y Π_i el plano de contacto calculado (ver Figura 5.2).

$$\Pi_i : \mathbf{n}_{\text{área } i} \mathbf{p}_{Ci} + l = 0, \quad \text{donde } l = -\mathbf{n}_{\text{área } i} \mathbf{p}_{Ci} \quad (5.1)$$

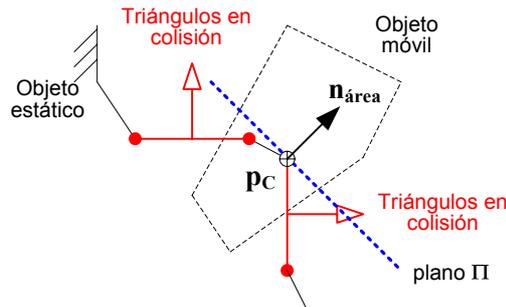


Figura 5.2: Plano de contacto Π situado en el vértice más alejado en la dirección de la normal $\mathbf{n}_{\text{área}}$.

Una vez calculado el plano de contacto, la penetración se calcula como la máxima distancia entre el plano de contacto y los vértices que se encuentran en el lado negativo del plano. Formalmente, se puede describir mediante la Ecuación (5.2) en la que V_M es el conjunto de vértices del objeto móvil y $dist()$ es la función que calcula la distancia de un punto (\mathbf{p}) a un plano.

$$d = \max \{ dist(\mathbf{p}, \Pi) \}, \quad \mathbf{p} \in V_M \wedge \mathbf{n}_{\text{área}} \cdot \mathbf{p} + l < 0 \quad (5.2)$$

En el trabajo presentado por Borro, únicamente se consideran triángulos en colisión aquellos que se encuentran en intersección. Es decir, no se tienen en cuenta aquellos triángulos del objeto móvil que se encuentran dentro del objeto estático pero que no se encuentran estrictamente en intersección con los triángulos estáticos. Por tanto, la penetración medida puede no coincidir con la real tal y como se muestra en la Figura 5.3.

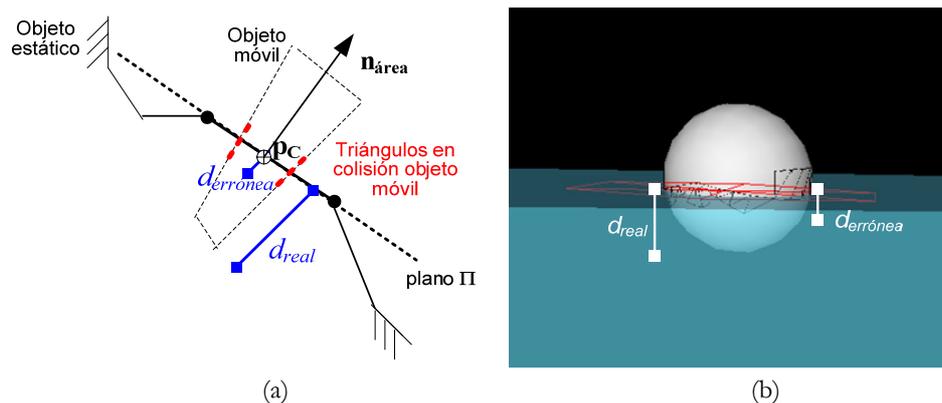


Figura 5.3: Error en el cálculo de la penetración utilizando únicamente las facetas en colisión del objeto móvil. Ejemplo 2D (a) y ejemplo 3D (b).

Con este método, la penetración calculada siempre es menor o igual, pero nunca mayor a la penetración real. Este aspecto es beneficioso de cara a la seguridad ya que se garantiza que la fuerza que se restituye al usuario no alcance niveles elevados de forma inesperada.

Sin embargo, el cambio más notable del método con respecto al planteado por Borro, consiste en utilizar la clasificación de voxels (capítulo 3) para detectar aquellos triángulos denominados internos y frontera. Así se consigue calcular una penetración correcta en casos como los de la Figura 5.3 y asegurando una continuidad que evita el paso a altas penetraciones en un breve intervalo de tiempo.

Este método presenta algunas limitaciones. La primera de ellas ocurre cuando se trabaja con superficies convexas, redondeadas debido a que estas superficies se aproximan mediante planos. La orientación del plano varía en función de las facetas estáticas en contacto detectadas al igual que lo hace el punto donde se define dicho plano.

Estos dos factores provocan que el plano pueda variar dando saltos discretos conforme cambien las facetas estáticas en contacto. Si el plano de contacto varía, también lo hará el valor de la penetración, provocando cambios bruscos en su magnitud que pueden llevar a discontinuidades en el sistema (Figura 5.4).

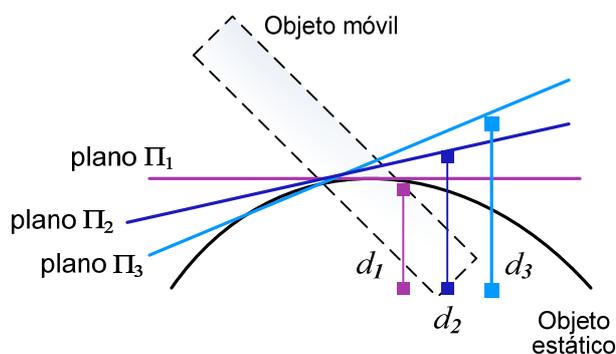


Figura 5.4: Posibles discontinuidades debidas al plano de contacto.

Aunque este método presenta algunas limitaciones a la hora de trabajar con ciertos objetos, tiene un excelente comportamiento en situaciones en las que el objeto estático está definido por grandes superficies planas. Además, como se verá en el capítulo 6, es de destacar su escaso coste computacional, aproximadamente de 1 ms. Estas características hacen que este método se tome como referencia para comparar el comportamiento del resto de métodos propuestos.

5.3 MÉTODO DE HIPPMANN MODIFICADO

Este método es una evolución del trabajo propuesto en Hippmann (2003), el cual está orientado a la simulación de la dinámica multibody, concretamente a la simulación de choques de peatones.

Tanto el método original como el modificado toman como punto de partida triángulos en contacto del objeto estático y móvil a partir de los cuales ha de calcularse un valor de penetración. Sin embargo, se han realizado algunos cambios para mejorar su rendimiento y aplicarlo en sistemas hápticos que requieren una mayor frecuencia de muestreo que las simulaciones mecánicas a las que este método original estaba destinado. Primero se resumirá brevemente el método original de Hippmann para luego pasar a describir las modificaciones realizadas en esta Tesis.

El método de Hippmann calcula la penetración en dos fases. En primer lugar, es necesario generar *elementos de contacto*, definidos en esta tesis como parejas de triángulos en contacto formadas por un triángulo del objeto estático y un triángulo del objeto móvil. Para detectar los triángulos que forman cada elemento de contacto, se toma como sistema de referencia el definido por cada triángulo en contacto del objeto estático y las coordenadas de los triángulos del objeto móvil se transforman a este nuevo sistema de referencia tal y como se muestra en la Figura 5.5.

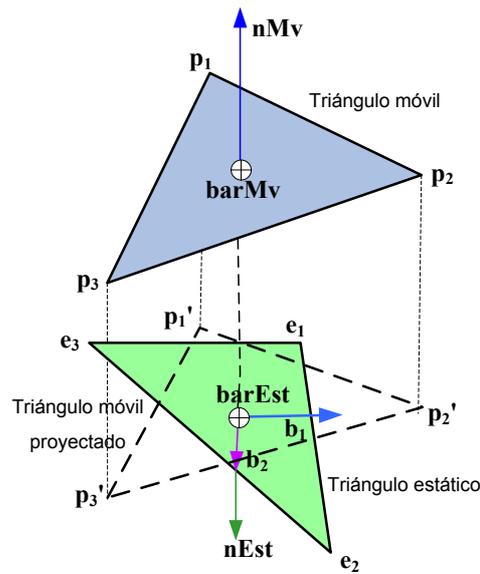


Figura 5.5: Elemento de contacto. Transformación del triángulo móvil al sistema de referencia definido por el triángulo estático.

El sistema de referencia definido por el triángulo estático se calcula mediante la Ecuación (5.3), tomando como origen del sistema el baricentro del triángulo estático.

$$\mathbf{b}_1 = \frac{\mathbf{e}_2 - \mathbf{e}_1}{|\mathbf{e}_2 - \mathbf{e}_1|} \quad (5.3)$$

$$\mathbf{b}_2 = \mathbf{nEst} \times \mathbf{b}_1$$

Donde \mathbf{e}_i representan vértices del triángulo estático y \mathbf{nEst} la normal del triángulo estático.

Para obtener las coordenadas del triángulo móvil (\mathbf{p}_i) en el sistema de referencia del triángulo estático, se realiza la transformación definida en la Ecuación (5.4), obteniendo las coordenadas del triángulo móvil proyectadas en el plano del triángulo estático (\mathbf{p}_i'), donde $i \in [1, 2, 3]$ indica los vértices del triángulo.

$$\mathbf{p}_i' = \mathbf{C}^{-1} \mathbf{p}_i \quad \text{siendo} \quad \mathbf{C} = [\mathbf{b}_1 \quad \mathbf{b}_2 \quad \mathbf{nEst} \quad \mathbf{barEst}]^T \quad (5.4)$$

Con las coordenadas de los dos triángulos (estático y móvil) expresadas en el mismo sistema de referencia, es necesario comprobar si forman un elemento de contacto. Para ello, se comprueba que el baricentro del triángulo estático está contenido en el triángulo móvil proyectado.

La segunda fase consiste en calcular un valor de penetración entre los dos triángulos que constituyen el elemento de contacto. El baricentro del triángulo estático es el origen de coordenadas del sistema de referencia y el valor de penetración viene dado por la componente z del baricentro del triángulo móvil en dicho sistema de referencia.

El método de Hippmann realiza la transformación de cada uno de los triángulos móviles en colisión detectados por cada uno de los triángulos estáticos en colisión.

El cambio más notable introducido para mejorar la eficiencia del método consiste en realizar un filtrado previo a la fase de transformación de las coordenadas de los triángulos móviles al sistema de referencia de los triángulos estáticos en colisión. Este filtrado consiste en detectar si el triángulo móvil forma parte de un elemento de contacto antes de realizar la transformación.

La detección se lleva a cabo mediante un rayo que se lanza desde el baricentro de cada triángulo estático en la dirección contraria a su normal. Si este rayo atraviesa al triángulo móvil, ambos triángulos formarán un elemento de contacto. En caso contrario, se lanzará otro rayo al siguiente elemento de M_c y así sucesivamente hasta encontrar un triángulo con el que exista intersección. Este proceso se repite para cada triángulo estático.

El hecho de testear la intersección de los triángulos antes de calcular las proyecciones permite reducir el coste computacional del método. La Figura 5.6 muestra un ejemplo de esta mejora realizando un experimento con un plano teselado con 30.000 triángulos y un toroide de 1.600 triángulos. Ambos métodos se testean en las mismas 400 posiciones

midiendo el tiempo de cálculo del valor de la penetración. La Figura 5.7 muestra el número de polígonos en contacto en cada posición.

Los resultados obtenidos con el método de Hippmann modificado muestran una mejoría significativa con respecto a los tiempos obtenidos con el método de Hippmann original. La mejora media del coste computacional se sitúa en torno a un 63%. Una vez demostrada la mejora realizada en este método, en adelante se hará referencia a dicha versión mejorada como Hippmann y no como Hippmann modificado.

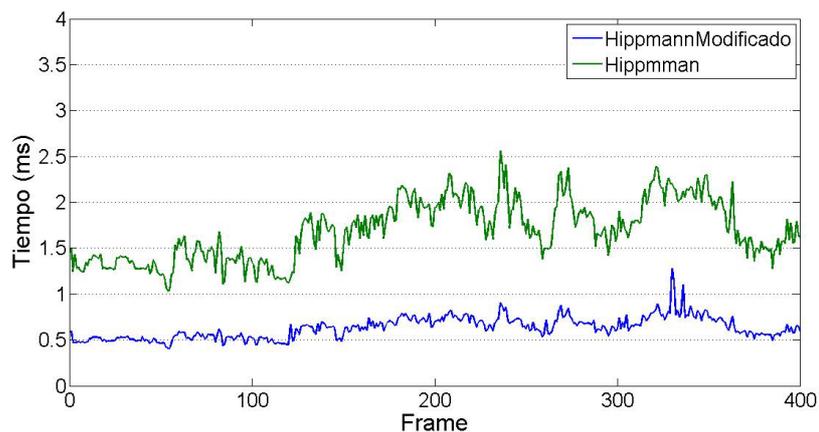


Figura 5.6: Coste computacional del cálculo de la penetración para el método de Hippmann y el método de Hippmann modificado.

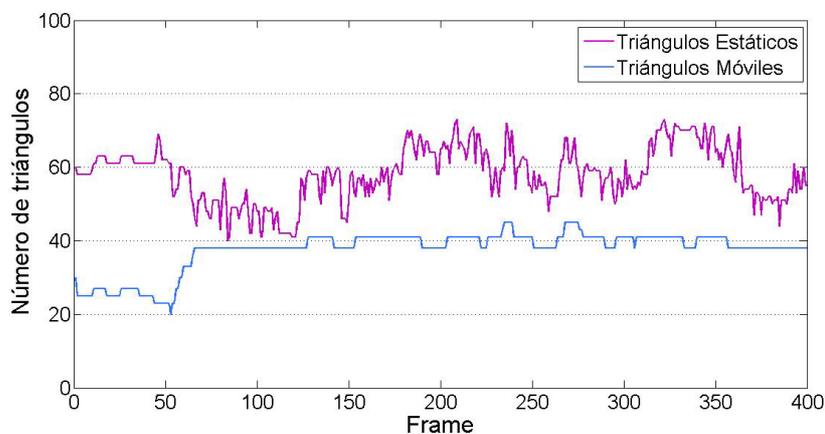


Figura 5.7: Número de triángulos en contacto.

Además de la etapa de filtrado, el método se ha paralelizado siguiendo las tácticas explicadas en el apartado 5.4.2.

5.4 MÉTODO RAYO -TRIANGULO

La filosofía de este método se basa en realizar un muestreo del volumen de intersección midiendo alturas a lo largo de todo este volumen para determinar la penetración entre los dos objetos. Como valor de la penetración puede utilizarse tanto el valor máximo de estas alturas como un valor medio, siendo este último el escogido en este trabajo.

Para determinar los puntos de muestreo de alturas se buscan pares de polígonos. Para ello, se lanza un rayo paralelo a la normal de contacto por cada triángulo estático en contacto (S) y se testea si este rayo interseca a algún triángulo en contacto del objeto móvil (M). La segunda fase del método utiliza este par de triángulos para el cálculo de una de las alturas del muestreo (ver Figura 5.8a).

Sin embargo, para que este muestreo sea correcto, las distancias entre los triángulos (h_i) deben ser mínimas, es decir, para cada triángulo estático se escoge aquel triángulo móvil que cumpla la condición descrita en la Ecuación (5.5), donde $triS$ es el triángulo estático, $triM$ el triángulo móvil y $dist$ representa la distancia entre dos puntos.

$$h_i = \min\{dist(triS_i, triM_i)\}, \quad triS_i \in S_c \wedge triM_i \in M_c \quad (5.5)$$

Si se utilizan las normales asociadas a cada triángulo estático para definir cada uno de los rayos del muestreo, se observa que los valores de alturas obtenidos no cumplen la expresión (5.5) e incluso se pueden obtener magnitudes de penetración mayores a las que realmente se han producido. La Figura 5.8a muestra un ejemplo de este caso.

Para que las alturas calculadas sean mínimas es necesario escoger el triángulo móvil que se encuentre por debajo del triángulo estático en la misma dirección en la que va a extraerse el objeto móvil. Esta normal se corresponde a la normal calculada en el apartado 4.3 (Figura 5.8b).

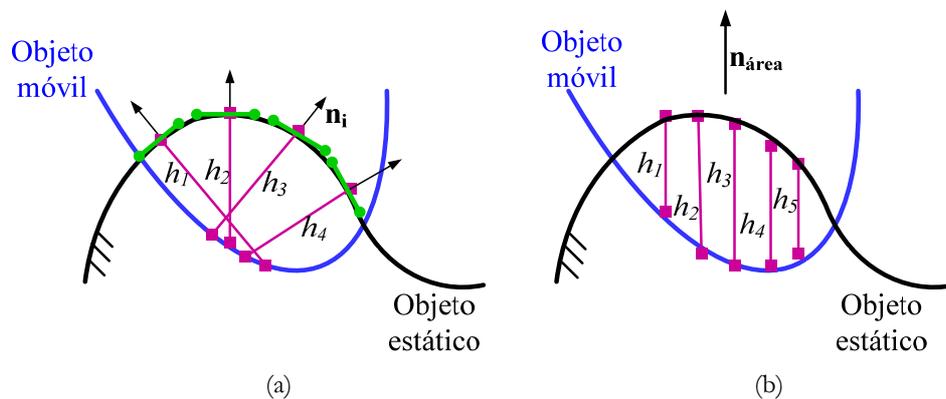


Figura 5.8: Discretización del volumen de intersección: utilizando rayos definidos con la normal asociada a cada triángulo estático representados en color verde (a) y utilizando rayos definidos mediante la normal resultante asociada a esa área de contacto (b). Las líneas denominadas h_i representan las alturas con las que se muestrea el volumen.

El método utilizado para determinar si existe intersección entre un rayo y un triángulo es el implementado por Möller (1997). Una vez detectada la intersección entre el rayo y el triángulo móvil, el segundo paso del algoritmo calcula el punto exacto de intersección en el plano del triángulo móvil (\mathbf{q}).

Finalmente, se calcula la distancia entre el baricentro del triángulo estático y el punto de intersección, obteniendo una altura del muestreo del volumen. Este proceso se repite para todos los triángulos estáticos en colisión.

La Figura 5.9 muestra un ejemplo 3D del muestreo que se lleva a cabo del volumen de intersección. En esta figura, el objeto azul representa los triángulos en contacto del objeto móvil mientras que la zona verde representa los triángulos en contacto del objeto estático y las líneas de color rojo los rayos del muestreo.

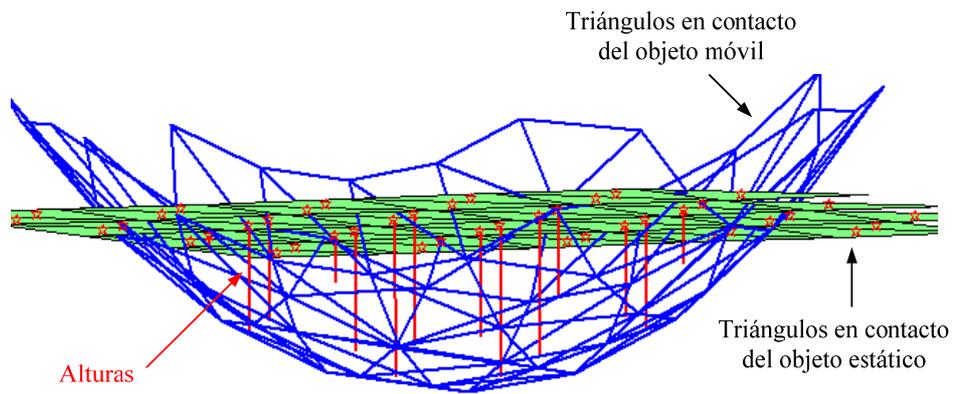


Figura 5.9: Ejemplo 3D del cálculo de la penetración con la técnica de los rayos.

La Figura 5.10 muestra de forma esquemática las diferentes fases de este método.

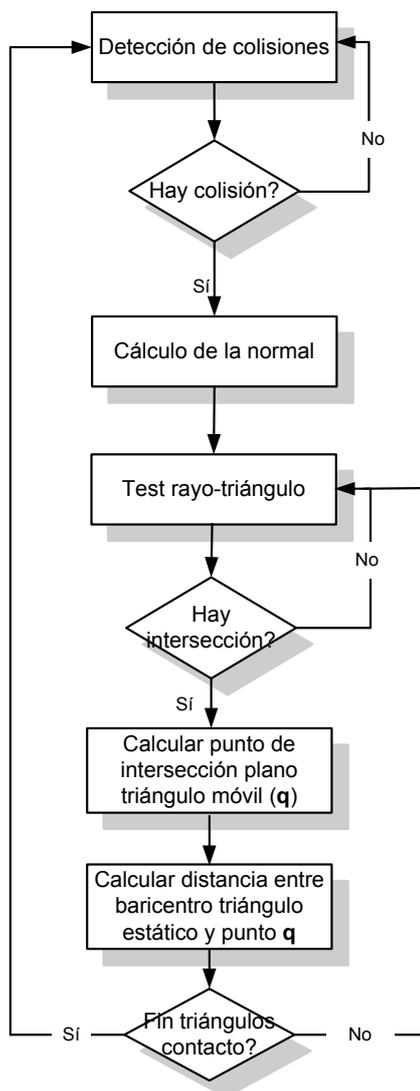


Figura 5.10: Esquema del método Rayo Triángulo.

5.4.1 PRECISIÓN DEL MÉTODO

Uno de los principales inconvenientes de este método es la influencia de la teselación de los modelos utilizados. El tamaño de los triángulos que definen el objeto estático condiciona el número de rayos con los que se realiza el muestreo del volumen de intersección. Cuánto más grandes sean estos triángulos, menor será el número de triángulos en contacto por lo que

también será menor el número de rayos con los que se calcula la penetración, disminuyendo así la precisión del resultado.

La diferencia entre las áreas de los triángulos del objeto estático y el móvil también es otro factor a tener en cuenta ya que también repercute en el cálculo de la penetración resultante. Si el área media de los triángulos estáticos supera el área de los triángulos del objeto móvil (es mayor que el doble del área de los triángulos del objeto móvil) pueden ocurrir situaciones como las que se muestra en la Figura 5.11. En este caso, el cálculo final de la penetración sería incorrecto ya que ninguno de los rayos lanzados desde los baricentros de las facetas en colisión del objeto estático intersectan los triángulos del objeto móvil.

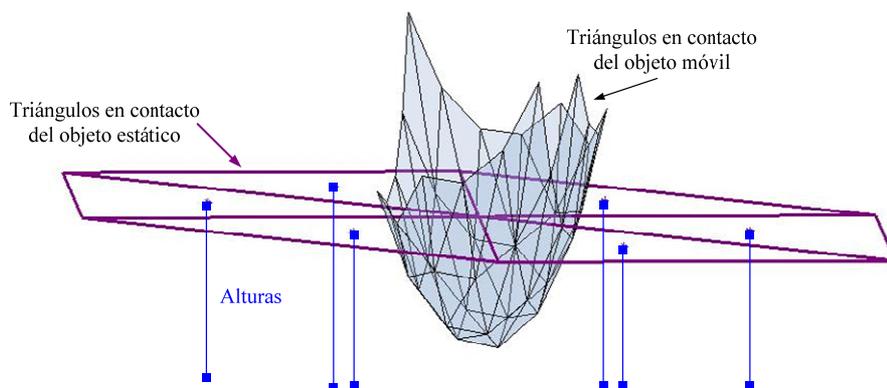


Figura 5.11: Error en el cálculo de la penetración ya que ningún rayo intersecta los triángulos del objeto móvil.

La solución planteada a este problema consiste en testear el área media de los triángulos en colisión de cada uno de los objetos y en base a esta proporción seleccionar la lista de triángulos en colisión más adecuada para lanzar los rayos. La lista elegida será aquella que tenga los triángulos de menor área para garantizar que el número de rayos con los que se discretiza el volumen sea suficiente para garantizar cierto nivel de precisión y evitar casos como los anteriormente planteados.

El objetivo de esta Tesis no ha sido el cálculo de una teselación óptima dependiente de cada modelo. Una opción para gestionar el problema de la teselación sería tener precalculadas varias teselaciones con diferente nivel de detalle y usar una u otra dependiendo de la precisión a la que se quiera llegar.

5.4.2 PARALELIZACIÓN DEL MÉTODO

A medida que el número de triángulos desde los que se lanzan los rayos aumenta, también lo hace la precisión del método a costa de un incremento del coste computacional del algoritmo por lo que es necesario buscar una solución de compromiso entre la precisión y el coste computacional.

El orden de complejidad del algoritmo es de $O(nm)$ siendo n el número de triángulos en contacto del objeto estático y m el número de triángulos del móvil. Para reducir este coste computacional, sobre todo en casos en los que la teselación de los modelos sea muy fina y se obtenga un número elevado de triángulos en contacto, se plantea la posibilidad de utilizar técnicas de paralelización. Esta paralelización del código se lleva a cabo mediante *threads* o hilos que permiten a una aplicación realizar varias tareas concurrentemente. De esta manera, el programa responde de manera más rápida a la interacción con el usuario.

Una de las principales ventajas de este método es su alto nivel de paralelización que resulta útil en el cálculo de la penetración con varias áreas de contacto. Una vez detectados los triángulos en colisión de cada uno de los dos objetos, éstos son separados en áreas de contacto. A partir de este momento, el cálculo de una normal, una penetración y el punto de contacto para cada una de estas áreas se puede realizar de forma independiente. Es por ello que estos cálculos pueden llevarse a cabo en dos hilos diferentes. La sincronización de los hilos lanzados se realiza mediante una barrera de sincronización, de modo que el hilo que antes termine tendrá que esperar a que finalicen el resto de los hilos. Esta espera es necesaria ya que el resultado obtenido en cada uno de los hilos se utiliza posteriormente para calcular un único resultado: una fuerza y/o un momento.

Sin embargo, hay casos en los que el reparto de tareas entre diferentes hilos puede no ser equitativo, provocando que alguno de estos hilos permanezca inactivo esperando a que el resto termine. Por este motivo se han distinguido dos propuestas diferentes de paralelización en función del número de rayos que se deben lanzar, ya que como se ha comentado anteriormente, condicionan la precisión y el coste computacional del método.

Si el número de triángulos de cada área se considera equilibrado, entonces se paraleliza cada una de las áreas existentes. En este caso, cada hilo trabaja con sus propias listas de triángulos (estáticos y móviles)

asignadas por lo que no se producen problemas de sincronización entre hilos y el coste añadido que supondrían las secciones críticas.

Por el contrario, si el número de triángulos de un área de contacto supera el doble de los triángulos de las áreas restantes, entonces se paraleliza el área que más triángulos contiene (en adelante, área más pesada). Para realizar esta paralelización, se particiona la lista de triángulos estáticos en contacto de dicha área y se asigna a cada hilo una partición diferente. En este caso tampoco surgen problemas para acceder a variables compartidas por lo que únicamente es necesario colocar la barrera de sincronización para esperar a que todos los hilos terminen.

El coste de crear un hilo se ha medido experimentalmente (con el hardware descrito en los siguientes párrafos) y viene a ser inferior a 1 ms aproximadamente. Aunque para muchas aplicaciones este coste sea casi despreciable resulta excesivo en relación con el orden de tiempos manejados para el cálculo de la penetración. Por este motivo, se opta por crear el hilo en preproceso y despertarlo o dormirlo según sea necesario en tiempo de ejecución. Experimentalmente, se ha comprobado que el coste medio para despertar un hilo es despreciable.

Hoy en día, con los procesadores multi-core o multi-núcleo lo ideal sería disponer de tantos núcleos como hilos se quieran utilizar en la paralelización del código. En caso contrario, es el sistema operativo el que se encarga de gestionar el acceso a la CPU de cada uno de los hilos. El acceso al procesador por diferentes hilos se realiza mediante los cambios de contexto, en el que se guarda la información del hilo que abandona la CPU y se deja el procesador libre a otro hilo. Estos cambios de contexto tienen un coste computacional que penaliza al método. Realizar una estimación de esta penalización resulta complicado ya que no se conocen a priori el número de cambios de contexto que se van a realizar durante la ejecución.

La Figura 5.12 muestra la mejora que se obtiene paralelizando el código del cálculo de la penetración para el caso en el que se detectan dos áreas en colisión. Esta prueba se ha realizado con dos paralelepípedos perpendiculares en forma de L (60.000 triángulos) y un toroide (1.600 triángulos) que colisiona en dos áreas de contacto simultáneamente. El PC utilizado para la prueba es un Intel Core 2 Duo a 2,4 GHz. El coste computacional del cálculo de la penetración para cada una de estas áreas se ha analizado a lo largo de 900 posiciones diferentes testeadas con la palanca háptica (Anexo A), obteniendo una mejora media de hasta un 30%. La

Figura 5.13 muestra el número de triángulos en colisión para cada una de estas áreas.

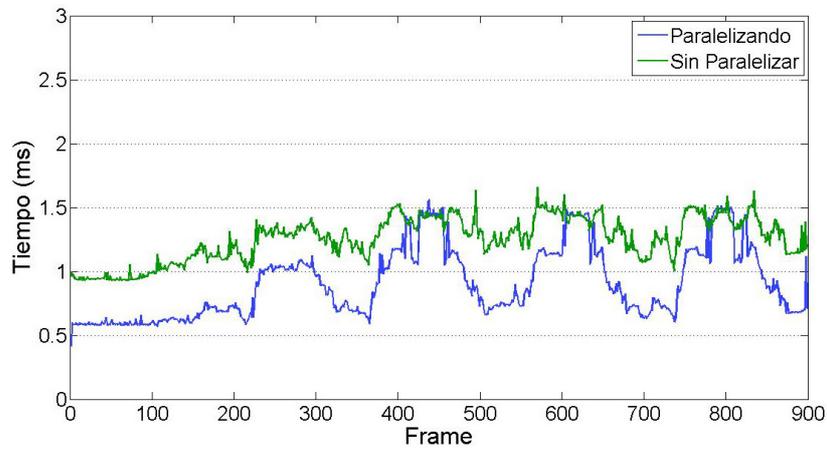


Figura 5.12: Tiempos de cálculo de la penetración con dos áreas de contacto.

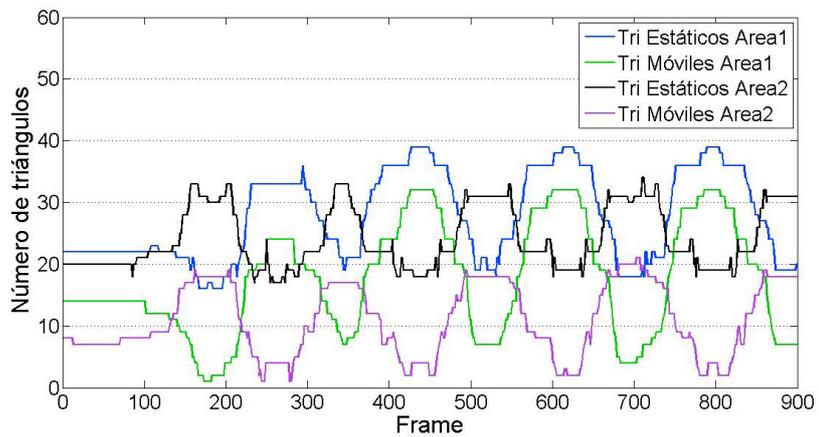


Figura 5.13: Número de triángulos en colisión asociados a cada área de contacto.

La Figura 5.14 muestra el esquema del método basado en rayos tras paralelizar su código. El diagrama representa la paralelización que se lleva a cabo utilizando un PC de 2 núcleos.

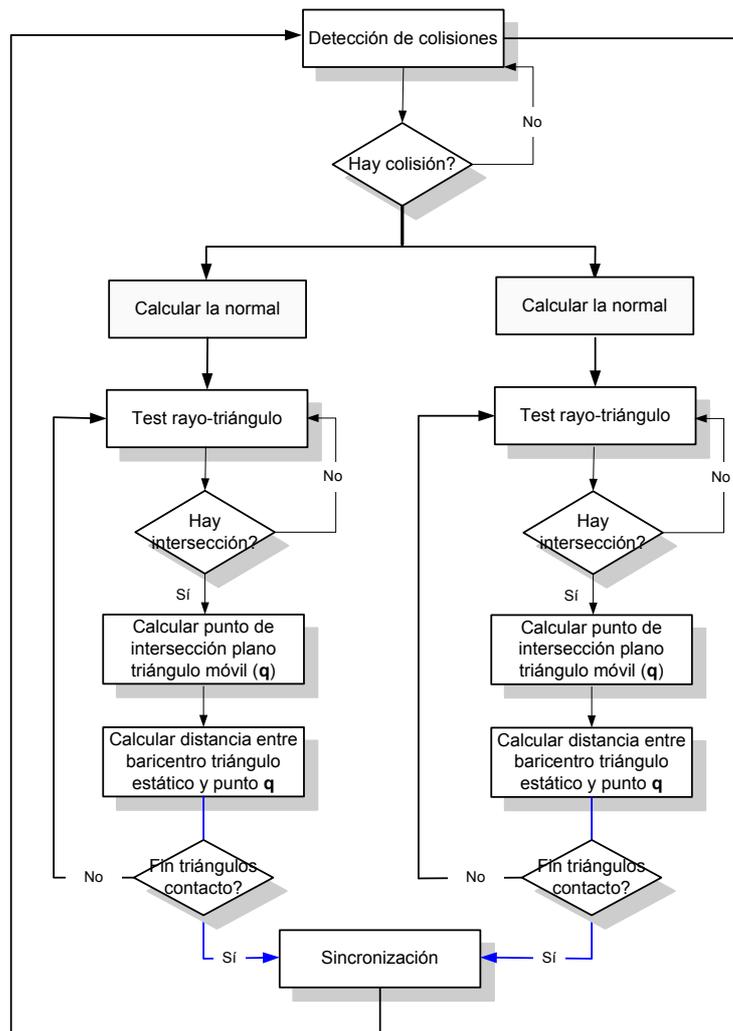


Figura 5.14: Paralelización del método basado en rayos.

5.5 MÉTODO Z-BUFFER

En los últimos años se ha dado un paso importante a la hora de aliviar la carga computacional de la CPU trasladando tareas de cálculo a la tarjeta gráfica. Para obtener cada vez una mayor calidad en los escenarios, se han añadido nuevas capacidades a las tarjetas gráficas que permiten programar ciertas tareas en la GPU (Unidad de Procesado de Gráficos).

Hoy en día, los recursos del procesador gráfico pueden utilizarse para tareas de propósito general como cálculos matemáticos ajenos al dibujo de escenarios (Owens et al. 2005). Estas técnicas son conocidas como GPGPU (GPU General Programming). Aunque la velocidad de cómputo del hardware gráfico de la GPU es más lenta que la de la CPU, las últimas generaciones de tarjetas gráficas permiten una mayor paralelización de las tareas a realizar por lo que aumenta enormemente su capacidad de cálculo permitiendo procesar numerosos píxel en paralelo de forma independiente. Además las GPUs actuales están optimizadas para el cálculo con valores de coma flotante, predominantes en los gráficos 3D.

Utilizar la GPU para resolver determinados problemas implica transformar dicho problema en una operación de dibujo. Las GPUs actuales tienen tres unidades programables: *vertex*, *geometry* y *píxel shaders*. El *vertex shader* permite modificar propiedades de un vértice para que repercutan en la geometría del objeto al que pertenece mientras que *geometry shader* permite añadir vértices a la geometría original. Por último, la unidad de *píxel shader* aplica un programa a cada píxel de la imagen que se dibuja. Este tratamiento individual de los píxeles permite que se realicen cálculos en tiempo real principalmente relacionados con la iluminación.

Los pasos que se siguen para resolver un problema empleando para ello la GPU y el píxel shader son los siguientes:

1. **Inicialización:** el contexto de renderizado se inicializa y los datos son transferidos de la memoria principal a la memoria de la tarjeta gráfica.
2. **Renderizado:** se dibuja la imagen. La GPU aplica el *píxel shader* a cada uno de los píxeles de la imagen final.
3. **Lectura:** se transfiere la imagen final de la memoria de la tarjeta gráfica a la memoria principal.

Cuando se trabaja con la GPU, el principal cuello de botella se encuentra en el paso 3, ya que se realiza una sincronización y una transferencia de los datos entre la memoria de la tarjeta gráfica y la memoria principal (Figura 5.15).

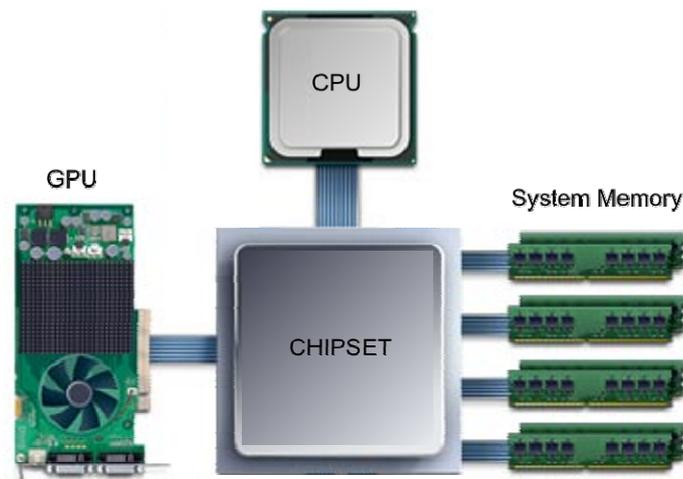


Figura 5.15: Transferencia de la información entre la GPU y la memoria principal.

Sacando partido a los avances de las tarjetas gráficas actuales, se plantea un método que permite aprovechar la capacidad de cómputo de la GPU para calcular la penetración producida en una colisión. Para ello se parte de la información de las listas de triángulos en contacto de ambos objetos (S_c y M).

5.5.1 FASES DEL MÉTODO

El método se basa en el algoritmo del z-buffer utilizado en el área de gráficos por computador para detectar facetas ocultas u objetos que están ocluidos por otros que también forman parte de la escena.

El algoritmo del z-buffer calcula las coordenadas de profundidad (coordenada z) de los objetos de la escena para determinar su visibilidad. Cuando un objeto es dibujado por la tarjeta gráfica, la profundidad del píxel generado se almacena en un buffer de datos (z-buffer). Este buffer se representa mediante un array de dos dimensiones (x - y) con un elemento por cada píxel de la pantalla. Si algún otro objeto de la escena se tiene que renderizar en el mismo píxel, la tarjeta gráfica compara las dos profundidades y elige el más cercano al observador. La profundidad elegida es entonces salvada en el z-buffer, reemplazando a la antigua. El z-buffer permite a la tarjeta gráfica reproducir correctamente la percepción de profundidad de forma que los objetos cercanos oculten a los más lejanos (Figura 5.16).

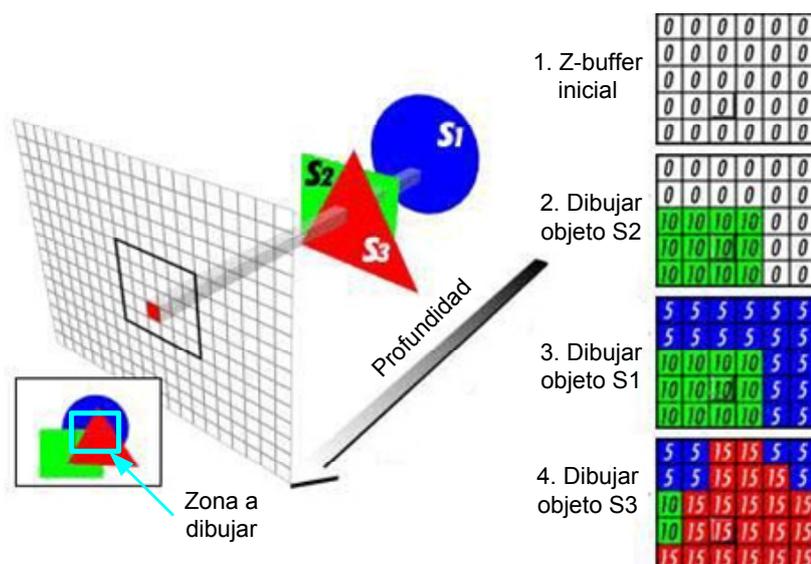


Figura 5.16: Funcionamiento del z-buffer.

La información calculada por el z-buffer es por lo tanto indispensable para visualizar correctamente la escena, pero además puede ser de utilidad para resolver problemas como el expuesto en este capítulo.

Para poder utilizar el z-buffer en el cálculo de la penetración, se han de seguir los siguientes pasos. El primero consiste en establecer el punto de vista de la cámara desde el cual va a ser aplicado el algoritmo del z-buffer, ya que el dibujado de un píxel y su valor de profundidad están condicionados a la posición en la que se encuentra el observador. Para definir la posición de la cámara y la dirección a la que mira, se utiliza la normal al área de contacto ($\mathbf{n}_{\text{área}}$) y el punto de contacto (\mathbf{p}_C). La posición final de la cámara se calcula sumando un cierto desplazamiento (h) al punto de aplicación en la misma dirección que $\mathbf{n}_{\text{área}}$ mientras que la dirección a la que mira la cámara será la contraria a $\mathbf{n}_{\text{área}}$.

Una vez posicionada la cámara, se dibujan las listas de triángulos desde el punto de vista calculado utilizando una proyección ortográfica para evitar la distorsión que producen las técnicas normales de proyección. En primer lugar, se envía a la tarjeta gráfica (GPU) la lista de triángulos en contacto del objeto estático (S_s) y posteriormente la del objeto móvil (M_o). Siguiendo el mismo concepto explicado en el apartado 5.4, la diferencia entre las profundidades obtenidas para cada objeto proporciona una aproximación de la penetración entre ambos objetos.

Para que la CPU pueda acceder a la información calculada por el z-buffer, la diferencia entre los valores de profundidad obtenidos por cada objeto se guarda en una textura siendo ésta la información enviada desde la GPU a la CPU. Aunque el uso de las tarjetas gráficas para realizar operaciones matemáticas permite reducir el coste computacional debido a su alto grado de paralelización, el verdadero cuello de botella se encuentra en la lectura o paso de información entre la GPU y la CPU. Por este motivo es necesario reducir al mínimo el número de lecturas necesarias desde la CPU para tener acceso a los resultados obtenidos mediante la GPU. El uso de los shaders permite reducir este tráfico de información con su correspondiente ahorro computacional.

La Figura 5.17 muestra las fases del proceso comentado utilizando y sin utilizar los *pixel shaders*. Si no se utilizan los shaders es necesario realizar dos lecturas de la tarjeta gráfica para obtener los valores de profundidad de los dos objetos en colisión (Redon and Lin 2005). Y una vez leída esta información, la diferencia entre las profundidades de cada textura se realiza en la CPU con el coste computacional añadido en función del tamaño de la textura (Figura 5.17a). En cambio utilizando los shaders, la diferencia entre las profundidades se realiza en paralelo para todos los píxeles directamente en la GPU, siendo necesaria únicamente una lectura de esta información desde la CPU (Figura 5.17b).

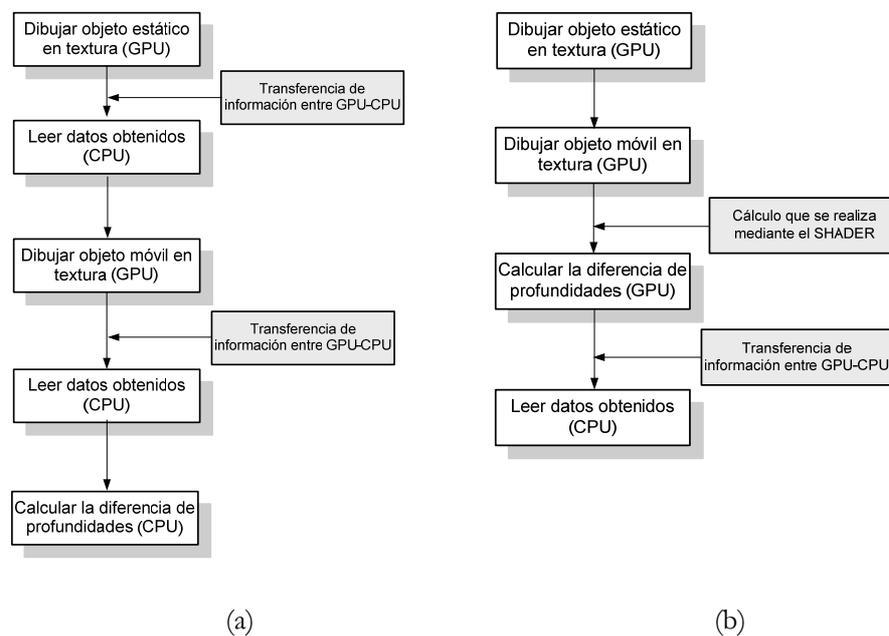


Figura 5.17: Esquema del método sin utilizar shaders (a) y utilizando shaders (b).

A continuación se muestra la solución aportada de forma esquemática (Figura 5.18).

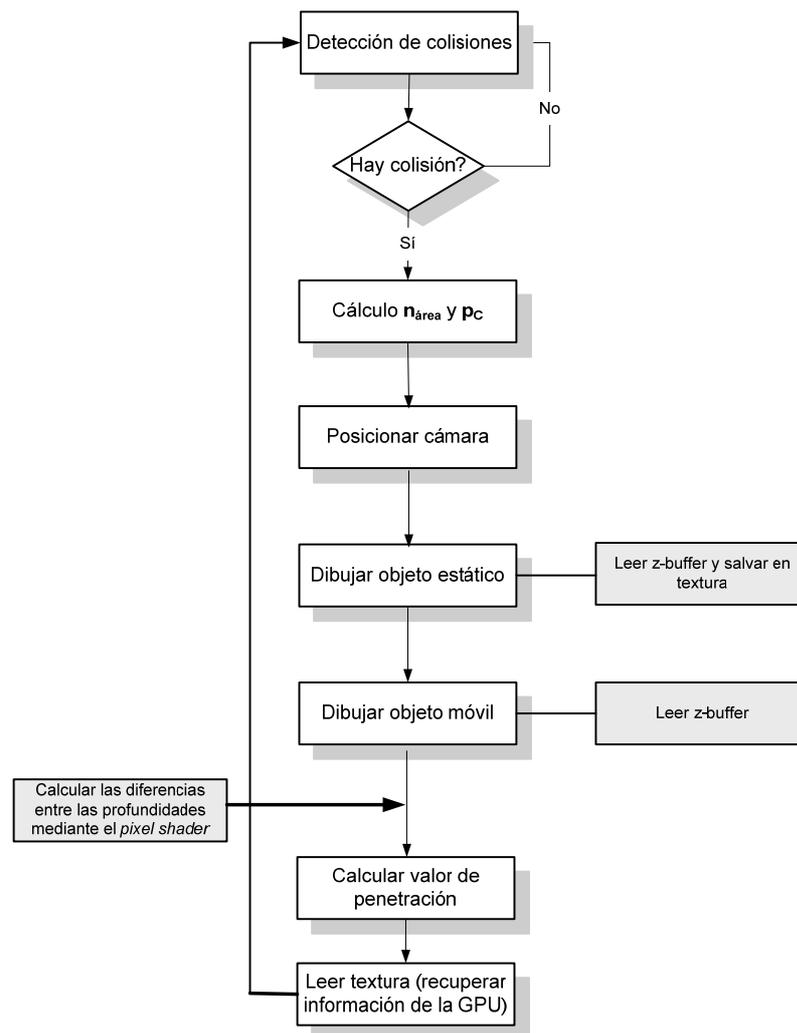


Figura 5.18: Diagrama de flujo del método que utiliza el z-buffer.

5.5.2 FACTORES QUE INFLUYEN EN LA EFICACIA DEL MÉTODO

Este método basado en el z-buffer permite calcular un valor de penetración con gran precisión, sin embargo está condicionado por diversos parámetros que se analizan a continuación.

Posicionamiento de la cámara

Colocar la cámara en una posición correcta con un punto de vista adecuado es indispensable para el correcto funcionamiento del método. Un

error en la posición de la cámara puede provocar errores en el dibujo de las facetas y por lo tanto errores en el cálculo de las profundidades que influirán en el valor final de la penetración.

La cámara debe desplazarse en base a la normal en la que se va a aplicar la fuerza de restitución ($\mathbf{n}_{\text{área}}$) para que de este modo el volumen de visión abarque la zona de contacto a analizar en su totalidad. Sin embargo, su punto de vista debe ser contrario a $\mathbf{n}_{\text{área}}$.

La Figura 5.19a muestra un posicionamiento incorrecto de la cámara. En este caso sólo se dibujarían los triángulos en colisión de uno de los objetos, de forma que no sería posible calcular el volumen de intersección entre ambos objetos. Por el contrario, en la Figura 5.19b la cámara se sitúa por encima de los dos objetos de forma que su volumen de visión contendría la intersección entre ambos.

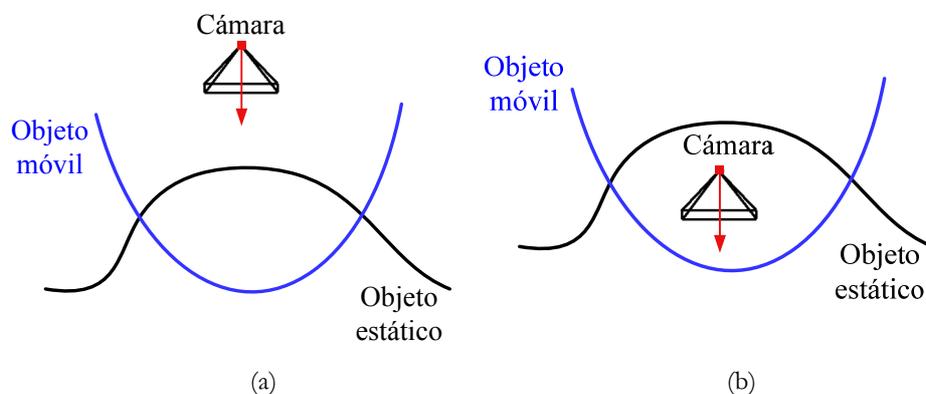


Figura 5.19: Posicionamiento incorrecto de la cámara que impide dibujar los triángulos del objeto estático (a) y posicionamiento correcto de la cámara (b)

Volumen de dibujado

Una vez que la cámara ha sido colocada correctamente, el tamaño del volumen de visión es otro factor que juega un papel importante en el cálculo de la penetración. El volumen utilizado debe ajustarse en la medida de lo posible a la zona de contacto que se quiere analizar. Un volumen de visión con un tamaño inferior al área en colisión no tendría en cuenta todos los posibles píxeles en intersección por estar fuera de este volumen, perdiendo así información necesaria y que podría influir en el valor final de la penetración. Si por el contrario, el volumen es demasiado grande, se

perdería precisión ya que se reduce el número de píxeles que representan la zona en colisión.

Por lo tanto, el tamaño óptimo del volumen de visión depende en gran medida del área de la zona de intersección. Para ajustar su tamaño de forma correcta en cada caso, se calcula la caja contenedora que contiene los triángulos en colisión del objeto estático y cuyas dimensiones de anchura y profundidad delimitarán el volumen de visión. La altura de este volumen quedará definida por el desplazamiento (h_c) utilizado para posicionar la cámara aplicado en ambas direcciones.

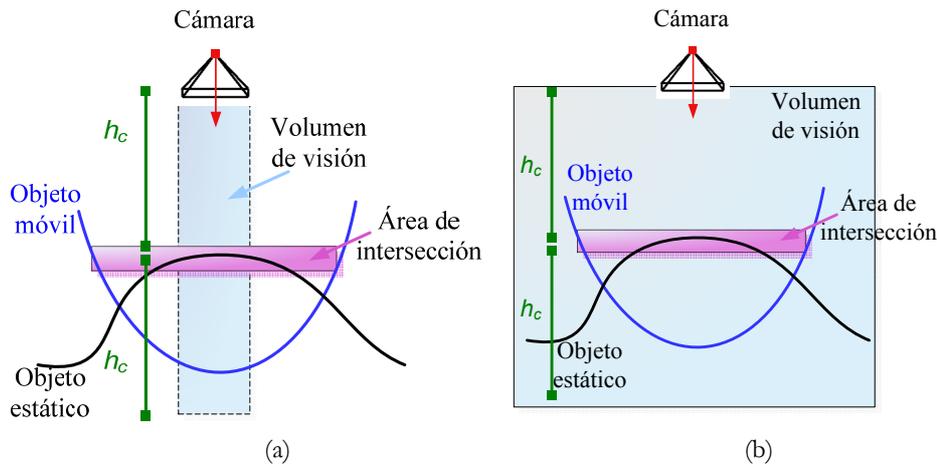


Figura 5.20: Volumen de visión que no engloba el total del área de intersección (a) y volumen de visión acorde al tamaño del área de intersección (b).

Tamaño de la textura

El tamaño de la textura que se utiliza para guardar la información determina la precisión del método. Cuanto mayor sea su tamaño, mayor será también el nivel de precisión obtenido aunque el tiempo necesario de lectura desde la CPU también se verá incrementado.

Por el contrario, utilizando una textura de menor tamaño, la precisión disminuye pero también lo hace el tiempo de lectura. Por lo tanto es necesario establecer una solución de compromiso entre el nivel de precisión que se quiere obtener en los cálculos y el tiempo del que se dispone para realizarlos.

La elección del tamaño de la textura más adecuado en cuanto a su precisión y su tiempo de lectura se ha realizado experimentalmente. Para

ello se han llevado a cabo diferentes experimentos en los cuales se repite el cálculo de la penetración en un path grabado utilizando diferentes tamaños de textura. La Figura 5.21 y la Figura 5.22 muestran los resultados obtenidos con un plano como escenario (11.000 triángulos) y una esfera como herramienta virtual (960 triángulos). En color negro se muestran los valores de penetración utilizados como referencia en la comparativa, calculados mediante el método de distancia punto-plano (apartado 5.2). A pesar de las limitaciones que presenta este método, se ha escogido por considerarse eficiente y preciso para este tipo de escenarios.

Como era de esperar, los resultados obtenidos muestran que cuanto mayor sea el tamaño de la textura utilizada, mayor es la precisión obtenida. Las texturas de menor tamaño 32x32, 64x64 y 128x128 se descartan ya que se obtienen resultados incorrectos o de muy baja precisión y con una gran variabilidad entre *frames* consecutivos lo que puede afectar negativamente al cálculo de la respuesta háptica. Por el contrario, con texturas de mayor tamaño, se observa que los valores obtenidos son más cercanos a los valores de referencia.

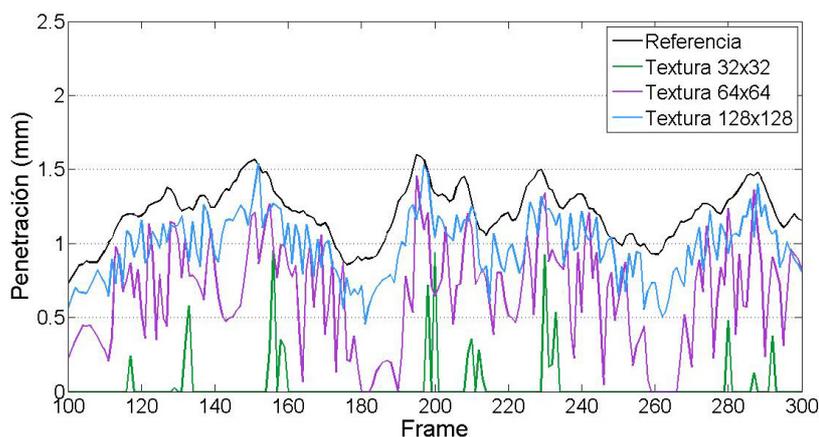


Figura 5.21: Penetración obtenida con tamaños de textura de 32x32, 64x64 y 128x128. En color negro se representa la penetración calculada como referencia.

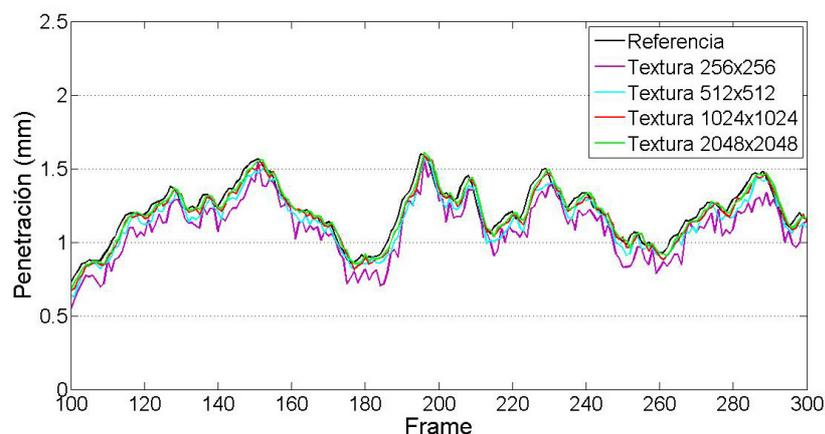


Figura 5.22: Penetración obtenida con tamaños de textura de 256x256, 512x512, 1024x1024 y 2048x2048. En color negro se representa la penetración calculada como referencia.

Sin embargo, además de la precisión se debe tener en cuenta el tiempo de transferencia de los datos de la tarjeta gráfica a la memoria principal, que como se ha comentado anteriormente, constituyen el principal cuello de botella del uso de la GPU. La Figura 5.23 muestra los tiempos empleados en el cálculo de la penetración por cada una de las texturas. En base a estos resultados, el tiempo empleado por las texturas de 2048x2048 y 1024x1024 se considera excesivo para poder ser utilizado en una aplicación de renderizado háptico ya que supera los 6 ms, por lo que estos tamaños de textura son descartados a pesar de la precisión obtenida. Este coste computacional únicamente hace referencia al cálculo de la penetración al que habría que añadir el coste de la detección de las colisiones, además del coste computacional asociado al cálculo del resto de parámetros necesarios para la respuesta háptica.

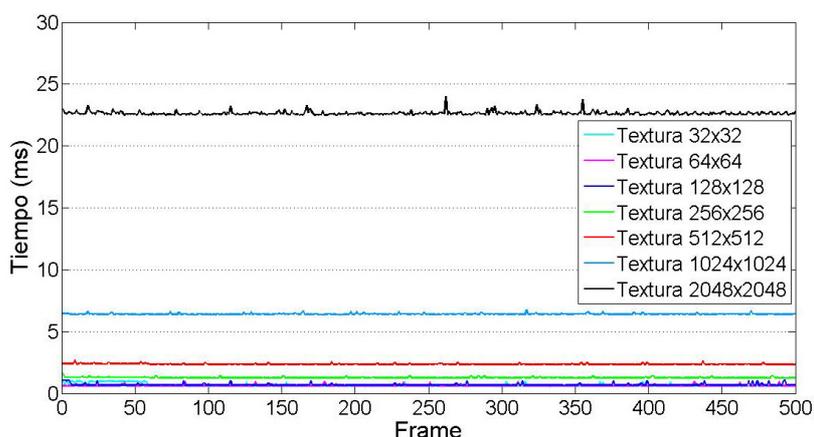


Figura 5.23: Tiempos de cálculo de la penetración con diferentes tamaños de textura.

La Figura 5.24 muestra los mismos datos que la Figura 5.23 pero se han eliminado las texturas de mayor tamaño para poder apreciar mejor los tiempos obtenidos con el resto de texturas utilizadas. Teniendo en cuenta que las texturas de tamaño 32x32, 64x64 y 128x128 han sido descartadas por su baja precisión y las variaciones que se obtienen, y las de mayor tamaño (1024x1024 y 2048x2048) por su alto coste computacional, la elección de la textura se reduce a los tamaños 256x256 y 512x512. Finalmente, se ha escogido la de menor tamaño ya que se considera que se alcanza un mejor compromiso entre precisión y tiempo de cálculo.

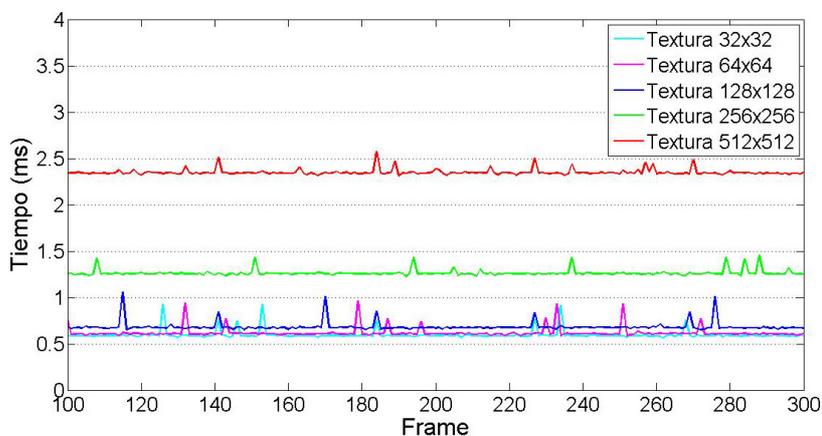


Figura 5.24: Tiempos de cálculo de la penetración con tamaños de texturas más pequeños.

Coste computacional

Al igual que en los métodos descritos anteriormente, se han analizado diferentes alternativas para reducir el coste computacional del proceso. Sin embargo, este método presenta dos principales restricciones que limitan la efectividad de cualquier optimización.

La principal restricción de este método es que no puede ser paralelizado para calcular la penetración en diferentes áreas de contacto. Las propuestas anteriores permiten paralelizar el código realizando los cálculos pertinentes en la CPU de forma independiente para cada área de contacto, sacando partido al doble núcleo del procesador. Sin embargo, en este método los cálculos se realizan en la única tarjeta gráfica de la cual dispone el PC, por lo que el acceso a sus recursos ha de hacerse de forma secuencial para cada área. Este hecho sumado al tiempo de transferencia inevitable de la información entre la GPU y la CPU, tienen como consecuencia que el tiempo total de cálculo de la penetración (t_{total}) sea proporcional al número de áreas de contacto detectadas ($numAreas$).

$$t_{total} = numAreas \times t_{penetración} \quad (5.6)$$

donde $t_{penetración}$ es el tiempo de cálculo de la penetración para una única área de contacto.

La segunda limitación del método está relacionada con el intercambio de información de colisión con el lazo de control (véase la arquitectura del sistema en el apartado 6.1). En los métodos anteriores, la detección y respuesta a dicha colisión se calcula en el propio módulo de colisiones por lo que la frecuencia del intercambio de mensajes con el lazo de control depende del tiempo de cálculo de dicha información.

Sin embargo, este método utiliza la tarjeta gráfica para realizar los cálculos correspondientes por lo que además del coste computacional que supone el intercambio de información entre la GPU y la CPU, es necesaria una sincronización entre el módulo de visualización y el módulo de colisiones para intercambiar la información necesaria para el proceso. Este intercambio de información se realiza a una frecuencia limitada por el *frame rate* del módulo de visualización normalmente inferior a la frecuencia de las colisiones y que varía en función del tamaño de los modelos utilizados, su resolución o el uso de visualización estereoscópica. En el apartado 6.4.3 se han realizado experimentos con este método en los que se analiza la

influencia en la percepción de la respuesta háptica restituida por el sistema del retraso existente entre la frecuencia del módulo de control y el cálculo de la respuesta de colisiones utilizando este método.

5.6 RESTITUCIÓN VISUAL

Una de las desventajas del método de penalización propuesto radica en que visualmente el usuario percibe cómo la herramienta virtual penetra en los objetos virtuales del escenario. En caso de poder implementar una rigidez elevada, la penetración visual es mínima y el efecto apenas es perceptible. En caso contrario, esta penetración visual puede llegar a degradar la percepción de rigidez del entorno, disminuyendo el realismo del contacto. Para solucionar este hecho, se puede combinar el método desarrollado con los métodos basados en restricciones.

Los métodos basados en restricciones utilizan las técnicas de acoplamiento virtual para restringir visualmente el movimiento del objeto móvil en las superficies de contacto. Si bien es cierto que la sensación táctil con estos métodos no es muy buena, visualmente consiguen que el movimiento del objeto móvil desacoplado sea del todo realista. Por ello, mediante una combinación de ambos métodos se puede obtener un método muy completo tanto para la percepción táctil como para la visual.

Para el cálculo de la respuesta háptica debida a la interacción del usuario con el entorno virtual, se utilizará el método de renderizado propuesto, sin embargo para la visualización de la herramienta virtual, se utilizará el método de *god-object* propuesto por Ortega (2006). Además, la misma información calculada en el algoritmo de respuesta (normales y puntos de contacto de las áreas en colisión) es la que se utiliza para calcular la dinámica del objeto móvil.

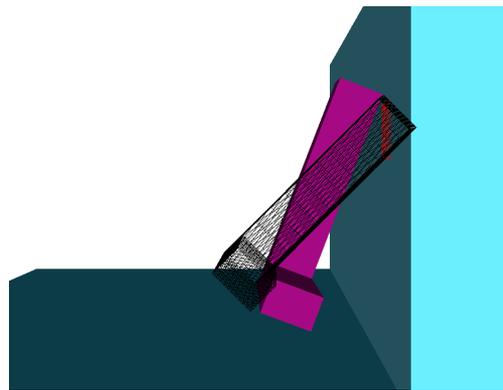


Figura 5.25: El cálculo de la respuesta háptica se realiza en base a la posición real del objeto manipulado por el usuario (representado mediante sus triángulos), mientras que visualmente se representa el objeto móvil desacoplado cuya posición es calculada mediante el método del *god-object*.

5.7 CONSIDERACIONES FINALES Y LIMITACIONES

Se han presentado cuatro métodos diferentes para el cálculo de la penetración analizando sus ventajas e inconvenientes. El método distancia punto-plano es el método con menor coste computacional, aunque presenta numerosas limitaciones para su aplicación. En el otro extremo, se encuentra el método del z-buffer que proporciona una precisión de píxel pero cuyo coste computacional asociado supera el deseado para las aplicaciones de renderizado háptico. El objetivo es llegar a un compromiso entre el nivel de precisión obtenido y el coste computacional.

Para reducir el coste computacional, se han planteado métodos de paralelización sacando partido a la arquitectura multicore. En el siguiente capítulo se realizarán diferentes pruebas tanto con modelos teóricos como con modelos reales para contrastar los resultados obtenidos con cada uno de ellos y escoger el que mejor se adecue a los requerimientos del sistema y que además proporcione una respuesta estable y agradable con la que el usuario se encuentre cómodo al interactuar en el entorno.

Tal y como se comprobará en los experimentos realizados, algunas de las soluciones propuestas no son válidas para calcular una respuesta háptica estable en situaciones complejas, bien por sus limitaciones o por su coste computacional asociado, aunque pueden utilizarse como métodos de referencia en situaciones determinadas para realizar una comparativa entre los diferentes métodos.

Por último, hay que resaltar que todos ellos utilizan la geometría como punto de partida para realizar sus cálculos por lo que errores en el mallado original de los modelos pueden provocar errores en los cálculos finales.

CAPÍTULO 6

RESULTADOS EXPERIMENTALES

Parte de este capítulo se ha presentado en:

Borro, D., Hernantes, J., Mansa, I., Amundarain, A., García-Alonso, A., and Matey, L., "Virtual Maintenance for Dense Environments", Proceedings of the Laval Virtual, 7th International Conference on Virtual Reality 2005 (VRIC 2005), pp. 141-148. Laval, France. April 20-24, 2005.

Hernantes, J., Díaz, I., Borro, D., and Gil, J.J., "Effective Haptic Rendering Method for Complex Interactions", submitted to Computer Graphics International 2009 (CGI'09). Victoria, British Columbia, Canada. May 26-29, 2009.

Las aplicaciones de renderizado háptico imponen una frecuencia de muestreo elevada (1 kHz) para evitar inestabilidades en el sistema. Sin embargo, esta frecuencia es difícil de alcanzar por los métodos que determinan la respuesta háptica sobre todo si se trabaja en entornos complejos. Una posible solución es reducir el coste computacional del cálculo de la respuesta disminuyendo también la precisión, lo que puede derivar en la aparición de errores o inestabilidades en el sistema.

Es un hecho demostrado que la existencia de inestabilidades en la respuesta háptica provoca una distorsión de la percepción incomodando al usuario y disminuyendo la usabilidad del sistema. Por este motivo, la elección de un método apropiado para garantizar una respuesta agradable

pasa por buscar un compromiso entre la estabilidad y el coste computacional.

Con el objetivo de contrastar el comportamiento de las diferentes alternativas propuestas, se han realizado diferentes experimentos analizando estos dos criterios (continuidad y coste computacional), además de tener también en cuenta los comentarios de usabilidad de los diferentes usuarios. Los experimentos se han realizado sobre un conjunto de modelos que cubren en la medida de lo posible varios aspectos como el tipo de respuesta háptica (fuerzas o pares), la complejidad de los modelos y de las situaciones en colisión (discontinuidades geométricas y múltiples contactos).

Algunas de estas alternativas no son válidas para todos los casos, bien por sus limitaciones con cierto tipo de escenarios como es el caso del método distancia punto plano o bien por su elevado coste computacional. Este es el caso del método del z-buffer cuya frecuencia está limitada a la frecuencia del módulo de visualización tal y como se explicará en el apartado 6.4.3.

En este capítulo se describe en primer lugar la arquitectura del sistema háptico utilizado y posteriormente la metodología a seguir en los experimentos. Al no disponer de un dispositivo háptico completo de 6 gdl, la validación de los métodos propuestos se realiza de forma desacoplada utilizando dos dispositivos diferentes: uno con 3 gdl traslacionales (PHANToM) y otro con 3 gdl actuados en rotación (palanca háptica). De esta forma se consigue validar el cálculo de las fuerzas en situaciones complejas y una vez demostrada su estabilidad, se analiza el problema de los momentos en cuyo cálculo también se utilizan las fuerzas.

Finalmente, se utiliza un sistema de 4 gdl (3 de rotación y 1 de traslación), que aunque no reproduzca un sistema completo de 6 gdl puede ayudar a validar la solución general propuesta. Una descripción más detallada de los dispositivos hápticos utilizados se puede ver en el Anexo A.

6.1 ARQUITECTURA DEL SISTEMA

En este apartado se describe brevemente el sistema completo desarrollado por el departamento de Mecánica Aplicada del CEIT y que se ha utilizado en la presente Tesis.

El sistema desarrollado llamado REVIMA (Desarrollo de una Herramienta de Realidad Virtual para la Simulación de proceso de montaje y

Mantenimiento) es una herramienta de realidad virtual para comprobar el diseño y la mantenibilidad de los motores aeronáuticos, aunque puede ser aplicado a cualquier sistema mecánico (Savall et al. 2002). Esta herramienta está diseñada para trabajar justo entre las etapas de diseño y construcción de la maqueta física, con el objetivo de ayudar a encontrar problemas de mantenibilidad/accesibilidad lo antes posible, y disminuyendo el número de maquetas físicas a construir reduciendo también los costes del proceso.

Otra de las funciones principales de esta aplicación es que puede ser utilizada como herramienta de entrenamiento antes de pasar a operar con motores reales, o como banco de pruebas para el diseño de operaciones de mantenimiento al proporcionar pistas sobre los problemas que pueden aparecer al realizar una de estas operaciones.

Por todo lo que se ha mencionado anteriormente, REVIMA proporciona un entorno adecuado para validar los objetivos de esta Tesis. En la Figura 6.1 se puede observar el sistema totalmente montado y en funcionamiento.

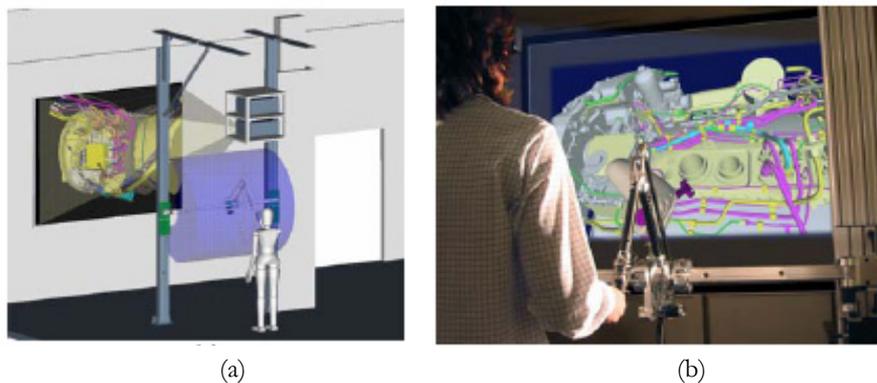


Figura 6.1: Modelo CAD del sistema (a) y el sistema real en funcionamiento (b).

Este sistema multidisciplinar incluye diferentes líneas de investigación: diseño mecánico, teoría de control, gráficos por computador, geometría computacional e interacción persona computador.

De forma general, el sistema tiene dos componentes principales: hardware y software. La componente hardware son los diferentes dispositivos hápticos que pueden conectarse y funcionar con el sistema. Por otro lado, el desarrollo software se divide en 4 módulos: el control del háptico, la visualización, la detección y respuesta de colisiones y la GUI (Graphical User Interface). Esta última se encarga de centralizar todos los

eventos además de ofrecer la interfaz gráfica al usuario. Los módulos software están basados en C++ y en la librería gráfica OpenGL bajo el sistema operativo Windows XP.

En este tipo de sistemas, el cálculo de las colisiones y de la respuesta a dichas colisiones representa el verdadero cuello de botella. Esto unido a la alta frecuencia necesaria por el módulo de Control para garantizar una respuesta estable, hacen que sea necesario separar dichos módulos para que se ejecuten en PCs o procesadores independientes. Trabajos previos (Gregory et al. 2000a, Mark et al. 1996, McNeely et al. 1999, Siira et al. 1996) ya han aplicado este tipo de arquitectura en la que se separa el control del háptico del módulo de simulación (visualización y cálculo de colisiones).

En la Figura 6.2 se puede observar un esquema del sistema global utilizado. Este sistema se ejecuta en un procesador de doble núcleo y una tarjeta de adquisición de datos (dSPACE). Por lo tanto, en total el sistema dispone de tres procesadores diferentes a los cuales se les pueden asignar tareas distintas.

La dSPACE se encarga de ejecutar el bucle o lazo de control a una frecuencia de 1 kHz. Es la encargada de controlar los dispositivos hápticos que se conectan al sistema. Se encarga de la lectura de posiciones del dispositivo y la lectura de información de colisiones con la que podrá calcular la fuerza o par que deben ejercer los motores del dispositivo háptico.

En uno de los procesadores restantes se ejecutan las tareas de visualización y GUI, mientras que el otro está dedicado de forma exclusiva a la detección y respuesta de colisiones. Esto permite tener dos procesos asíncronos en los que el cómputo de uno no tiene influencia en la frecuencia de cálculo del otro. Este PC además posee 2 GB de memoria RAM suficientes para almacenar tanto la geometría de la escena como las estructuras de datos utilizadas por el módulo de colisiones.

Las principales razones por las que se ha elegido esta arquitectura en concreto son:

- El lazo de control necesita ejecutarse con prioridad de tiempo real para conseguir una frecuencia de muestreo de al menos 1 kHz.
- La interfaz háptica permite conectar diferentes dispositivos hápticos a la aplicación.

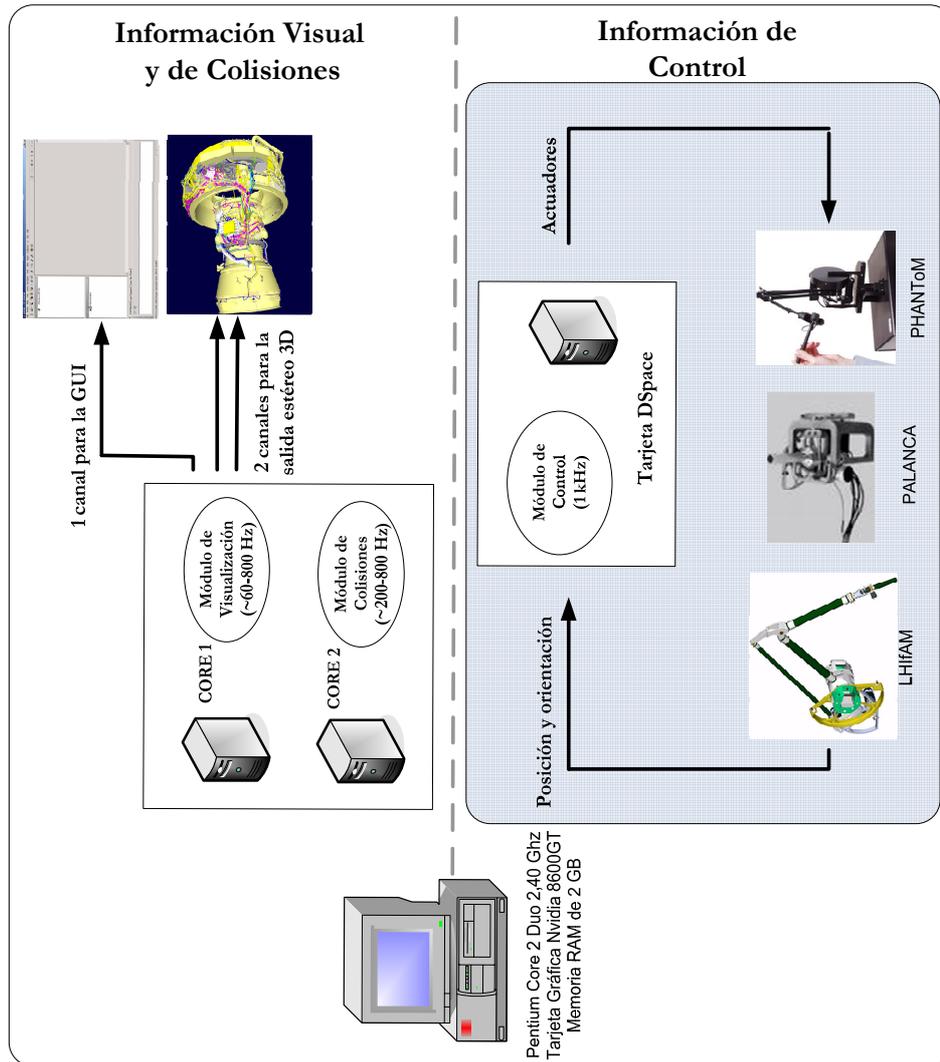
La salida al usuario del sistema se realiza a través de 3 canales de vídeo. Uno muestra la interfaz gráfica gracias a la cual se maneja toda la aplicación, y dos canales más para la salida visual que permite una visualización en estéreo de la escena virtual.

A continuación se describen brevemente los diferentes módulos del sistema y la información que se intercambian cada uno de los procesadores.

6.1.1 DISPOSITIVOS HÁPTICOS

La arquitectura del sistema permite conectar diferentes dispositivos para ser utilizados en cualquier tarea. Concretamente, puede utilizarse un ratón 3D (6 gdl Space Mouse de 3D conexion) que aunque no proporcione una restitución de fuerzas, permite movimientos en todo el entorno.

Además pueden intercambiarse diferentes dispositivos hápticos como los mencionados en el Anexo A: PHANToM Premium 1.0, palanca de 3 gdl rotacionales y LHIFAM.



Orden de ejecución (sin tener en cuenta que la comunicación es asíncrona)

- | | |
|---|---|
| 1 - Módulo de Control: Lectura de los encoders | 5 - Módulo de Colisiones: Detección de colisiones con la nueva posición de la herramienta |
| 2 - Módulo de Control: Conversión a cuaternios | 6 - Módulo de Colisiones: Cálculo de la respuesta de colisión |
| 3 - Módulo de Control: Envío de las coordenadas al Módulo de Visualización | 7 - Módulo de Colisiones: Envío de la información de contacto al Módulo de Control |
| 4 - Módulo de Visualización: Actualización de la imagen gráfica con la nueva posición de la herramienta | 8 - Módulo de Control: Cálculo de la fuerza a restituir por el dispositivo háptico |
| | 9 - Módulo de Control: Restitución de esfuerzos dando los valores correctos a los motores |

Figura 6.2: Esquema de la arquitectura del sistema REVIMA.

6.1.2 MÓDULO DE VISUALIZACIÓN

El módulo visual es el encargado de todo lo relativo a la escena gráfica que se visualiza. Las tarjetas de última generación existentes en el mercado garantizan un *frame rate* interactivo con modelos masivos. Esta frecuencia soporta los requerimientos mínimos que exigen un mínimo de 20 Hz para conseguir una visualización interactiva. A partir de este valor el ojo no es capaz de distinguir la diferencia con frecuencias mayores. El *frame rate* de la aplicación puede incrementarse de diferentes formas:

- Si la maqueta es demasiado densa y pesada para interactuar con ella, se puede activar un pre-proceso de simplificación tal y como se muestra en la Figura 6.3.

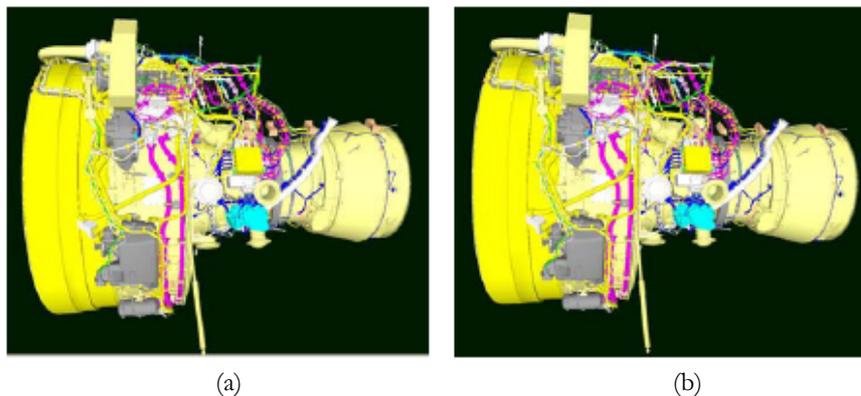


Figura 6.3: Modelo de 1.615.172 polígonos original (a) y simplificado un 40% (b).

- Las tareas de accesibilidad normalmente se realizan en áreas muy específicas de la maqueta. En tiempo de ejecución es posible definir volúmenes de trabajo como aparece en la Figura 6.4. El resto de la maqueta no es visible y ello incrementa el *frame rate* visual.

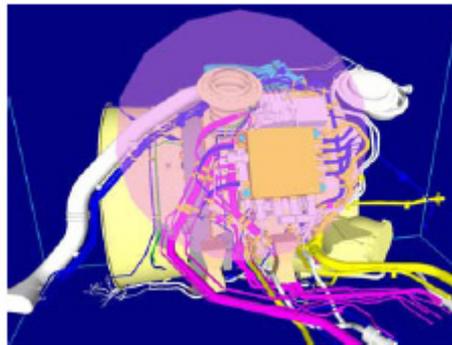


Figura 6.4: Definición de un volumen de trabajo alrededor de un elemento.

- El modulo visual implementa técnicas de visualización orientadas a la optimización del *frame rate* (Amundarain et al. 2003, Mansa et al. 2006): *frustum culling* y *occlusion culling*.

6.1.3 MÓDULO DE DETECCIÓN Y RESPUESTA DE COLISIONES

El módulo de detección y respuesta de colisiones es el encargado de detectar las intersecciones ocurridas en la escena virtual y calcular y enviar la información necesaria al módulo de control para el cálculo de una restitución de fuerzas y/o pares. El estudio de este módulo ha sido el objetivo principal de la presente Tesis.

La frecuencia del módulo de colisiones debería ser lo más parecida a la del módulo de control para evitar inestabilidades en el sistema háptico. Dependiendo del número de triángulos en colisión detectados, los 1000 Hz no están al alcance del cálculo de las colisiones. Por esta razón, el módulo de control tiene que implementar técnicas de interpolación y extrapolación que estimen valores en los periodos de muestreo en los que el módulo de colisión no envía información. El sistema será más estable cuando ambas frecuencias (la de control y la de colisiones) sean lo más parecidas posibles, de forma que el módulo de Control necesite predecir menos valores.

La frecuencia del módulo de colisiones depende en gran medida de la densidad de polígonos de la zona en la que se esté trabajando. Dependiendo de la complejidad geométrica de los modelos utilizados, se podría determinar que el intervalo de frecuencias del módulo de Colisiones en condiciones normales de trabajo opera entre los 200 y 800 Hz. Se ha comprobado que este rango tan amplio de frecuencias no afecta a la

estabilidad de la respuesta háptica gracias al módulo de control y a las técnicas utilizadas en la respuesta de colisión.

Este módulo comparte la memoria con el módulo de visualización para el acceso a la geometría. Una vez realizados sus correspondientes cálculos, la información es enviada al módulo de control para restituir la fuerza y/o par adecuados a cada colisión.

6.1.4 MÓDULO DE CONTROL

El objetivo del control es la restitución de esfuerzos a través del dispositivo háptico usando la información obtenida por el módulo de colisiones. Se han implementado diferentes modelos de contacto: elástico, viscoelástico y viscoelástico no lineal cuyos valores de rigidez (k) y viscosidad (b) se ajustan de forma que mantengan la estabilidad global del sistema.

De forma general, las principales tareas de este módulo son:

- Lectura de los encoders del dispositivo háptico y envío de esa posición y orientación al módulo de visualización. Esta operación se realiza cada periodo de muestreo del lazo de control (1 kHz).
- Recepción de la información sobre las colisiones ocurridas en la escena virtual. Esta información se recibe usualmente a una frecuencia menor que la de control (exactamente a la frecuencia de cálculo del módulo de colisiones) por lo que el módulo de control utiliza técnicas de interpolación y extrapolación para predecir los valores necesarios en ausencia de información por parte del módulo de colisiones (Adachi et al. 1995, Gil 2003, Lin and Gottschalk 1998, Mark et al. 1996).
- Una vez calculada la respuesta adecuada a la colisión, el módulo de control aplica los valores adecuados en los motores del dispositivo para obtener la restitución de esfuerzos.

Además de estas funciones, el módulo de control es el encargado de añadir propiedades a los objetos de la escena como la fricción o asociar una componente de peso diferente a cada objeto teniendo en cuenta su tamaño y material.

6.1.5 COMUNICACIÓN ENTRE MÓDULOS

La arquitectura elegida está centralizada en un único PC por lo que se evita el intercambio de información por la red y la posible pérdida de estos mensajes. La comunicación entre los diferentes módulos del sistema se realiza mediante un intercambio de información que varía según cada módulo.

En cada *frame*, el módulo de control envía al módulo de visualización la información de posición y rotación de la herramienta virtual a partir de los datos obtenidos por los encoders. Esta información es imprescindible para que no se produzca un desmapeo entre la posición virtual del objeto móvil y la posición real que varía con el movimiento del dispositivo háptico.

En caso de detectarse una colisión, es necesario determinar qué tipo de respuesta se va a restituir: fuerza y/o momento. Esto es así, ya que como se ha comentado anteriormente, los experimentos realizados desacoplan ambas respuestas usando en cada caso el dispositivo háptico correspondiente al experimento.

Al restituir una fuerza, la información que se envía al módulo de control es la normal de contacto, penetración y el punto donde se calcula la colisión. Una vez calculada esta información, es el módulo de control el encargado de calcular la fuerza resultante aplicándole el modelo de contacto apropiado.

En el caso de los momentos, el módulo de colisiones es el encargado de calcular el par final a devolver. También es necesario indicarle qué rotación (cuaternio) tiene el objeto en el momento del cálculo de la colisión.

En la Figura 6.5 se representan los diferentes tipos de mensajes que se cruzan los módulos del sistema. Cada mensaje tiene un primer campo que identifica el tipo y con ello la cantidad de información que lleva el mensaje.

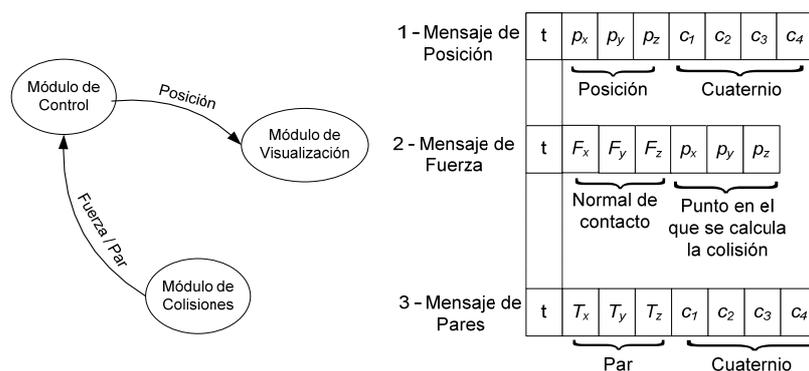


Figura 6.5: Mensajes intercambiados entre los módulos de REVIMA.

6.2 OBJETIVOS Y DESCRIPCIÓN DE LOS EXPERIMENTOS

El principal objetivo de esta Tesis es proporcionar una sensación háptica agradable al usuario, evitando grandes discontinuidades en la respuesta. Este propósito se complica por el hecho de utilizar una geometría que aproxima discretamente a los objetos y que por lo tanto, implícitamente, ya introduce discontinuidades en el sistema.

Otro factor a tener en cuenta en los sistemas hápticos es el tiempo de respuesta. Usualmente los dispositivos hápticos se utilizan en sistemas de Realidad Virtual en los que es necesaria una frecuencia mínima para que el usuario interactúe en tiempo real con el entorno (1 kHz). Sin embargo, en la mayoría de los casos el cálculo de colisiones representa el cuello de botella en este tipo de aplicaciones y su frecuencia es menor. Posibles soluciones a este problema pasan por aceptar una precisión menor o utilizar modelos geométricos más sencillos. Pero en general, este requisito obliga a encontrar un equilibrio entre la calidad de la respuesta calculada y el tiempo consumido.

Por lo tanto, establecer un método de evaluación de diferentes métodos de restitución háptica no es una tarea trivial ya que requiere valorar aspectos muy diferentes: continuidad, tiempo de cálculo y percepción del usuario.

En cuanto al tipo de experimentos diseñados para la validación de los resultados, se han intentado analizar situaciones muy diversas cuyos cálculos implicasen complejidad geométrica por la aparición de discontinuidades o

complejidad computacional por el número de objetos que constituyen la escena.

Como ya se ha visto en el estado del arte del capítulo 2, existen varios trabajos (McNeely et al. 1999, Otaduy 2004) que estudian el problema del cálculo de fuerzas y pares con dispositivos hápticos. Sin embargo, aunque exhaustivos y avalados, estos trabajos no tienen como objetivo el estudio de cómo influyen las diferentes discontinuidades que se pueden presentar en simulaciones interactivas: cambios bruscos de dirección o módulo tanto en fuerzas como en momentos. En este trabajo, se pretende analizar el comportamiento de diversos métodos en situaciones concretas de discontinuidad para posteriormente integrar y aplicar las soluciones a problemas complejos en entornos reales.

Generalmente, para comparar algoritmos se utiliza un path idéntico, es decir, se define una trayectoria del objeto móvil y posteriormente se analiza el comportamiento de los algoritmos usando ese path (path grabado). El path es un conjunto de posiciones discretas consecutivas y próximas entre sí.

Existen diversos modos de definir un path, pero es frecuente usar el propio dispositivo háptico para mover el objeto móvil, almacenando el conjunto de posiciones de dicho movimiento. La ventaja de usar un mismo path para comparar algoritmos reside en que el análisis se realiza en igualdad de condiciones. La desventaja desde el punto de vista de la percepción es que se está haciendo una comparación que no es real ya que al utilizar un mismo path para realizar la comparación con el resto de métodos, se condicionan los resultados porque cada método se comportaría de forma diferente ante las discontinuidades.

Debido a la comentada falta de realismo de los “paths grabados”, en los experimentos que se recogen en la Tesis se da más importancia a los experimentos sin path o “experimentos interactivos”. Se considera experimento interactivo a cada prueba realizada individualmente para cada método sin ningún tipo de movimiento pregrabado, teniendo el usuario total libertad para mover el dispositivo háptico. El problema de este tipo de experimentos reside en que resulta difícil realizar exactamente el mismo movimiento para todas las pruebas y que además las simulaciones tengan las mismas posiciones y duración. Sin embargo, se consideran los más realistas ya que los movimientos del usuario están condicionados por cada método, al contrario que con los paths grabados en los que los movimientos están prefijados de antemano.

6.3 CRITERIOS DE EVALUACIÓN Y FACTORES QUE LOS INFLUENCIAN

Tal y como se ha comentado, no resulta una tarea fácil definir los criterios a utilizar para evaluar el comportamiento de un método de respuesta de colisiones ya que, además de analizar resultados objetivos como la normal y la penetración, entran en juego factores subjetivos que van a condicionar la percepción del usuario en situaciones complejas.

Para cubrir tanto el campo objetivo como el subjetivo, se ha considerado oportuno analizar los siguientes criterios en los experimentos para determinar la bondad de cada método:

- **Variaciones en el cambio de normal.** Las variaciones bruscas en la normal de contacto debidas a errores en su cómputo o a discontinuidades geométricas, pueden provocar discontinuidades en la respuesta háptica que influyen negativamente en la percepción final.
- **Variaciones en la penetración.** Las variaciones bruscas de la magnitud de la penetración en *frames* sucesivos pueden provocar que el usuario perciba discontinuidades en la respuesta provocando una percepción molesta en la aplicación.
- **Variaciones en el par resultante.** Al igual que las variaciones de las normales de contacto y las magnitudes de penetración, las variaciones bruscas en el par resultante pueden provocar inestabilidades y que la respuesta no sea coherente con la esperada, desorientando al usuario. Los dos factores anteriores (normal de contacto y penetración) tienen también influencia en las variaciones que se produzcan en el momento resultante, ya que con ellos se calcula la fuerza final a restituir utilizada en el cómputo del momento final.
- **Coste computacional.** Debe ser mínimo para garantizar una interacción realista con el entorno. El caso ideal sería realizar la detección de las colisiones y de la respuesta en un tiempo máximo de 1 ms para que la frecuencia del módulo de colisiones fuese igual a la del módulo de control. Con el hardware actual, esta frecuencia es difícil de conseguir ya que depende de la complejidad geométrica de los modelos utilizados.

- **Percepción subjetiva.** Además de los resultados numéricos, la percepción que tiene el usuario que utiliza el sistema es realmente importante a la hora de validar la respuesta calculada. De hecho, la predisposición del usuario y aspectos psicológicos influyen en la percepción del entorno incrementando la inmersión en estos sistemas. Por este motivo, se tienen en cuenta los comentarios de los usuarios. Sin embargo, un estudio completo sobre la percepción de la respuesta háptica recibida queda fuera del alcance de esta Tesis.

Estos criterios utilizados para evaluar la calidad de la respuesta calculada, se ven influenciados por los siguientes parámetros:

- **Rigidez.** En el mundo real, el usuario siente un valor de rigidez infinito cuando colisiona con un objeto rígido que no le permite penetrar en dicho objeto. En un entorno virtual, resulta imposible reproducir estos valores de rigidez tan altos ya que provocan inestabilidades en el sistema. Esta Tesis no aborda el estudio de este problema. En los experimentos se utiliza un valor de rigidez que no permita una alta penetración de los objetos pero que garantice la estabilidad al sistema.
- **Teselación.** El cálculo de la penetración puede verse afectado por el nivel de teselación de cada uno de los modelos. Con teselaciones más burdas, el cálculo de la magnitud de la penetración será menos preciso y en cambio, con mallados más finos la precisión en el cálculo será mayor aunque el coste computacional también se verá incrementado. Esta Tesis no aborda el problema de mallados óptimos.

El tamaño de celda o nivel de voxelizado podría considerarse otro parámetro que influiría sobre todo en el criterio de coste computacional. Sin embargo, como ya ha sido explicado en la Sección 3.2, previo a la simulación, la detección de colisiones calcula el voxelizado que hace mínimo el coste computacional para ese entorno virtual en concreto.

Para llevar a cabo un estudio exhaustivo de cada método propuesto, se analiza cada uno de los criterios expuestos tanto desde un punto de vista objetivo, evaluando su comportamiento a lo largo del tiempo, como desde un punto de vista subjetivo en el que el usuario valore si la respuesta que recibe tras una colisión es suave y estable. En los siguientes apartados, se

describen los experimentos diseñados y se muestran sus resultados, primero para el caso de fuerzas y posteriormente para el cálculo de momentos.

6.4 EXPERIMENTOS CON FUERZAS

En este apartado se realizan una serie de experimentos para contrastar las diferentes soluciones propuestas y determinar cuál aporta una respuesta más estable. El análisis se realiza en situaciones complejas con discontinuidades geométricas, ya que un cálculo de fuerzas apropiado en estas situaciones es un buen punto de partida para garantizar también la estabilidad de los momentos, en cuyo cálculo se utilizan dichas fuerzas.

El desacoplamiento en el estudio de la respuesta de fuerzas y pares permite también validar la solución propuesta para ser aplicada en sistemas hápticos que siguen utilizando dispositivos hápticos de 3 gdl actuados bien porque no necesitan gran precisión en la respuesta o por el elevado coste de los dispositivos de 6 gdl actuados.

Para comprobar la estabilidad final de la fuerza, el estudio se centra en los dos factores que condicionan la fuerza resultante: la dirección y su magnitud. Los modelos utilizados representan situaciones complejas de colisión tales como discontinuidades geométricas o contactos múltiples. La complejidad de estos casos se debe principalmente a que se producen cambios bruscos de dirección o magnitud que provocan inestabilidades en la fuerza resultante. Este tipo de situaciones son comunes en entornos reales como el de la Figura 6.6, por lo que es necesario proporcionar una solución a estos casos para garantizar una interacción agradable con el sistema. Para evitar que el análisis de cada caso se vea afectado por múltiples factores, las pruebas se han realizado utilizando modelos sencillos que reproducen de forma aislada cada situación a analizar (Tabla 6.1).

Concretamente, se analizan cuatro tipos de discontinuidades en las que el cálculo de la respuesta háptica puede presentar problemas de estabilidad: una esquina convexa, esquina cóncava, un contacto doble sin discontinuidades geométricas (contacto doble con superficies redondeadas) y finalmente, un contacto doble que además presenta discontinuidades geométricas (contacto doble con esquinas). Estos dos últimos casos se referencian en la Figura 6.6 como doble contacto tipo 1 y doble contacto tipo 2 respectivamente. En el resto del apartado se hará referencia a estos casos mediante esta nomenclatura.

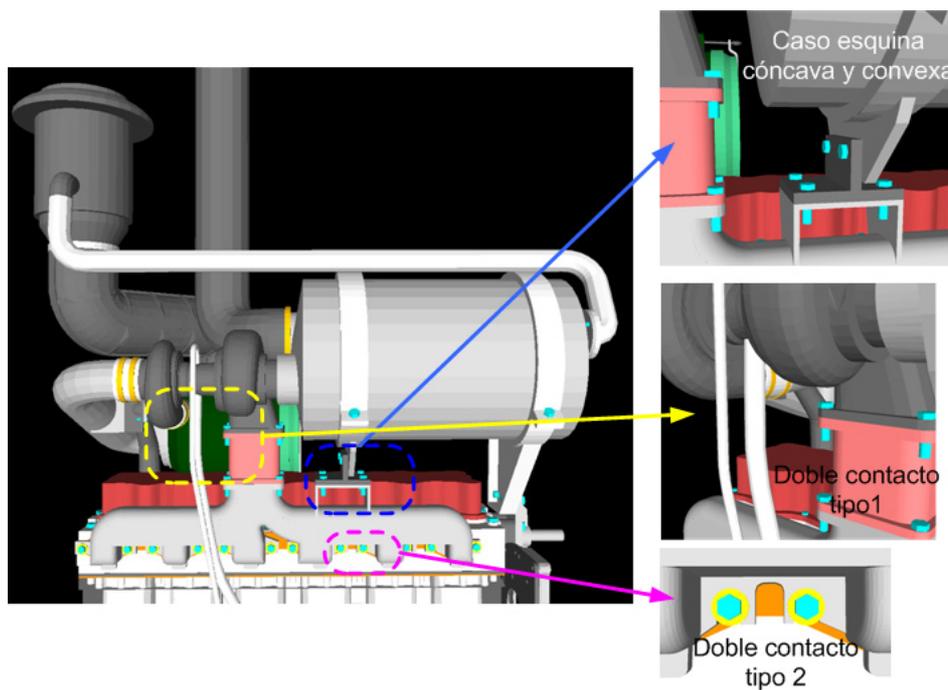


Figura 6.6: Entorno real en el que mediante un zoom se muestran zonas con discontinuidades geométricas y múltiples contactos.

Modelo	Nº de triángulos	Figura
Esquina cóncava/convexa	60.000 triángulos	
Doble contacto tipo 1	20.000 triángulos	
Doble contacto tipo 1	50.000 triángulos	

Tabla 6.1: Modelos estáticos utilizados en las pruebas de fuerzas.

Para todas las pruebas se ha utilizado una esfera definida con 1500 triángulos como objeto móvil y el dispositivo háptico utilizado es el PHANToM Premium 1.0 (véase Anexo A.1).

6.4.1 PATH GRABADO

En esta sección se ha grabado un path con un método en concreto, para posteriormente replicar este path con el resto de métodos.

En este ejemplo se estudia el comportamiento de cada uno de las propuestas en el caso de un doble contacto de tipo 2 (Tabla 6.1).

El path utilizado se graba con el método de distancia punto plano y la discontinuidad se ha recorrido en dos ocasiones. El estudio de la variación de la normal se analizará en el siguiente apartado cuando se trate de nuevo este ejemplo utilizando experimentos interactivos.

La Figura 6.7, si bien contiene todos los valores de la penetración calculados a lo largo del path, estos datos no son representativos ya que una penetración elevada no conlleva una inestabilidad si se ha llegado a ese valor de forma gradual. Para ello, se ha optado por analizar la penetración con otras gráficas donde se representa para cada método la variación de este valor (delta de penetración entre el *frame* actual y el anterior).

Este delta multiplicado por el valor de rigidez utilizado permite determinar la variación producida en la fuerza resultante. En el método de distancia punto-plano, las gráficas muestran variaciones a veces cercanas a 4 mm que equivaldrían a cambios en el valor de la fuerza de 4 N siendo $k=1000$ N/mm. De un instante a otro se estaría actuando el PHANTOM muy por encima de la máxima fuerza (en continuo) que el dispositivo es capaz de ejercer, provocando que el usuario sienta fuertes vibraciones.

Observando las variaciones de los valores de penetración, se pueden diferenciar aquellas posiciones en las que el objeto móvil atraviesa la discontinuidad. A la vista de los resultados de las variaciones de cada una de las propuestas, se puede afirmar que estas variaciones son menores utilizando el método de rayos.

Aunque los resultados obtenidos son satisfactorios, este experimento se considera insuficiente para asegurar que el método basado en rayos es el más estable, por las razones anteriormente comentadas relacionadas con la falta de realismo al utilizar un path grabado. Por este motivo, a partir de ahora se realiza el estudio de todos los casos como experimento interactivo.

También se muestran el número de triángulos en colisión y el coste computacional asociado a cada método que como media es próximo al periodo de muestreo ideal de 1 ms. Estos datos son muy similares para el resto de los casos de fuerzas analizados, por lo que en sucesivas gráficas

serán omitidos para centrar la atención en las variaciones de dirección y magnitud de las fuerzas calculadas.

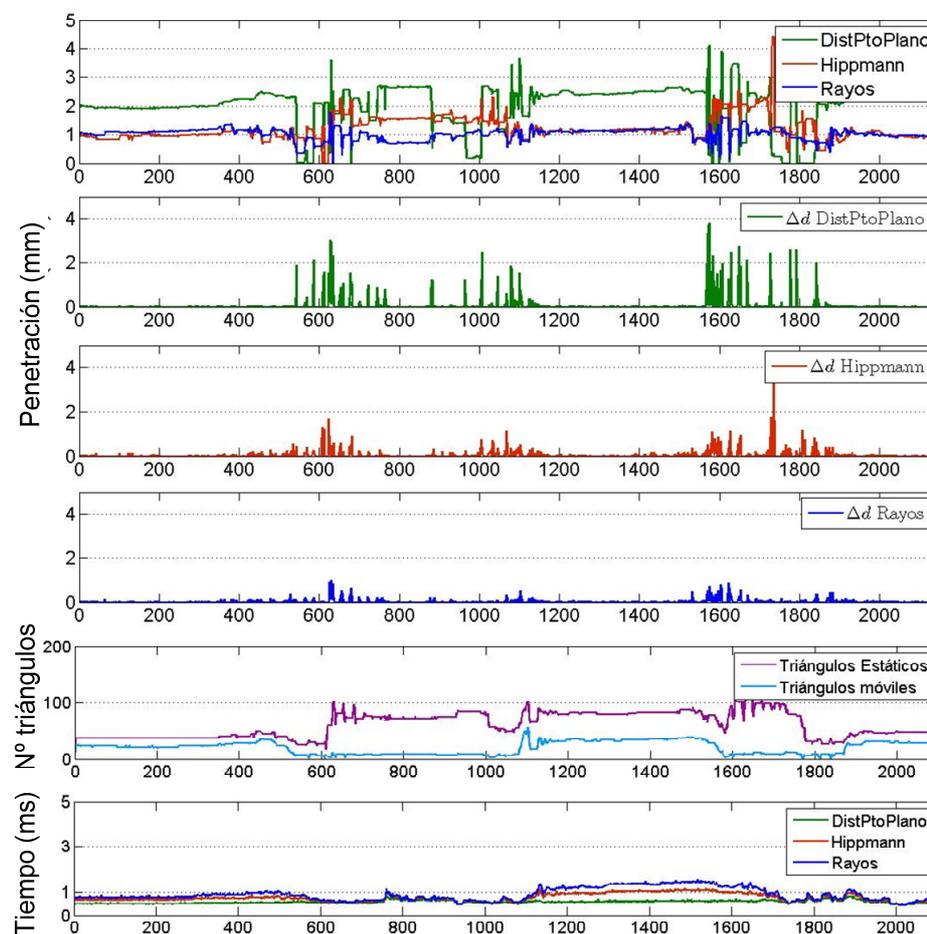


Figura 6.7: Doble contacto tipo 2: Penetraciones obtenidas en la simulación con los diferentes métodos y las variaciones que se producen en sus magnitudes.

6.4.2 EXPERIMENTOS INTERACTIVOS

El primer experimento consiste en simular el recorrido de una esfera por una esquina convexa. En este tipo de discontinuidad, la normal de la superficie cambia su dirección de forma brusca, concretamente 90° . Este cambio puede provocar en el usuario una sensación incómoda que influye negativamente en la percepción global del sistema. Para evitarlo, la solución propuesta realiza un cambio de normal progresivo (Figura 6.8). A medida

que la esfera se va acercando a la esquina, se detecta que en la colisión están implicadas dos zonas de contacto, y la dirección de la fuerza comienza a variar de forma progresiva tal y como se ha explicado en el capítulo 4. Esta solución permite al usuario recorrer el borde de la esquina de forma suave, sintiendo la discontinuidad existente en la geometría pero evitando que el dispositivo le obligue a abandonar el contacto de forma violenta. En la figura, las posiciones de la esfera son orientativas para dar una idea del movimiento, no son posiciones reales ni exactas.

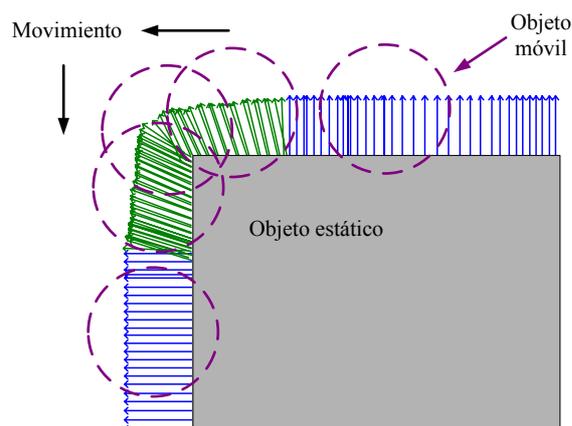


Figura 6.8: Variación de la normal a medida que la esfera se acerca a la esquina.

Además de la variación de la dirección, también se analizan las penetraciones calculadas por cada método a lo largo de cada interacción. Como este valor no es significativo, se analizan las variaciones de las penetraciones (Figura 6.9). En cuanto a estos valores, se observa que utilizando el método de Hippmann, estas variaciones son mayores (de hasta 4 mm) y más numerosas, indicando que este método tiene un comportamiento más inestable que el resto. En cuanto a los otros dos métodos, su comportamiento es similar a la vista de los resultados.

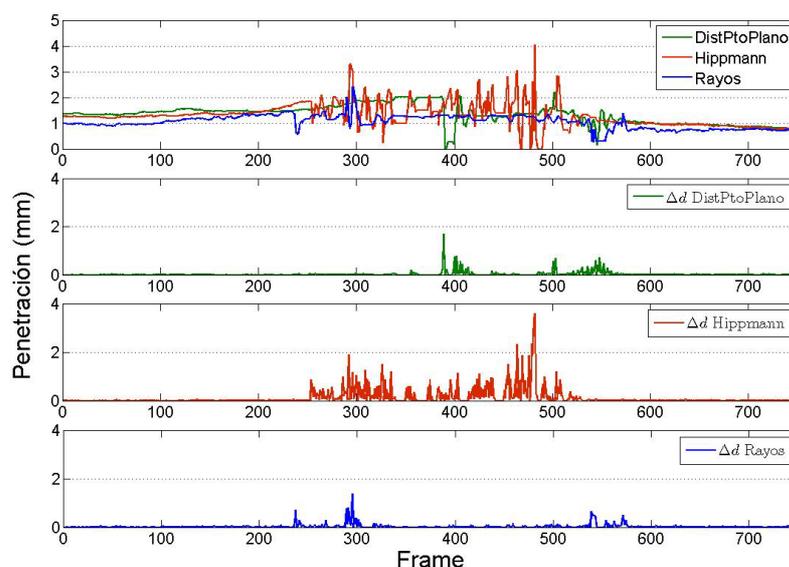


Figura 6.9: Esquina convexa: Penetraciones y variaciones de la penetración a lo largo de la simulación para cada método.

Los ejemplos restantes reproducen casos con colisiones múltiples. La dificultad en estos casos reside en proporcionar una respuesta estable utilizando un dispositivo de fuerzas. Para conseguirlo, es necesario que los cambios en la dirección no sean muy bruscos y que la variación del módulo de la fuerza resultante no sea muy elevada cuando la segunda área de contacto entra a formar parte de la colisión. En caso contrario, pueden producirse rebotes e inestabilidades en la respuesta háptica provocando en el usuario una sensación desagradable.

El segundo experimento consiste en recorrer una esquina cóncava (Figura 6.10), en la que se observa que cuando la esfera entra en contacto con las dos áreas de contacto la normal varía 45° . La variación progresiva de la normal calculada permite al usuario poder mantener la esfera en una posición con contacto múltiple, a la que se le denominará posición de equilibrio en los sucesivos ejemplos.

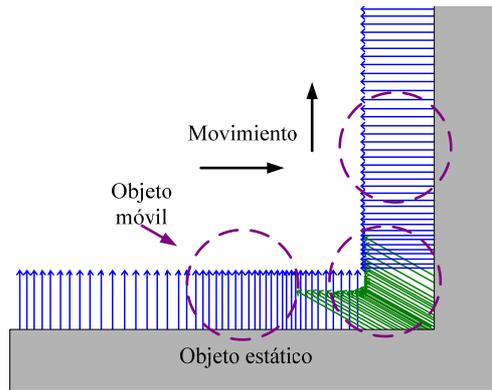


Figura 6.10: Esquina cóncava: Variación de la normal cuando se recorre la esquina cóncava.

La Figura 6.11 muestra las penetraciones obtenidas así como las variaciones de penetración que se producen entre posiciones consecutivas para cada uno de los métodos. La zona sombreada representa las posiciones en las que se produce la colisión con dos áreas de contacto. El movimiento evoluciona como indican la Figura 6.10 y una vez completado vuelve en sentido contrario pasando dos veces por la zona de discontinuidad.

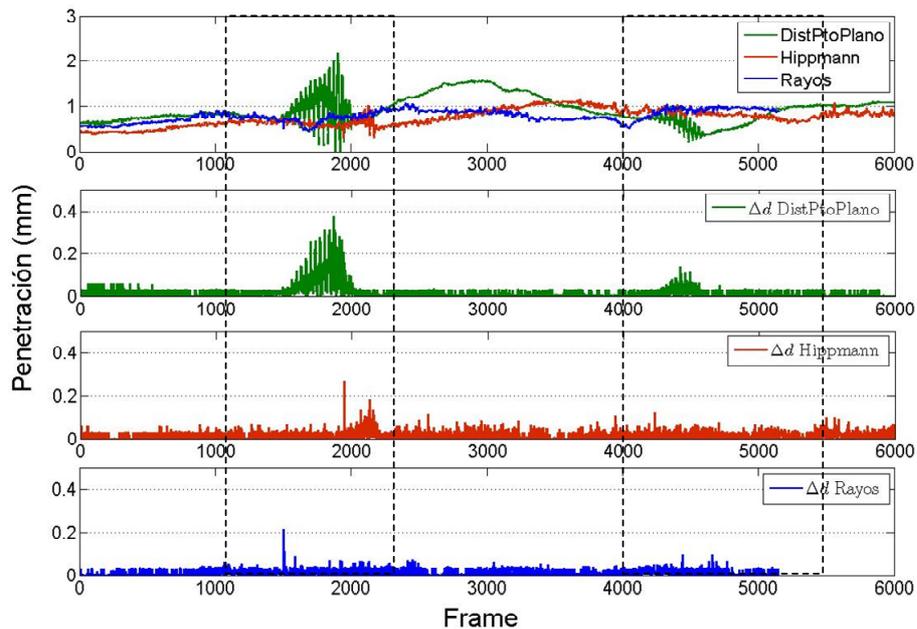


Figura 6.11: Esquina cóncava. Penetraciones obtenidas y variaciones entre *frames* consecutivos.

En base a estos resultados, se concluye que en este caso los resultados obtenidos con el método de Hippmann o con el método de rayos son muy similares mientras que en el caso del método distancia punto plano, las variaciones en la magnitud de la fuerza son más notorias. Estas variaciones son debidas al efecto rebote que se produce al interactuar con el dispositivo háptico.

El tercer experimento simula el recorrido de una esfera sobre un hueco definido por superficies redondeadas (doble contacto tipo 1) que podrían representar dos tuberías. En este caso, la normal resultante debe ir variando progresivamente en función de la geometría en contacto tal y como se observa en la Figura 6.12. Cuando la esfera entra en contacto con dos áreas simultáneamente, la normal resultante es totalmente vertical permitiendo al usuario llegar a una posición de equilibrio al mantener la esfera en dichas posiciones de forma estable.

El movimiento se realiza de izquierda a derecha y se vuelve de nuevo a la posición inicial con un movimiento de derecha a izquierda. En este ejemplo, la dirección de la fuerza resultante varía constantemente debido a la geometría curva que define el modelo estático. Tal y como se ha explicado anteriormente, el cálculo de la penetración se realiza en base a la dirección resultante, por lo que pequeñas variaciones en la dirección provocan también variaciones en su módulo.

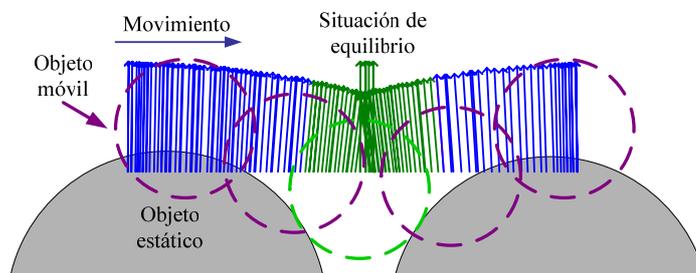


Figura 6.12: Doble contacto tipo 1: Normales obtenidas a lo largo de la simulación en superficies redondeadas.

Los resultados obtenidos en la simulación (Figura 6.13) muestran que las variaciones en la magnitud de la fuerza son mayores utilizando el método distancia punto plano, ya que con este método se aproxima cualquier superficie mediante un plano con la disminución de precisión que eso

implica. Los otros dos métodos obtienen una mejora notable aún más significativa con el método de rayos.

Las zonas sombreadas de la figura representan las posiciones en las que se produce la colisión múltiple. Para asegurar que la esfera tenga un movimiento suave en dichas posiciones, las variaciones de la penetración no deben ser muy elevadas lo que se consigue con los dos últimos métodos.

Al igual que ocurre en el ejemplo anterior, la progresiva variación de la normal unida a las variaciones poco pronunciadas en la magnitud de la penetración permiten obtener posiciones de equilibrio cuando la esfera entra en contacto con dos áreas simultáneamente permitiendo al usuario mantener estable el objeto móvil en dichas posiciones.

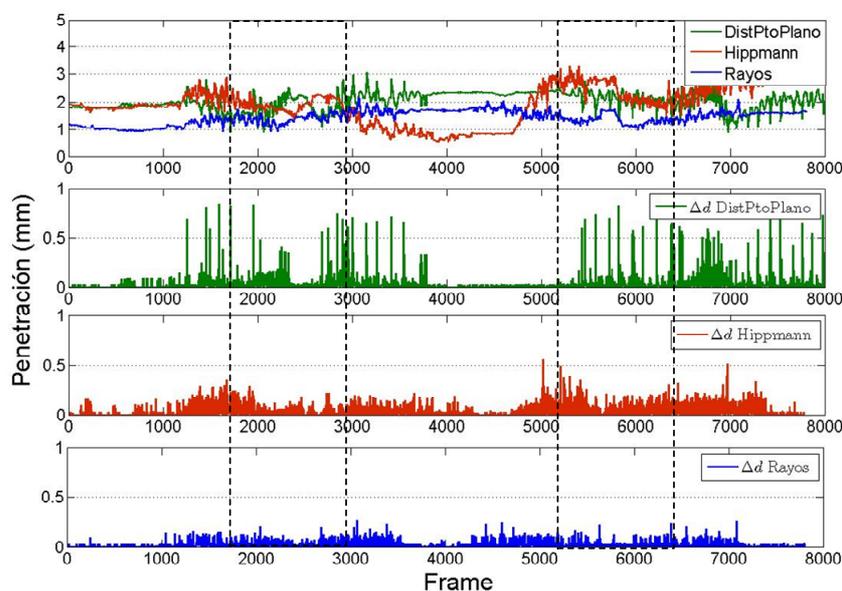


Figura 6.13: Doble contacto tipo 1: Penetraciones y variaciones del módulo de penetración obtenidas a lo largo de la simulación para cada método.

Finalmente, se analiza un ejemplo que combina un contacto múltiple con discontinuidades geométricas (Figura 6.14), simulando el paso de una herramienta por un hueco rodeado de esquinas. El movimiento seguido por la herramienta es igual que en el caso anterior, de izquierda a derecha y de derecha a izquierda, realizando dos pasadas por la discontinuidad a estudiar.

Los resultados obtenidos muestran que la normal cambia progresivamente a medida que se va aproximando al borde evitando así un

cambio brusco en su dirección. Al igual que en el caso anterior, cuando se produce la colisión con múltiples áreas, el cambio de dirección de la fuerza se realiza de forma progresiva proporcionando una respuesta suave y continua a pesar de las discontinuidades existentes.

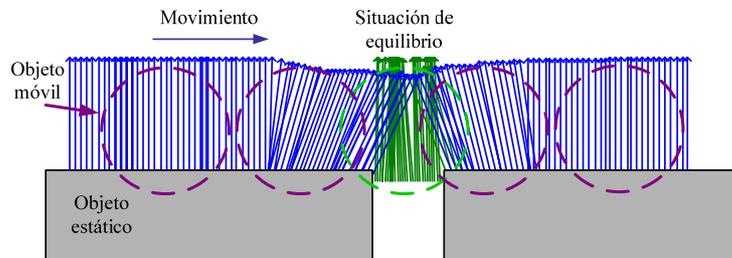


Figura 6.14: Doble contacto tipo 2: Variación de las normales en un hueco con discontinuidades geométricas.

La Figura 6.15 muestra la penetración y las deltas de dicha magnitud obtenidas a lo largo de la interacción.

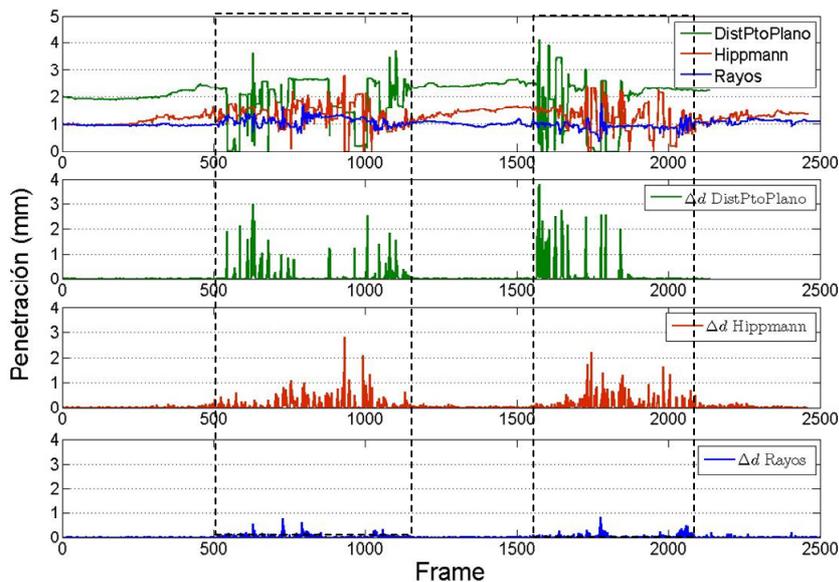


Figura 6.15: Doble contacto tipo 2: Penetraciones y variaciones del módulo de penetración obtenidas a lo largo de la simulación para cada método.

Los resultados confirman que el método basado en rayos ofrece una mayor estabilidad aún con la presencia de múltiples contactos y discontinuidades geométricas, ya que se obtienen variaciones

significativamente menores que en el resto de los casos. Las variaciones de 3 mm obtenidas con el resto de métodos implican cambios muy bruscos en la fuerza, afectando negativamente en la percepción final del usuario.

Con estos resultados se puede concluir que, en general, el método basado en rayos obtiene una respuesta más estable en todas las situaciones complejas como las que se han estudiado.

6.4.3 CONSIDERACIONES ESPECIALES AL MÉTODO DEL Z-BUFFER

El método de respuesta de colisiones ideal es aquel que calcula una fuerza de colisión adecuada a una velocidad cercana al periodo de muestreo del módulo de control (1 ms). Ambos objetivos, una buena señal de fuerza y un cálculo rápido de la misma son contrapuestos lo que implica la necesidad de llegar a un compromiso entre estas dos variables.

Como ya se ha comentado en el apartado 5.5, el uso del hardware gráfico para el cálculo de la penetración obtiene buenos resultados a nivel de precisión, pero por el contrario es dependiente del *frame rate* del módulo de visualización. Se ha realizado un experimento para comparar el comportamiento de este método con el método basado en rayos que consiste en una colisión entre una esfera y una pared (Figura 6.16). Para ello se han analizado los valores de penetración obtenidos por cada método recogidos a la frecuencia del módulo de control (1 kHz). Se observa en las figuras que existen repeticiones en los valores de penetración que reflejan la frecuencia a la cual el módulo de colisiones está calculando la penetración. Hay tantos valores repetidos como periodos de muestreo del control sin información de colisiones.

Comparando ambas figuras entre sí, se observa que la variación de la señal de penetración es más suave con el método de rayos. Esto principalmente es debido a que el coste computacional de dicho método es menor. Por el contrario, el elevado coste computacional del método del z-buffer provoca saltos en la señal final de penetración recibida por el módulo de control.

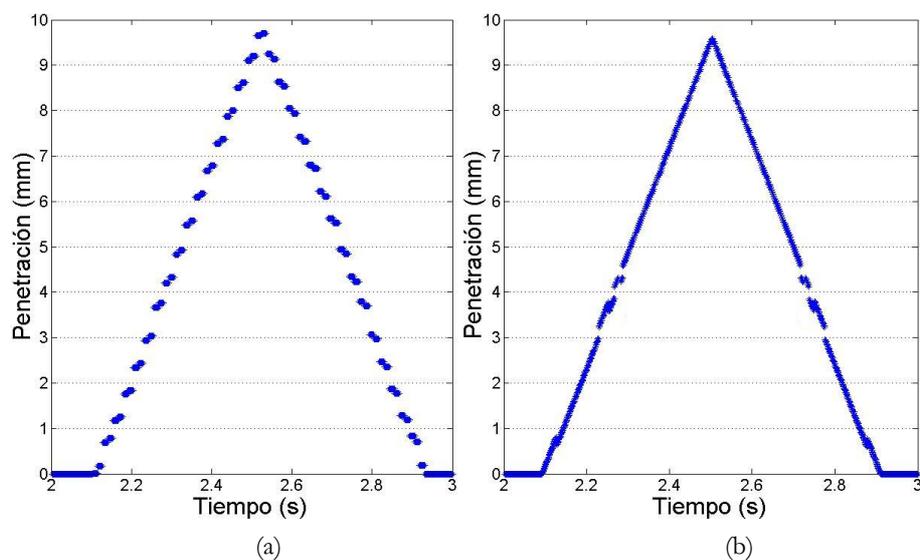


Figura 6.16: Penetración obtenida con el z-buffer (a) y con el método de rayos (b).

El retraso existente entre la información calculada por el módulo de colisiones y la frecuencia del módulo de control (1 kHz) provoca que durante varios periodos de muestreo el módulo de control carezca de información de colisiones suficiente para controlar el dispositivo háptico y cuando la obtiene, ésta difiere de la última información recibida. Este efecto puede observarse en la Figura 6.17, que es un zoom de la Figura 6.16.

Analizando el número de mensajes que se repiten en cada ciclo, se observa que el método que utiliza el z-buffer tiene un coste computacional de 13 ms de media mientras que el método basado en rayos realiza los cálculos en 2 ms (tiempo transcurrido entre mensaje y mensaje de colisión). Estos retrasos con respecto a la frecuencia del módulo de control provocan variaciones en los valores de penetración, que en el caso del z-buffer pueden llegar hasta los 0,5 mm en 1 ms. Concretamente, dicha variación se traduce en una variación de fuerza de aproximadamente 0,5 N, un 33 % de la máxima fuerza (continua) que puede restituir el dispositivo PHANTOM, lo que se traduce en un golpe para el usuario. En el caso del método de rayos, este salto es de aproximadamente 0,05 mm lo que permite obtener una respuesta más continua mejorando también la percepción final.

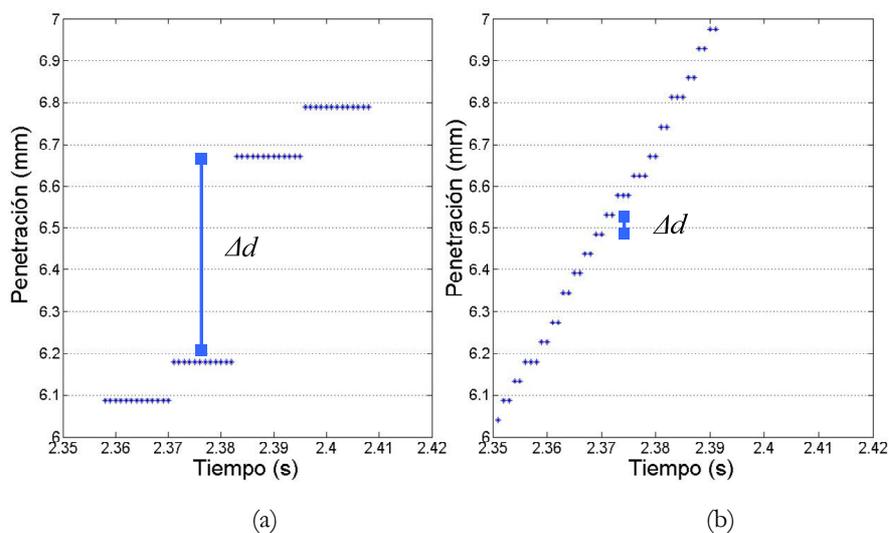


Figura 6.17: Zoom de las penetraciones obtenidas con el z-buffer (a) y con el método de rayos (b).

Para suavizar estos cambios bruscos, el módulo de control puede utilizar filtros o técnicas de interpolación. Sin embargo, cuanto mayor sea el retraso de la información de colisión, la señal original debe modificarse en mayor grado y como consecuencia esta señal acaba siendo menos parecida a la original. Además, se acentúa el retraso entre la percepción visual y el movimiento cinestésico del usuario, y la percepción táctil del evento de contacto.

Por último, es necesario destacar que el ejemplo de colisión utilizado en este apartado es un contacto sencillo con una pared rígida. En colisiones más complejas, además de las variaciones en la penetración, se producen variaciones en la dirección de la normal de contacto acentuando aún más si cabe la problemática de enviar señales con un retraso alto.

En general, el método z-buffer obtiene penetraciones más precisas que el resto de métodos pero a costa de introducir un retraso mayor en el cálculo de la fuerza a restituir. Este incremento en el coste computacional provoca también un retraso en el intercambio de información al módulo de control provocando que la información recogida sea más brusca y menos efectiva y por lo tanto influya negativamente en la percepción final de la respuesta háptica. Por eso, un método como el de rayos con un mejor

compromiso entre precisión y velocidad de cálculo obtiene mejores resultados.

6.5 EXPERIMENTOS CON PARES

Los criterios utilizados para evaluar los resultados obtenidos con el dispositivo de pares son diferentes a los usados con el dispositivo de fuerzas (normal y penetración). En este caso, el estudio se centra en la continuidad del par resultante que engloba el comportamiento individual de las diferentes áreas de contacto que se ven implicadas en la colisión. Hay que tener en cuenta que se analizan situaciones de colisión que implican hasta cuatro áreas de contacto simultáneamente lo cual multiplicaría el número de gráficas (cuatro de penetraciones para cada método analizado) lo que haría muy difícil el análisis de resultados.

La elección del escenario en el que se ha estudiado el cálculo de momentos viene dada porque gran parte de tareas de accesibilidad y mantenimiento que se pueden realizar en modelos industriales, se corresponden con movimientos de abrazaderas a través de tubos (ver ejemplo real en el apartado 6.6) o contactos con múltiples objetos. Se han utilizado dos modelos estáticos formados por tres objetos verticales mientras que el objeto móvil viene representado por una abrazadera definida por 1500 triángulos. El primer escenario (Figura 6.18a) consta de 6.000 triángulos mientras que el segundo (Figura 6.18b) de 25.000 triángulos. Los tres problemas que se van a estudiar en este escenario son:

- Doble contacto sólo con la abrazadera: en este caso se gira la abrazadera para que colisione con el tubo central. Normalmente, esta colisión vendrá dada por dos zonas de contacto.
- Doble contacto de las aletas: en este ejemplo se gira la abrazadera únicamente en el grado de libertad del roll consiguiendo una doble colisión de las aletas de la abrazadera con los objetos laterales.
- Contacto en cuatro áreas: este caso combina los dos anteriores resultando en una colisión múltiple de cuatro áreas.

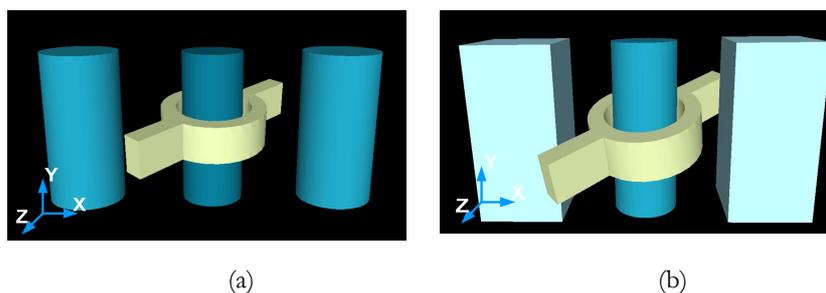


Figura 6.18: Escenario teórico para testear el cálculo de los momentos: objetos laterales cilíndricos (a) y objetos laterales paralelepípedos (b).

6.5.1 PATH GRABADO

El path utilizado en este experimento consiste en tres giros de la abrazadera colisionando en todo momento con el tubo central y evitando la colisión de las aletas con los objetos laterales. Para esta prueba, en primer lugar se ha grabado un path utilizando uno de los métodos analizados, en este caso el método de Hippmann, y posteriormente se ha utilizado este mismo path para calcular el momento generado por cada método en cada una de las posiciones grabadas. Por lo tanto, la respuesta háptica obtenida por cada método no se ha obtenido en una sesión interactiva.

El par obtenido en cada una de las posiciones del path se representa mediante puntos ya que se considera que proporciona una mejor visión de las posibles discontinuidades existentes en cada uno de los métodos y de los intervalos sin información (Figura 6.19). En estos resultados no se han incluido los datos que se obtendrían con el método distancia punto plano por tratarse de un modelo en el que dicho método no obtiene buenos resultados tal y como se ha justificado en el apartado 5.2.

Los resultados obtenidos muestran que el comportamiento del método del z-buffer y el método basado en rayos es similar y no se aprecian grandes discontinuidades. Por el contrario, los resultados obtenidos con el método de Hippmann presentan mayores variaciones en la magnitud del par entre posiciones consecutivas. Como puede observarse, la componente T_y del par resultante es prácticamente nula por el tipo de movimiento analizado.

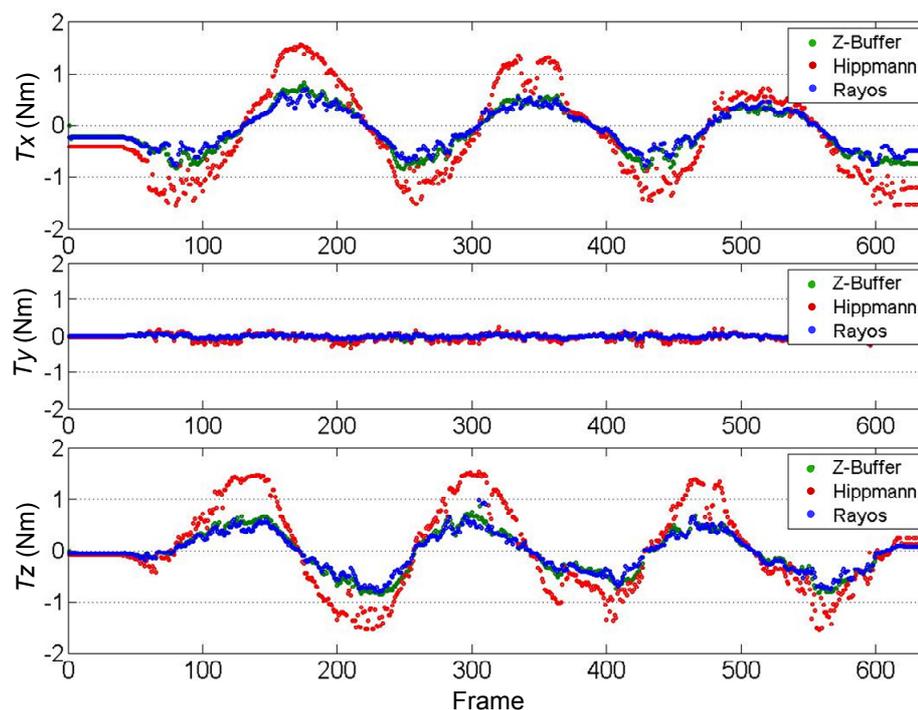


Figura 6.19: Par resultante en cada uno de los ejes (T_x , T_y y T_z).

Para ver con más detalle dichas gráficas, la Figura 6.20 muestra un zoom de un rango determinado de posiciones. Las zonas señaladas indican aquellas zonas en las que la variación del par es más significativa, provocando que se sientan discontinuidades en la respuesta háptica recibida. En estas zonas se puede observar que el método de Hippmann muestra variaciones en la magnitud del par de hasta 0,5 Nm entre instantes de tiempo consecutivos. En el caso de la palanca háptica empleada, el par máximo continuo restituible al usuario es de 0,7 Nm por lo que esta variación en la magnitud del par supone un incremento del 71 % del par total restituible en tan sólo un instante de tiempo. Este cambio de magnitud supone un salto muy brusco que es percibido por el usuario como vibraciones o golpes en el dispositivo y que degradan la percepción final de la colisión. La Figura 6.21 muestra las variaciones producidas en T_x a lo largo de la simulación para una mejor apreciación del comportamiento de cada método.

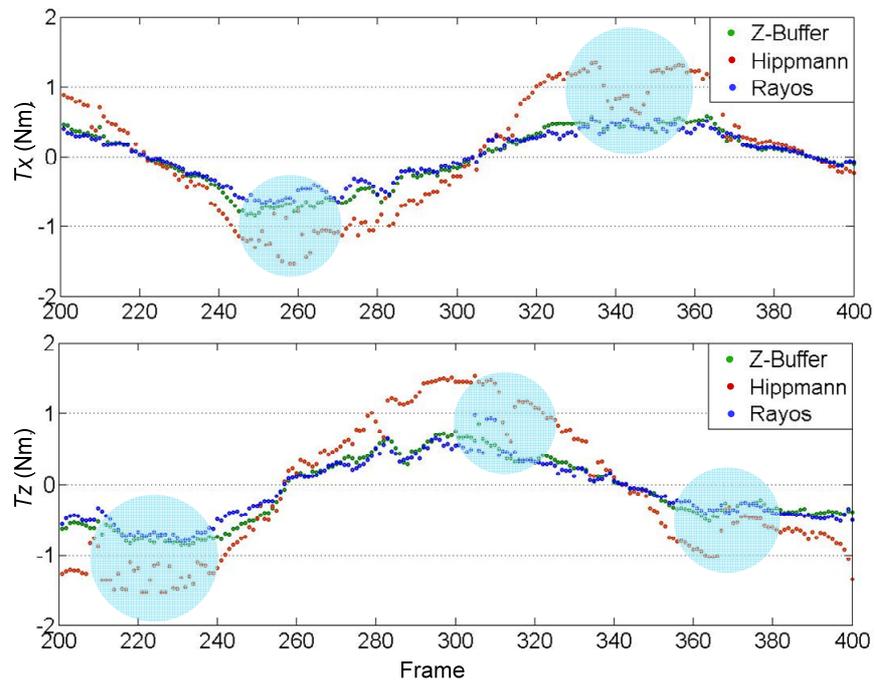
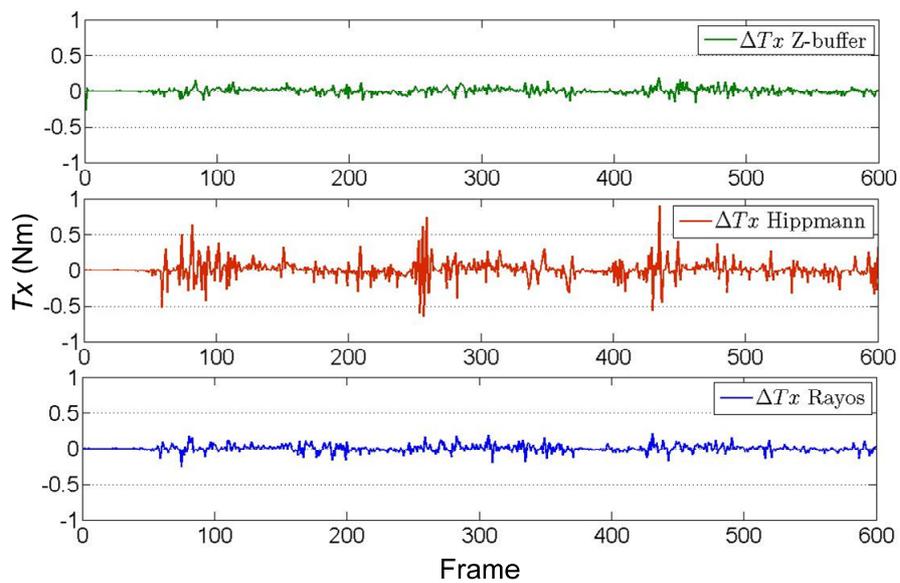


Figura 6.20: Zoom de la zona con discontinuidad.

Figura 6.21: Variaciones en T_x a lo largo de la simulación

Por último, se analiza el coste computacional de cada uno de los métodos (Figura 6.22) y el número de triángulos en contacto implicados en la colisión (Figura 6.23).

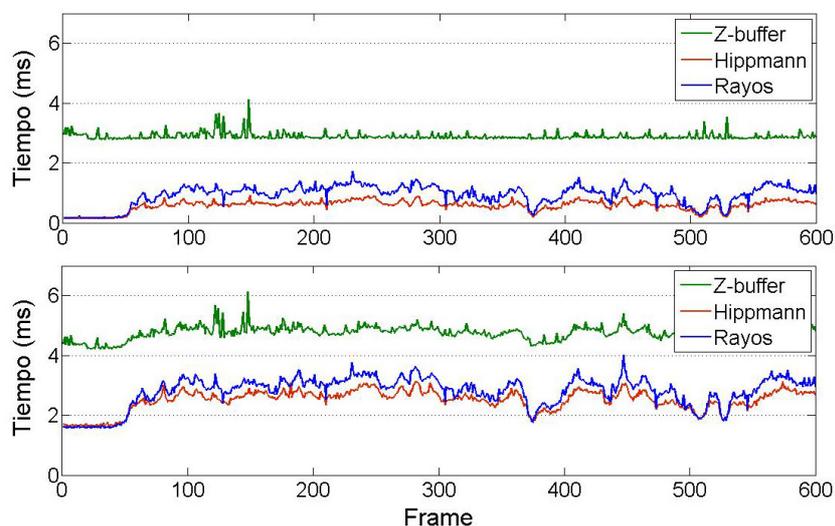


Figura 6.22: Tiempo de respuesta de colisión y coste computacional final (tiempo de detección de colisiones + tiempo de respuesta)

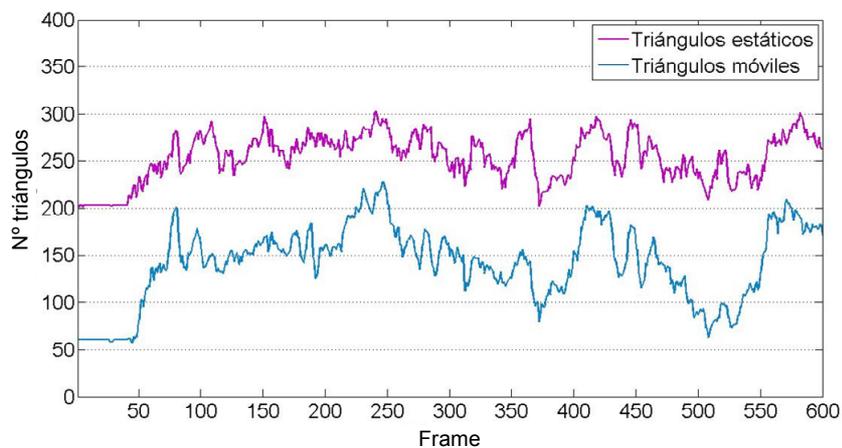


Figura 6.23: Número de triángulos en colisión común para todos los métodos.

El tiempo de respuesta es inferior en el caso del método de Hippmann que con el resto de los métodos analizados, aunque tal y como se ha comprobado no garantiza un mejor comportamiento ya que las

variaciones en el par calculado son mayores. Con respecto a los otros dos métodos, se puede concluir que el método de rayos consigue alcanzar el compromiso entre una respuesta continua y su coste computacional asociado.

6.5.2 EXPERIMENTOS INTERACTIVOS

Una vez descartado el método basado en el z-buffer por las razones comentadas en el apartado 6.4.3, en este apartado los experimentos se han llevado a cabo con los métodos de Hippmann y rayos.

Los experimentos se han realizado para varios usuarios y los resultados han sido coherentes en todos ellos por lo que únicamente se mostrarán los resultados de uno de los usuarios.

Como primera prueba, se ha querido realizar el mismo movimiento que en el apartado anterior girando la abrazadera en tres ocasiones sobre el tubo central. De esta forma, se quiere demostrar que el comportamiento del método de rayos en una interacción real presenta mejores resultados que el método de Hippmann tal y como se observa en la Figura 6.24. De hecho, este último método presenta variaciones en el par que alcanzan los 0.5 Nm provocando inestabilidades en la respuesta final.

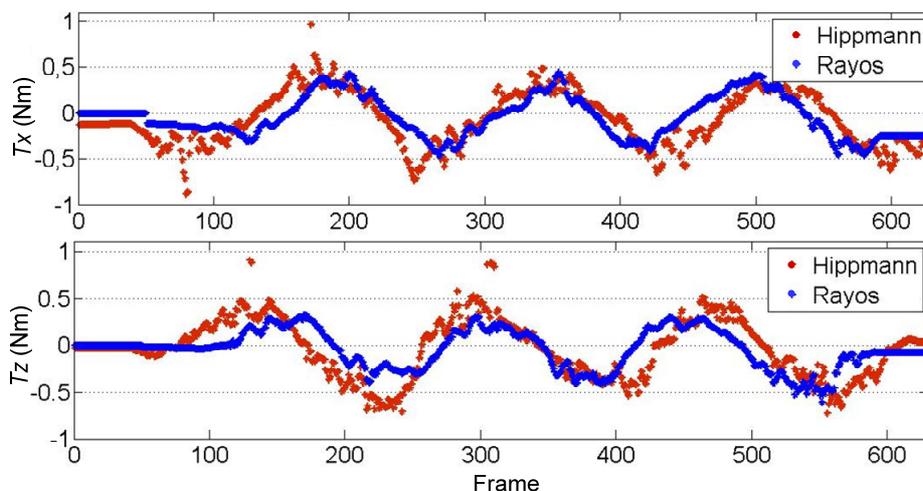


Figura 6.24: Simulación de tres giros de la abrazadera sobre el tubo central de la maqueta.

En segundo lugar, con el objetivo de analizar diferentes movimientos, se ha realizado una simulación (Figura 6.25) en la que se analizan colisiones de

distinta complejidad con el entorno. Los movimientos reproducidos en la secuencia son los siguientes:

- 1) Colisiones de la abrazadera con la tubería central del escenario (2 áreas de contacto).
- 2) Colisiones de las aletas de la abrazadera con los dos tubos adyacentes (2 áreas de contacto).
- 3) Colisiones múltiples tanto con la tubería interior como con las exteriores (4 áreas de contacto).

Las tres primeras gráficas de la figura muestran los pares calculados por el módulo de respuesta de colisión en cada periodo. La cuarta gráfica indica el número de triángulos estáticos y móviles que ha detectado el módulo de detección en cada colisión. Y la última gráfica indica el tiempo que han tardado los módulos de detección y respuesta en calcular la colisión, es decir, el tiempo total de cálculo de colisiones en cada periodo.

Los resultados obtenidos muestran que una vez más la respuesta del método basado en rayos es más continua y esta diferencia se hace más notoria en el caso de la colisión compleja (caso 3). Para analizarlo con más detalle se muestra un zoom de dos intervalos de la simulación (Figura 6.26). Dichos resultados muestran una mayor dispersión en los valores de par obtenidos con el método de Hippmann y por lo tanto una mayor discontinuidad. En referencia al coste computacional, se aprecia que no existe una diferencia significativa entre ambos métodos, obteniendo una frecuencia adecuada para los sistemas hápticos.

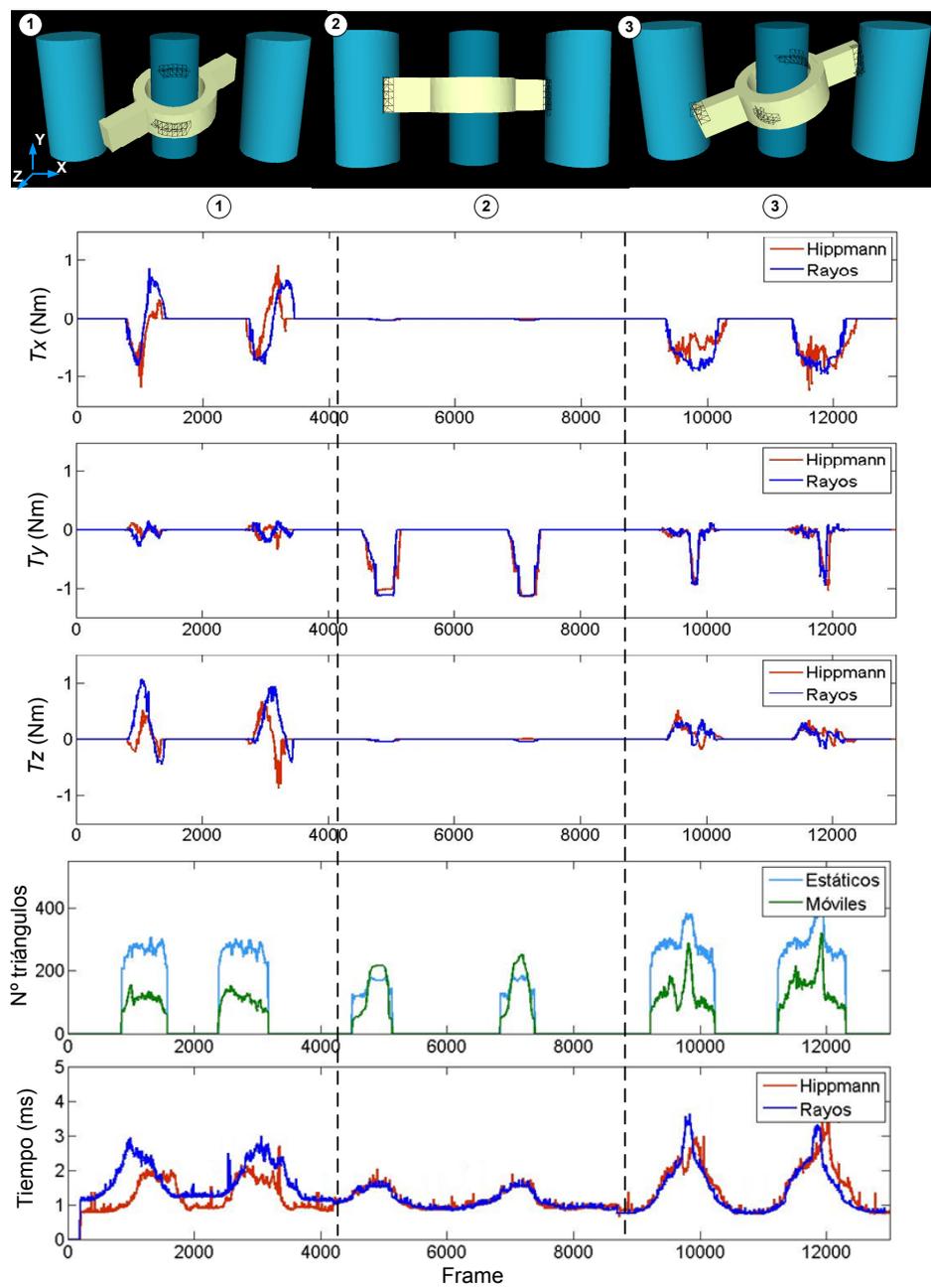


Figura 6.25: Simulación de los diferentes tipos de colisión con un entorno de objetos cilíndricos.

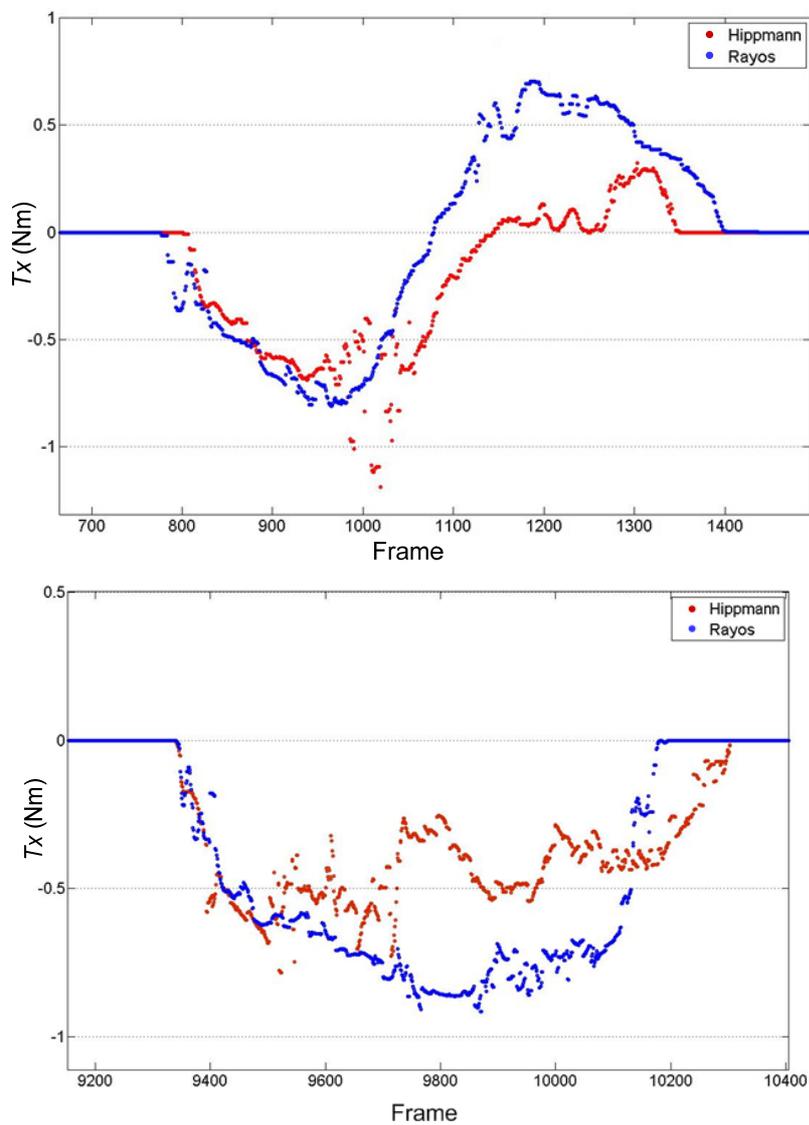


Figura 6.26: Zoom de las discontinuidades en el cálculo del momento en el eje x (T_x) para el experimento de objetos cilíndricos.

Para completar el análisis del cálculo de la respuesta háptica de momentos, a la complejidad de tener situaciones con múltiples contactos se añade la aparición de discontinuidades geométricas de tipo C^1 . El experimento que aparece en la Figura 6.27 muestra el mismo experimento anterior pero sustituyendo los tubos laterales cilíndricos por paralelepípedos.

Al igual que en el escenario anterior, se observa que el método de rayos obtiene una respuesta más continua en los pares. Incluso en el caso de mayor complejidad (zona 3), el coste computacional de este método mejora al de Hippmann.

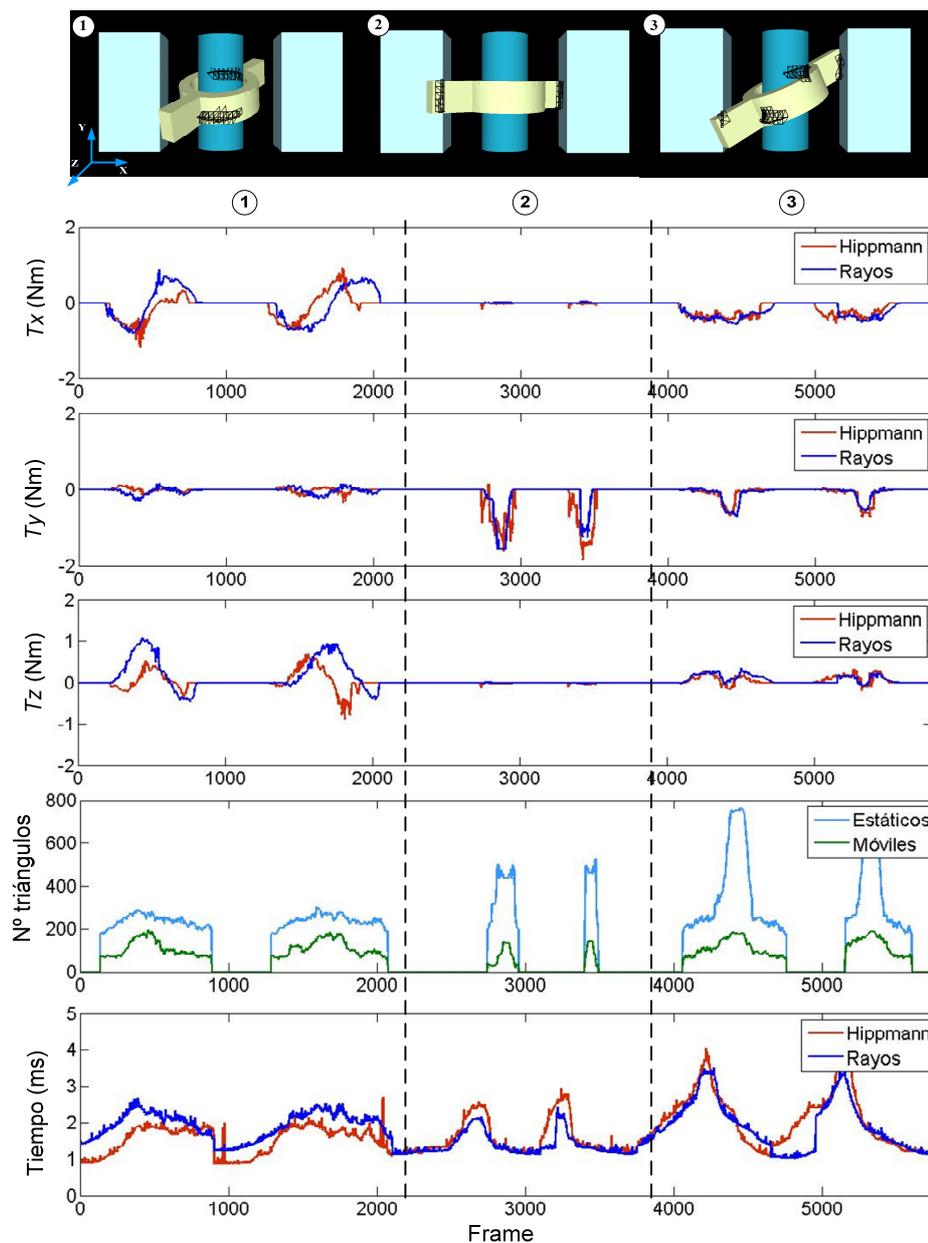


Figura 6.27: Simulación de los diferentes tipos de colisión con un entorno de objetos paralelepípedos.

También se ha comparado el comportamiento del método desarrollado en esta Tesis con el método desarrollado en Ortega (2006), en

el que se realiza una generalización del método del *god-object* aplicado a un sistema de 6 gdl.

Tal y como se ha hecho en experimentos anteriores, se ha simulado el mismo movimiento con los dos métodos, concretamente, se han realizado tres giros de la abrazadera alrededor de la tubería. Los resultados obtenidos (Figura 6.28) demuestran que el método basado en rayos obtiene una respuesta más continua que el método del *god-object*. De hecho, este último método presenta un gran número de discontinuidades a lo largo del tiempo de simulación. En cuanto al coste computacional, hay que resaltar que este último método calcula una respuesta a la colisión en un tiempo inferior a 1 ms. Sin embargo, aún siendo este tiempo considerablemente inferior al método basado en rayos, queda justificado que su comportamiento es más inestable.

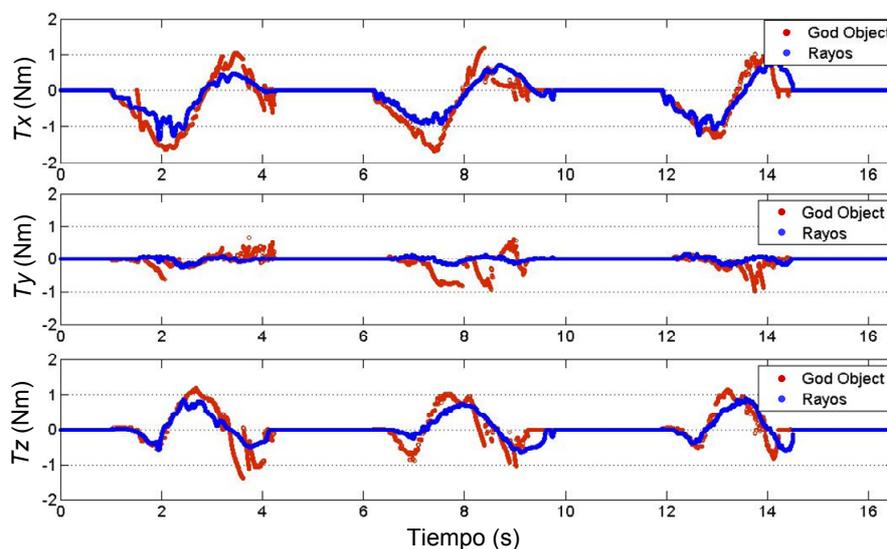


Figura 6.28: Comparación del método basado en rayos con el método del *god-object*.

6.6 EXPERIMENTOS EN TAREAS DE DESMONTAJE

Una vez analizados los resultados de los métodos de respuesta de colisión propuestos, se concluye que el método de rayos tiene un mejor comportamiento que el resto por el compromiso que alcanza entre la continuidad conseguida y su coste computacional. En este apartado, se analiza el comportamiento del método seleccionado en entornos reales más complejos que los anteriores.

El objetivo final de este trabajo es la aplicación del cálculo de la respuesta de colisión a tareas de accesibilidad o de montaje/desmontaje proporcionando al usuario una respuesta realista además de agradable. Concretamente, se ha estudiado su comportamiento en tareas de desmontaje orientadas a la extracción de piezas en modelos aeronáuticos (Figura 6.29). En este tipo de escenarios tan complejos es usual encontrar la pieza a extraer rodeada de todo tipo de elementos (tornillos, tubos...) por lo que su extracción resulta una tarea compleja.

La simulación de este tipo de tareas en entornos tan complejos permite detectar en fase de diseño si una pieza es fácilmente extraíble o por el contrario si su emplazamiento es poco accesible por lo que el diseño del modelo tendría que modificarse para evitar problemas una vez construidos los modelos reales.

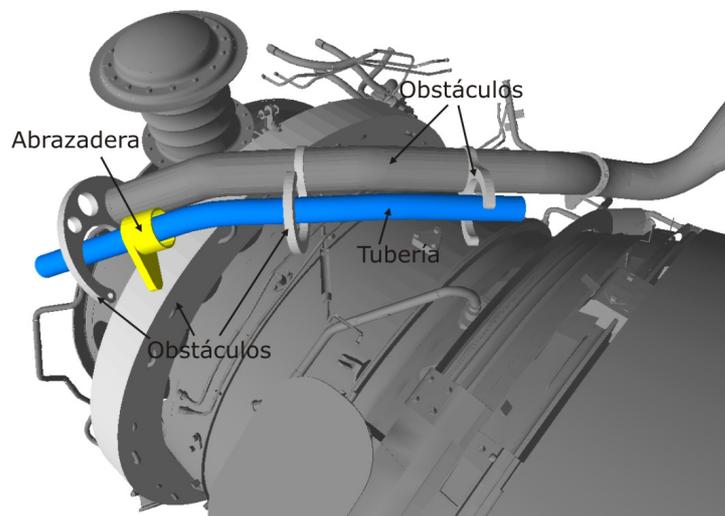


Figura 6.29: Entorno de simulación de la extracción de una abrazadera a través de una tubería de una turbina de avión.

Por lo tanto, la tarea diseñada para el experimento es similar a la extracción de una abrazadera por una tubería. Una vez que la abrazadera ha sido aflojada, el camino de salida se realiza siguiendo la trayectoria espacial que marca la propia tubería (Figura 6.30). La existencia de obstáculos a lo largo de este camino, obliga al usuario a rotar la abrazadera además de a desplazarla. El entorno utilizado está formado por una turbina de avión definida con 100.000 polígonos y la herramienta virtual (abrazadera) de 2.000 triángulos.

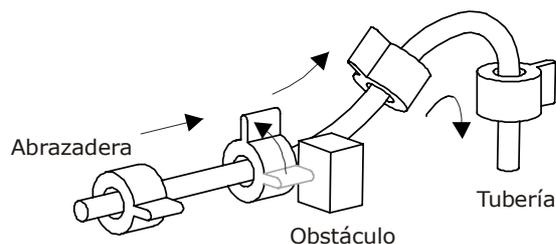


Figura 6.30: Extracción de la abrazadera de una tubería.

Para poder simular el desplazamiento de la abrazadera a lo largo de la tubería es necesario incorporar una guía lineal a la palanca háptica. Este último gdl está sensorizado y actuado. Las características principales de estos dispositivos (palanca y guía lineal) se detallan en el Anexo A.

El sistema completo con el que se ha realizado el experimento se muestra en la Figura 6.31. De esta forma se dispone de un sistema de 4 gdl (3 de rotación y uno de traslación).

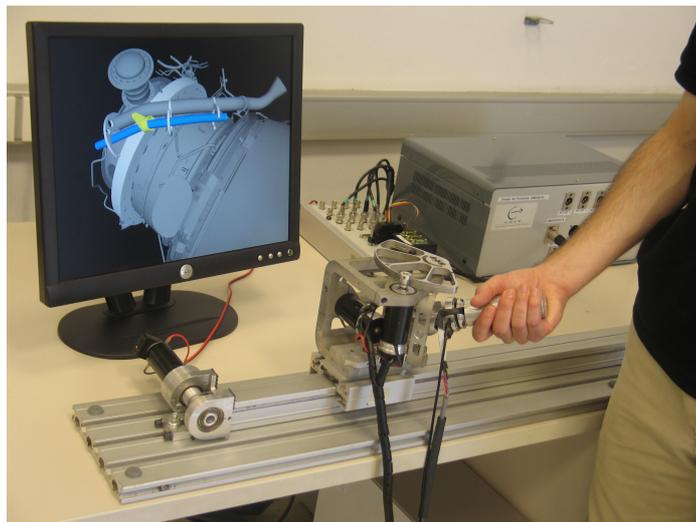


Figura 6.31: Sistema utilizado para hacer el experimento.

La simulación se realiza de derecha a izquierda y a lo largo del recorrido de la extracción. Se han considerado diferentes tipos de colisión que pueden producirse con el entorno en los diferentes ejes. Las principales fases son (ver Figura 6.32):

1. Posición inicial
2. Colisión de la abrazadera con la arandela. En este caso la guía lineal proporciona una fuerza para evitar que la abrazadera siga avanzando linealmente (un área de contacto).
3. La abrazadera se encuentra dentro del obstáculo y al girar en el eje x colisiona en la parte superior de la arandela (un área de contacto)
4. Colisión al girar sobre el eje z (un área de contacto)
5. Colisión compleja con par resultante en varios ejes (dos áreas de contacto)
6. Posición final

Esta figura además de mostrar el par que se ejerce en cada una de las componentes (T_x , T_y y T_z), muestra la fuerza ejercida por la guía lineal (F_x). También se indican el número de triángulos en colisión en cada uno de los casos además del coste computacional en cada posición. El coste computacional en cada posición se presenta descompuesto en el coste de la detección de los triángulos en contacto y el coste del cálculo de la respuesta de colisión.

Los resultados muestran que la respuesta obtenida es continua aún tratándose de un entorno complejo y que el coste computacional es suficiente para garantizar la estabilidad final del sistema, como se puede comprobar en la continuidad de las gráficas de pares obtenidas.

En la Figura 6.33 otro experimento muestra con mayor detalle el comportamiento del método en el caso de producirse una colisión compleja cuya respuesta implica el cálculo del momento en los tres ejes. Concretamente, se pasa de un estado de movimiento libre a colisionar con dos áreas de contacto y por último con tres áreas de contacto. Los triángulos dibujados en negro representan los triángulos en contacto del objeto móvil.

Los resultados obtenidos muestran que al igual que en el caso anterior, la respuesta calculada no presenta discontinuidades por lo que garantiza la estabilidad del sistema.

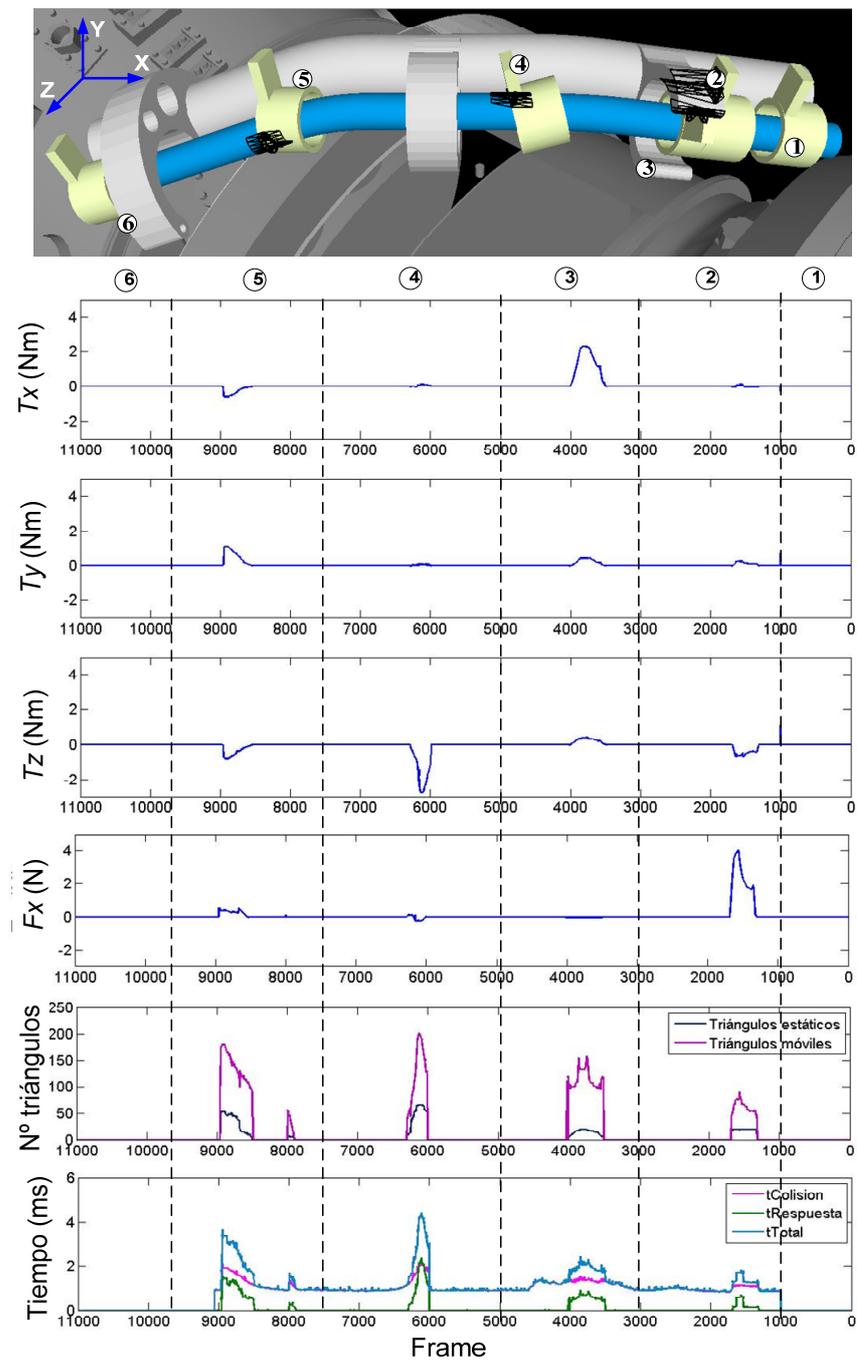


Figura 6.32: Escenario real de desmontaje para testear el cálculo de los momentos.

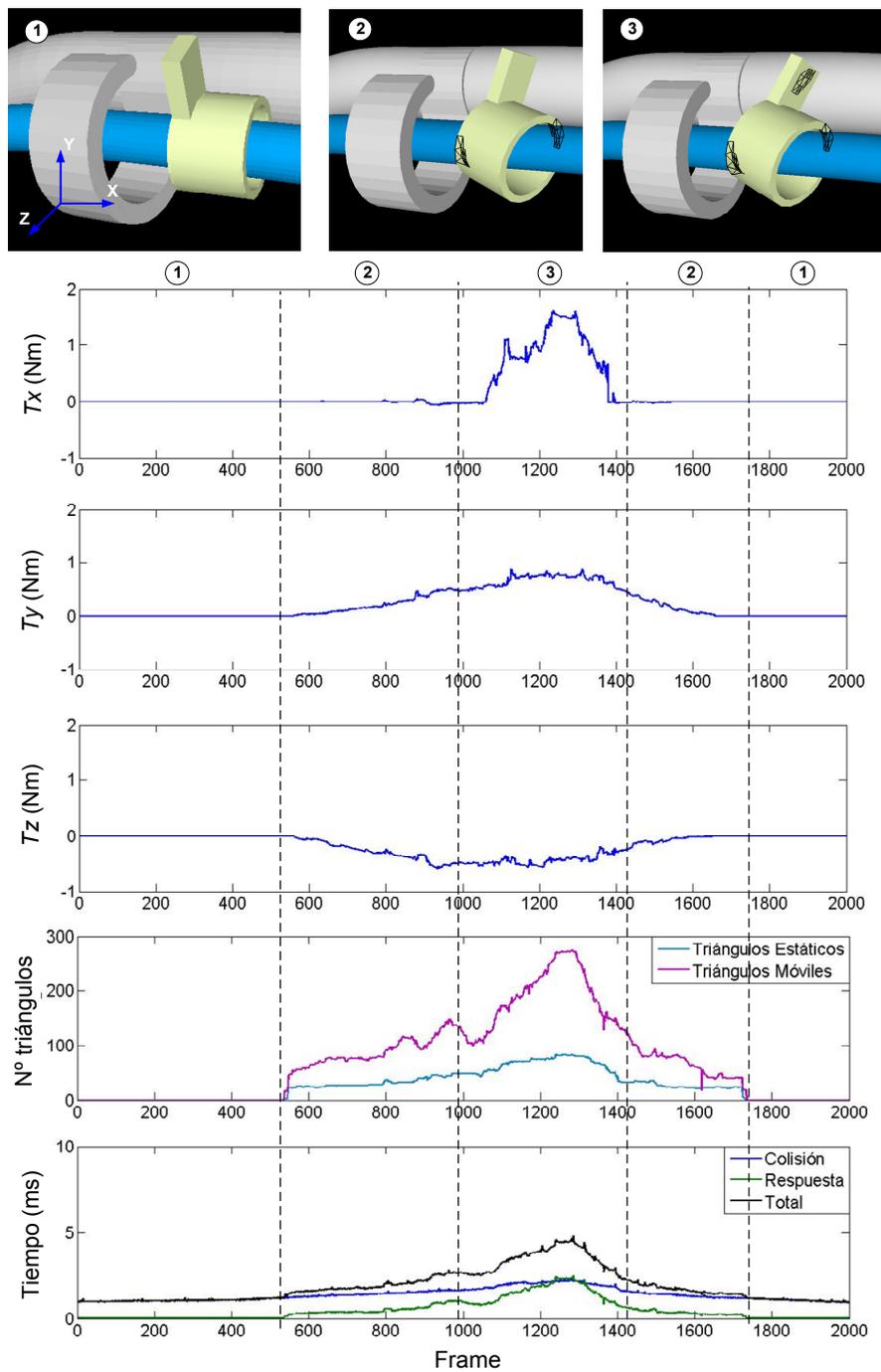


Figura 6.33: Caso detallado de una colisión compleja en un entorno real.

6.7 LAZO DE CONTROL

La respuesta háptica calculada por el módulo de respuesta de colisión es enviada al módulo de control para que se aplique al dispositivo háptico. Los algoritmos de control se encargan de adaptar estas fuerzas y pares ideales calculados por el módulo de respuesta a las capacidades mecánicas del dispositivo háptico, y garantizar una frecuencia de refresco de al menos 1 kHz en los actuadores. Para ello, el módulo de control utiliza algoritmos de interpolación y predicción (Gil 2003) que permiten estimar la fuerza y el par que debe devolverse al usuario cuando no exista información de colisión, minimizan el error de retraso entre los distintos módulos y disminuyen los cambios bruscos en las fuerzas o pares. Estas operaciones son especialmente necesarias para no degradar la sensación háptica final en el usuario.

Nótese que cuanto más continua y rápida sea la señal de fuerza y par a tratar por el módulo de control, mejor funcionarán los algoritmos de predicción y estimación utilizados. Por lo tanto, es esencial que el módulo de respuesta de colisión proporcione una información de contacto adecuada.

A continuación se muestran dos ejemplos de la actuación del módulo de control sobre el par calculado por el módulo de respuesta de colisión en los experimentos virtuales descritos anteriormente.

La Figura 6.34 es una ampliación de la primera colisión producida en la Figura 6.27 (*frames* 250-650), para el par en el eje X devuelto al usuario. En este caso, los datos se toman a la frecuencia del módulo de control (1 kHz), y se muestran tanto el par calculado por el módulo de respuesta de colisión como el par final aplicado al usuario mediante los algoritmos de control.

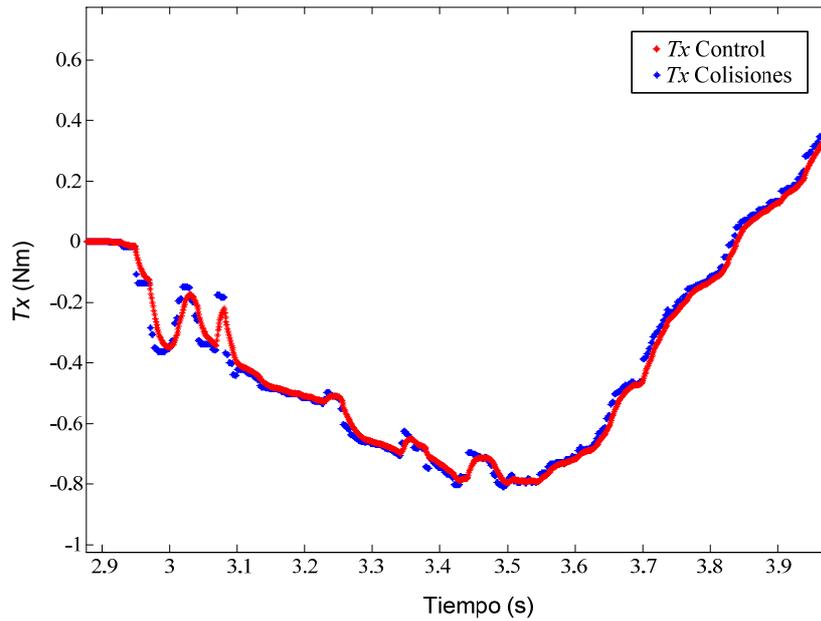


Figura 6.34: Actuación del algoritmo de control.

Debido a que la frecuencia de cálculo del módulo de respuesta de colisión es inferior al periodo de muestreo del módulo de control, se observa cómo los valores de par calculados por el módulo de respuesta se repiten varios periodos de muestreo. En la figura se puede observar cómo la actuación del algoritmo de control permite dar una respuesta adecuada en estos periodos, prediciendo la variación del par en instantes sucesivos, y además eliminando posibles “saltos” en la señal cada vez que se actualiza. De esta manera se evita que estos cambios pudieran ser percibidos por el usuario como pequeñas vibraciones que pudieran condicionar el buen funcionamiento de la aplicación.

La Figura 6.35 es una ampliación de los primeros instantes de la colisión producida en la zona 3 de la Figura 6.32 (*frames* 3450-3650), para el par en el eje X devuelto al usuario. Al igual que para el escenario anterior se muestran tanto el par calculado por el módulo de respuesta de colisión como el par calculado por el módulo de control. En este caso también, se observa el refinamiento en la señal logrado mediante los algoritmos de control desarrollados.

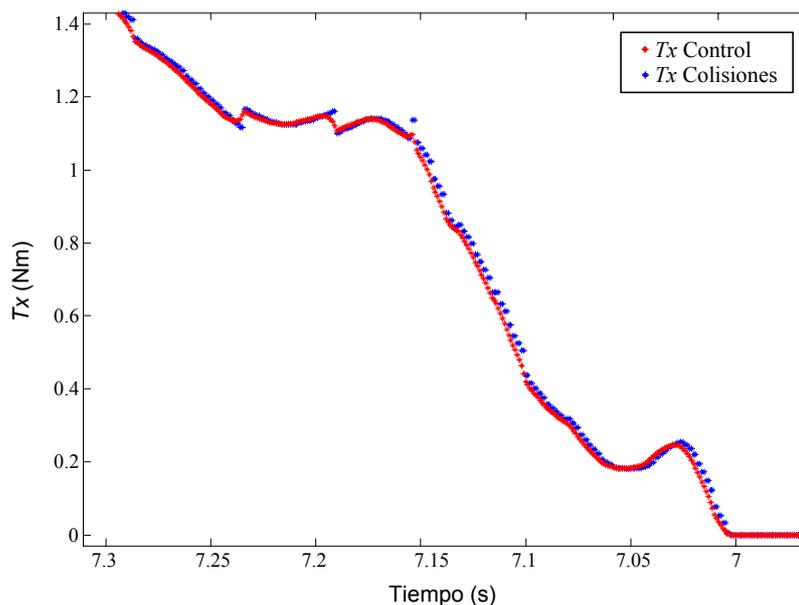


Figura 6.35: Actuación del algoritmo de control.

6.8 RESUMEN DE RESULTADOS

En primer lugar se ha descrito la arquitectura global del sistema, especificando las frecuencias de cada uno de los módulos que lo componen y el intercambio de información entre estos módulos.

Se han diseñado y realizado una serie de experimentos para llevar a cabo la comparativa entre las diferentes propuestas y seleccionar la más adecuada a los entornos utilizados en esta Tesis. Como el principal objetivo es garantizar una respuesta agradable y continúa para el usuario en situaciones complejas, era necesario que el método de evaluación contemplase la percepción real del usuario con cada uno de los métodos propuestos. Por ello, se ha concluido que la mejor forma de contrastar los algoritmos sea utilizar pruebas interactivas para cada método realizando aproximadamente el mismo movimiento en cada caso y utilizando en cada prueba el método correspondiente.

Utilizando modelos sencillos o teóricos, se ha realizado un análisis de forma desacoplada de las fuerzas y los momentos obtenidos en situaciones de múltiples contactos y discontinuidades geométricas.

El estudio de las fuerzas se ha centrado en el análisis de las variaciones de la dirección y de la magnitud de las fuerzas resultantes, que son los factores que van a condicionar la estabilidad de la respuesta háptica final. Los resultados obtenidos muestran que el método basado en rayos es el que mejor se comporta en este tipo de situaciones calculando una respuesta suave y continúa sin grandes discontinuidades que puedan incomodar al usuario.

En el caso de los pares, el análisis se realiza directamente sobre el par obtenido en cada una de sus componentes. La calidad de la respuesta es similar con los métodos basados en el z-buffer y el método basado en rayos. Sin embargo, el último tiene un coste computacional considerablemente menor y es fácilmente paralelizable por lo que se considera más adecuado para ser utilizado en este tipo de sistemas.

Finalmente, se ha analizado el comportamiento del método escogido en un entorno real que simula una tarea de desmontaje en entornos aeronáuticos obteniendo buenos resultados tanto en estabilidad como en coste computacional.

CAPÍTULO 7

ESTRATEGIAS MULTISENSORIALES

Los resultados de este capítulo se han publicado en:

Díaz, I., Hernantes, J., Mansa, I., Lozano, A., Borro, D., Gil, J.J., and Sánchez, E., "Influence of Multisensory Feedback on Haptic Accessibility Tasks", *Virtual Reality, Special Issue on Multisensory Interaction in Virtual Environments*, Vol. 10, No. 1, Pp. 31-40. May 2006.

Martín, J., Savall, J., Díaz, I., Hernantes, J., and Borro, D. (2009). "Haptic Wrists: An Alternative Design Strategy Based on User Perception". *Journal of Computing & Information Science in Engineering* (**in press**).

Los seres humanos son capaces de percibir el entorno que les rodea mediante todos sus sentidos. Aunque en la mayoría de los casos la vista sea el sentido predominante, el resto de los sentidos son también necesarios para llevar a cabo las tareas cotidianas.

Actualmente, la posibilidad de ofrecer al usuario la capacidad de utilizar todos sus sentidos de una forma natural en un entorno virtual supone uno de los principales objetivos del área de investigación de la Realidad Virtual. Por este motivo, a lo largo de estos años se han ido introduciendo nuevos sentidos (oído, tacto...) a estos sistemas con el objetivo de mejorar e incrementar su realismo (Pai 2005).

Si bien las simulaciones virtuales creadas hoy en día son visualmente muy realistas, los dispositivos hápticos están todavía muy lejos de transmitir

a los usuarios unos estímulos táctiles tan reales como los de la vida real. Además, el efecto de inmersión en este tipo de sistemas no es total, puesto que el usuario asume que no se encuentra en un mundo real. Es por ello que la combinación de información proveniente de distintos canales sensoriales puede beneficiar la percepción de realismo del entorno virtual, e incluso puede ser utilizada para solventar carencias del sistema.

A pesar de las limitaciones de los sistemas de Realidad Virtual, los escenarios virtuales también presentan algunas ventajas con respecto a los reales. El hecho de tener que generar y programar los estímulos sensoriales que el usuario va a percibir, permite también que estos puedan ser modificados. Es decir, los estímulos pueden generarse para reproducir aquella verdad física que pretenden representar, o bien para que el usuario perciba aquella verdad que interese que perciba. Este hecho resulta de gran relevancia puesto que de esta manera es posible sustituir carencias de software y hardware por “ilusiones” en la percepción del usuario.

En este apartado se desarrollan y evalúan dos estrategias multisensoriales con el objetivo de ahondar en el mundo de la percepción multisensorial en los entornos de realidad virtual. El objetivo es analizar cómo pueden utilizarse, combinarse e interactuar los distintos canales sensoriales disponibles para mejorar tanto la realización como la percepción de tareas virtuales. Como ya se ha mencionado anteriormente, las diferentes modalidades sensoriales pueden combinarse en un entorno virtual con el objetivo de *aumentar* la percepción de un evento, o *sustituir* un canal sensorial no disponible en el sistema por otro. La primera analiza si la incorporación de un nuevo canal sensorial mejora la eficiencia a la hora de realizar una tarea virtual. En cambio, la segunda sustituye el canal háptico por otras modalidades sensoriales y analiza su viabilidad para llevar a cabo tareas que requieren de la restitución de momentos. Mediante estos dos experimentos, se quiere demostrar que la predisposición del sujeto y aspectos psicológicos también influyen en la percepción del entorno y pueden incrementar la inmersión de los usuarios en estos sistemas.

A pesar de las ventajas potenciales que presentan las estrategias multisensoriales, hay que tener en cuenta que en un escenario virtual la información sensorial no se procesa de manera tan natural a como se hace en un entorno real. Y como consecuencia, los estímulos sensoriales provenientes de los distintos canales pueden llegar a sustituirse, complementarse o interferirse entre ellos. El hecho de disponer de varios canales sensoriales no garantiza que una tarea se realice de forma más

óptima y por este motivo, es necesario analizar el impacto que tiene en el usuario el hecho de añadir o sustituir diferentes estímulos en el entorno virtual.

7.1 AMPLIACIÓN SENSORIAL

En los últimos años, diversos autores se han planteado la posibilidad de combinar varios sentidos para mejorar el realismo como solución a las limitaciones existentes en la restitución háptica.

Por ejemplo, uno de los mayores inconvenientes de los dispositivos hápticos es que los objetos virtuales no pueden representarse físicamente tan rígidos como en el mundo real. En (Srinivasan et al. 1996), los autores propusieron un experimento en el que simulaban dos muelles virtuales que se comprimían en función de la fuerza que ejercían los usuarios mediante un dispositivo háptico. Los autores, sin que los usuarios tuviesen conocimiento de ello, alteraban en los experimentos la rigidez “visual” de los muelles; es decir, aplicando la misma fuerza háptica un muelle se comprimía más que el otro. Los resultados de los experimentos demostraban que el efecto de cambiar la rigidez visual de los muelles provocaba que los usuarios percibieran el muelle más o menos rígido en términos de fuerza. En (Avanzini et al. 2006, DiFranco et al. 1997), los autores modificaban el volumen del sonido emitido al colisionar con un objeto virtual rígido. Del experimento se demuestra que aumentando o disminuyendo el volumen del sonido emitido en una colisión háptica se puede alterar la percepción de rigidez física del objeto colisionado.

Además de la dificultad de mostrar los objetos virtuales hápticamente rígidos, otra de los grandes retos en los sistemas hápticos reside en la percepción de texturas mediante estímulos hápticos. En este campo, Guest et al. (2002) sugieren que los usuarios pueden ser influenciados por estímulos auditivos en la percepción de texturas de superficies. En sus experimentos analizaron la influencia que podía tener la manipulación de las frecuencias auditivas en la percepción táctil de las rugosidades de diversas superficies de objetos. En la misma línea, McGee et al.(2001) proponen una combinación de estímulos auditivos y hápticos como solución para mejorar la calidad de la información sobre la textura de las superficies de los objetos virtuales.

Existen sin embargo diversos estudios que señalan que la interacción multisensorial no siempre beneficia en la consecución de determinadas

tareas, o simplemente no mejora los resultados obtenidos con respecto a tareas que no combinan estos estímulos. En (Mc Gee et al. 2000) por ejemplo, se describen las diferentes maneras en las que se puede combinar la información háptica con la auditiva para mejorar la percepción de texturas: conflictiva, redundante o complementaria. Y para tareas de discriminación de texturas, Guest et al. (2003) concluyen que la combinación del sentido del tacto y la visión da peores resultados que cuando ambos sentidos actúan independientemente.

En (Lecuyer et al. 2002), los autores investigan el efecto de añadir información de los eventos de colisión proveniente de los canales háptico, visual y auditivo para una tarea virtual de inserción de una pieza. Los resultados muestran que ninguno de estos canales tiene un impacto positivo en la reducción del tiempo para completar la tarea respecto a realizarla de manera puramente visual. Sin embargo, parece que el movimiento de los usuarios al colisionar sí se reduce cuando hay información adicional. Los autores sugieren que a más información el sujeto presta más atención a la colisión, pero en consecuencia también realiza la tarea más despacio. En un experimento similar (Edwards et al. 2004), los autores también recomiendan el uso de información proveniente de distintos canales, tan sólo en función del objetivo de la tarea virtual.

7.1.1 INFLUENCIA DEL SONIDO EN TAREAS DE ACCESIBILIDAD

El uso de sistemas de Realidad Virtual es cada vez más frecuente en áreas tales como la medicina o la industria. Sin embargo, existen numerosas tareas que resulta difícil llevar a cabo únicamente recibiendo un feedback visual. Este es el caso por ejemplo de la simulación de tareas de accesibilidad cuyo objetivo es la búsqueda del camino más adecuado para acceder a un determinado elemento de difícil acceso dentro de un sistema. En las simulaciones de operaciones quirúrgicas por ejemplo, es necesario determinar el mejor camino para poder llegar al órgano dañado sin perjudicar el resto de órganos.

En el campo de la industria, la accesibilidad debe ser analizada por ejemplo en tareas de mantenimiento de sistemas mecánicos donde se utilizan modelos complejos con un espacio de trabajo limitado. En estos casos normalmente resulta complicado acceder a un componente específico debido al elevado número de elementos (cables, tuberías...) que pueden encontrarse por el camino y a su alrededor (Figura 7.1). En este contexto el

estudio de accesibilidad es necesario para encontrar un camino adecuado para proceder al ensamblado o desensamblado de una pieza determinada.

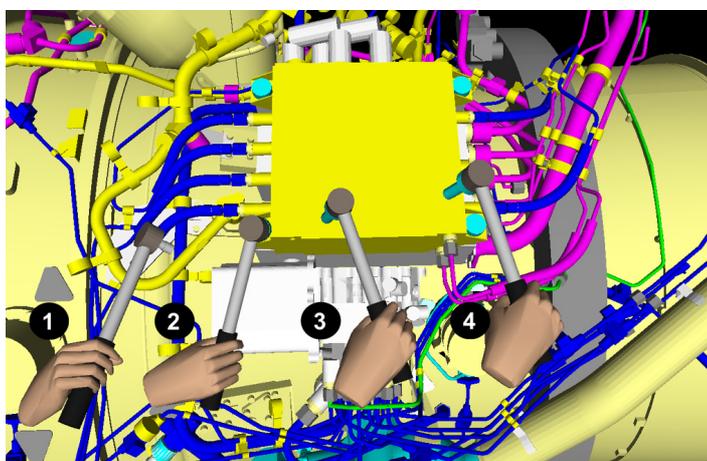


Figura 7.1: Secuencia de desensamblado de una pieza en un motor de avión

Obviamente, un análisis visual no es suficiente para detectar posibles partes inaccesibles, salvo que el modelo sea muy simple. Además, el operario debe explorar y manipular el modelo o diferentes partes de él para completar el análisis, por lo que además del sentido visual es necesaria la utilización de otros sentidos. En este caso concreto, el sentido del tacto es fundamental ya que el hecho de recibir una reflexión de fuerzas al colisionar con los objetos virtuales, mejora la percepción, así como el análisis de las tareas de mantenimiento (Savall et al. 2002). Sin embargo, y como ya se ha mencionado anteriormente, la restitución háptica no es todavía tan óptima como para asemejarse fielmente a la realidad. Por ello, lo que se pretende evaluar es la capacidad del sonido como instrumento para:

- 1) mejorar la percepción de realismo de la simulación virtual, y
- 2) mejorar las interacciones con el entorno, dando al usuario una mejor percepción de los eventos de colisiones con los objetos virtuales.

En este caso, el objetivo tanto del canal háptico como del auditivo es alertar al usuario de las colisiones con el entorno. A priori parece que al añadir el canal auditivo aumentará el realismo de la operación y ayudará al operario a mejorar el rendimiento general de la tarea y a realizarla de manera más natural. Sin embargo, y como ya se ha mencionado anteriormente, los distintos canales sensoriales no se integran del mismo modo en el mundo

virtual que en el real. Para poder analizar todas estas hipótesis, se ha realizado un estudio estadístico a partir de la evaluación de una tarea virtual de accesibilidad ejecutada por múltiples usuarios.

7.1.2 DESCRIPCIÓN DEL SISTEMA MULTISENSORIAL

El dispositivo háptico utilizado para evaluar la influencia de la combinación de los distintos canales sensoriales es el LHIFAM (véase Anexo A). Tanto la descripción del sistema como los algoritmos de renderizado utilizados ya han sido descritos en capítulos anteriores. La novedad reside en el módulo de renderizado auditivo que se ha añadido al sistema.

En este caso, el tipo de sonido generado es un sonido de contacto entre objetos metálicos. Para ello se utiliza el algoritmo implementado en el Departamento de Mecánica Aplicada del CEIT de síntesis modal de audio descrito en (van den Doel et al. 1998). Este algoritmo está basado en la dinámica de vibraciones y puede simular efectos de forma, localización del contacto o tipo de material. La gran ventaja de este algoritmo es que el sonido emitido depende de la fuerza de contacto debida a la interacción. De esta manera, más allá de emitir un sonido pregrabado en el instante adecuado, se consigue que los estímulos auditivos y hápticos cambien al mismo tiempo cuando el usuario aplica distintas fuerzas al objeto.

Tan importante como generar un sonido adecuado es que los distintos estímulos hápticos, auditivos y visuales estén debidamente sincronizados. De manera independiente, se conoce que una visualización adecuada de un evento debe representarse al usuario a una frecuencia de 20-25 Hz, y que la restitución háptica debe comandarse a una frecuencia de al menos 1.000 Hz. En cuanto al sonido, una frecuencia óptima viene a ser de 44.100 Hz (valor utilizado en los CDs de música comerciales), aunque una frecuencia en torno a los 20.000 Hz se considera suficiente.

Una buena sincronización entre todos los módulos es esencial debido a que cada módulo trabaja a una frecuencia distinta. Si los estímulos no están perfectamente sincronizados, la percepción de los eventos puede ser muy pobre y puede reducir la concentración del usuario al realizar una tarea. No sería muy realista sentir la fuerza de un contacto con un objeto, y escuchar el correspondiente sonido más tarde.

La Figura 7.2 muestra la manera en la que se sincronizan los distintos módulos de renderizado. La sincronización más crítica es la referente al sonido y a la fuerza, debido a que son estímulos que el ser humano es capaz

de sentir a altas frecuencias. En (DiFilippo et al. 2000) muestran que un límite válido para el retardo entre los estímulos hápticos y auditivos está en 2 ms.

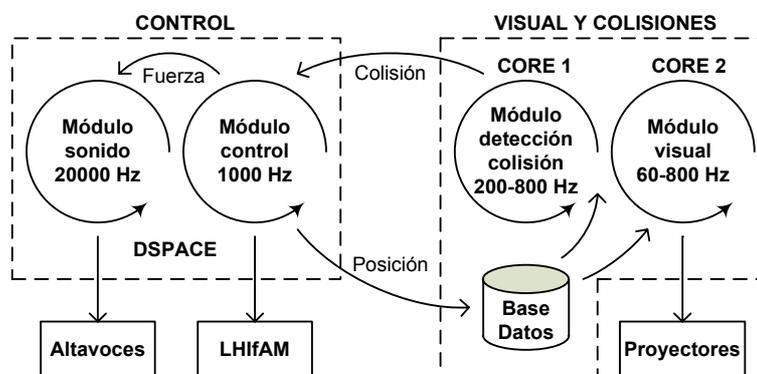


Figura 7.2: Descripción de los distintos módulos que componen el interfaz háptico multisensorial.

El módulo de sonido carece de información de la fuerza háptica durante varios periodos de muestreo debido a que trabaja a una frecuencia superior a la del lazo de control, problema análogo al que ocurre entre los módulos de control y colisiones. En este caso, se ha comprobado que manteniendo la fuerza háptica constante durante los periodos sin información se obtienen buenos resultados en la generación de sonido, con lo que no ha sido necesario implementar ningún algoritmo de interpolación o predicción.

7.1.3 DESCRIPCIÓN DEL ESCENARIO

La elección del tipo de escenario a utilizar en las pruebas no fue una tarea sencilla ya que era necesario solventar principalmente dos problemas. En primer lugar, las tareas de accesibilidad generalmente implican desplazamientos precisos y una buena percepción del entorno en el que se está trabajando. Sin embargo, la mayoría de los participantes en el estudio no estaban familiarizados con aplicaciones de realidad virtual y menos aún con la simulación de tareas de accesibilidad en estos entornos. En segundo lugar, los modelos utilizados en este tipo de tareas permiten múltiples caminos de acceso al elemento objetivo debido a su complejidad y al elevado número de componentes lo que dificulta la elección de un patrón de movimiento único para realizar una comparación fiable de todos los resultados obtenidos por cada usuario.

Teniendo en cuenta estas limitaciones, se ha optado por utilizar un laberinto ya que se considera un escenario sencillo e intuitivo donde los usuarios pueden llevar a cabo los ensayos sin necesidad de tener ningún conocimiento o experiencia en entornos virtuales, además de permitir fijar un único path que todos los participantes pudiesen repetir a lo largo de todas las pruebas. Este modelo además obliga a los usuarios a moverse en todo el espacio de trabajo definido por el dispositivo háptico denominado LHIFAM.

Las dimensiones del laberinto son de 1 m x 1 m para cubrir todo el espacio de trabajo del dispositivo. De esta forma los participantes se ven obligados a interactuar en diferentes posiciones ergonómicas tal y como sucede en las operaciones de accesibilidad. Inicialmente se posiciona una esfera en la entrada del laberinto. El diámetro de esta esfera es de 80 mm, y la separación entre las paredes del laberinto es de 82 mm, de tal manera que el usuario debe ser muy preciso en sus movimientos para no colisionar con las paredes. Por último, las paredes del laberinto son suficientemente altas como para garantizar que el usuario no pueda escaparse por la parte superior, mientras que en la parte inferior existe un suelo rígido.

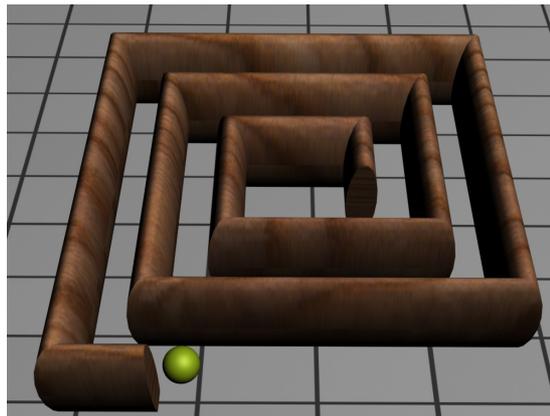


Figura 7.3: Laberinto virtual empleado en los experimentos.

El modelo de contacto utilizado para las colisiones entre la esfera y las paredes es un modelo viscoelástico. Los coeficientes de rigidez y viscosidad utilizados para dicho modelo son 600 N/m y 1 Ns/m, respectivamente.

Además, para el movimiento tangencial por las paredes el control implementa una fuerza de fricción propuesta por Salisbury et al. (1995a). El sonido generado por la esfera al colisionar con las paredes del laberinto es

un sonido metálico proporcional a la fuerza háptica devuelta al usuario en la colisión. Y en caso de permanecer en contacto, el sonido generado es proporcional a la fuerza de fricción háptica, tipo *stick-slip*.

7.1.4 USUARIOS Y PROCEDIMIENTO

En el experimento han participado 12 individuos, de los cuales 9 eran hombres y 3 mujeres, con una edad comprendida entre los 25 y los 32 años y sin ningún tipo de deficiencia auditiva, visual o táctil. La mayoría de estos participantes no había tenido ninguna experiencia anterior en el uso de sistemas hápticos y todos ellos desconocían las hipótesis a demostrar mediante el experimento.

Los usuarios deben completar un recorrido desde la entrada del laberinto hasta el centro y una vez completado, realizarlo a la inversa. Para cada prueba, el modelo del laberinto es proyectado utilizando visualización estereoscópica. Inicialmente, se permitió a todos los participantes interactuar por un corto periodo de tiempo con el dispositivo háptico utilizando un escenario distinto al del laberinto, con la finalidad de que pudiesen familiarizarse con el sistema.

En cada una de las 4 pruebas, los participantes percibían la información del evento de colisión mediante la combinación de diferentes estímulos sensoriales: únicamente feedback visual (V), visual con feedback auditivo (VS), visual con feedback háptico (VF) y finalmente visual, sonido y fuerza (VSF).

Cada participante realizó el experimento cuatro veces, una por cada combinación de feedbacks posible con un orden distinto para cada participante. También se estableció un periodo mínimo de una hora entre prueba y prueba realizada por el mismo usuario, para evitar de esta forma cualquier curva de aprendizaje.

En las instrucciones proporcionadas a cada uno de los usuarios se resaltaba que el objetivo del experimento era evitar las colisiones entre la esfera y las paredes del laberinto. También se les informó de que percibirían un tipo de estímulo diferente asociado a la colisión en cada una de las pruebas.

Para determinar la calidad y efectividad con la que se ha llevado a cabo la tarea, se toma la penetración como parámetro evaluador. Cuanto menor sea este valor, mayor será la precisión en la tarea de accesibilidad lo

que a su vez implica una mejora en la percepción del entorno por parte del usuario.

7.1.5 RESULTADOS

En la Figura 7.4 se muestran los valores medios de penetración obtenidos por cada participante además de una representación gráfica de estos resultados mediante un diagrama de cajas. Cada caja representa los rangos donde se concentran el 50% de los datos obtenidos, cuyos extremos vienen determinados por los cuartiles 1 y 3. La línea interior a las cajas representa las medianas del conjunto de datos. Las líneas que salen de cada caja indican la posición del máximo y el mínimo valor de la distribución (valores extremos) eliminando los valores atípicos de la misma.

Sujeto	V	VS	VF	VSF
1	5,18	4,80	1,52	1,25
2	4,48	3,87	0,92	0,68
3	4,03	3,08	0,70	0,69
4	5,31	3,55	0,78	0,72
5	4,28	3,23	1,01	0,72
6	7,46	5,65	1,11	0,89
7	5,66	5,98	0,97	0,87
8	7,31	4,32	1,06	0,65
9	5,13	3,77	1,14	1,41
10	4,61	4,18	1,11	0,69
11	4,65	4,13	0,94	0,69
12	4,94	4,13	0,81	0,49
Media	5,25	4,22	1,01	0,81
Desv. Est.	1,09	0,88	0,21	0,26

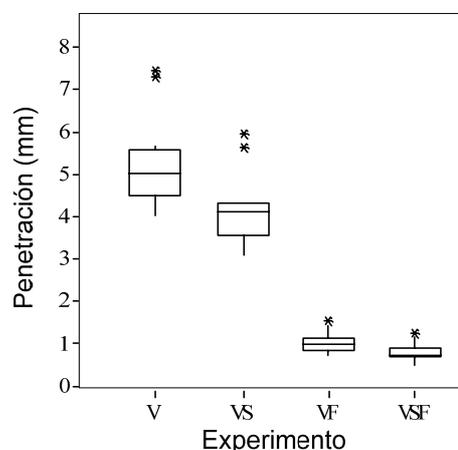


Figura 7.4: Penetración media de colisión (mm) por cada usuario en cada modalidad, y diagrama de cajas con los datos (la línea dentro del rectángulo representa el valor de la mediana, y los asteriscos los valores atípicos).

Analizando cada usuario individualmente, se puede afirmar que todos los participantes obtienen mejores resultados con reflexión de fuerzas como era de esperar ya que este feedback incrementa enormemente el realismo de la simulación. El interés del análisis estadístico realizado reside en el estudio de la influencia del efecto del sonido, por lo que los datos obtenidos se dividen en dos grupos: experimentos sin fuerzas (V y VS) y experimentos con fuerzas (VF y VSF). En el grupo de experimentos sin fuerzas, todos los participantes obtuvieron mejores resultados en las pruebas con sonido excepto uno (sujeto 7). En el grupo de experimentos con fuerzas, también todos mejoraron con el sonido excepto uno (sujeto 9). En ambos casos, los

valores de penetración obtenidos fueron estadísticamente atípicos en comparación con los obtenidos por el resto de sujetos.

Para determinar diferencias estadísticas, se ha realizado un test *t* *pareado* entre los experimentos V y VS. Los resultados muestran una diferencia significativa entre ambos grupos ($p = 0,002$), con una mejora en la tarea del 20% si se usa el sonido. El mismo análisis entre los grupos VF y VFS muestra también diferencias estadísticamente significativas ($p = 0,005$), con una reducción del 19% en el valor medio de la penetración.

En general, los resultados indican que los usuarios realizaron mejor la prueba cuando disponían de una combinación de todas las modalidades disponibles. Tomando como experimento base los realizados sólo con el estímulo visual, los valores medios de los experimentos con visión y sonido mejoran los resultados en un 20%, mientras que para los experimentos con visión y reflexión de fuerzas la mejora es del 80%. Por último, en el experimento que combina todos los estímulos, la mejora es del 84%. Por lo tanto, con estos datos, se confirma que incluir el estímulo auditivo es beneficioso para la consecución de tareas de accesibilidad virtuales. También se concluye que para la accesibilidad que se estudia en este experimento, el canal sensorial del tacto es el más importante.

Por otra parte, se observó que en general, en los experimentos con sonido los usuarios tardaban más tiempo en completar la prueba. Esto puede llevar a pensar que en este caso si el parámetro de calidad del experimento hubiese sido el tiempo total en realizar la prueba, puede que el resultado con ayuda del canal auditivo no hubiese sido mejor.

A pesar de que en las instrucciones se dijo a los usuarios que el tiempo total en completar la tarea no sería medido, en realidad sí que lo fue. Estos tiempos fueron muy dispares entre usuarios debido a que cada uno realizaba el experimento a velocidades muy distintas. La Figura 7.5 muestra los valores obtenidos.

Sujeto	V	VS	VF	VSF
1	60	62	40	53
2	67	122	106	100
3	70	82	80	90
4	80	91	85	80
5	68	96	69	80
6	68	77	53	64
7	95	90	70	68
8	76	110	69	91
9	96	100	70	85
10	142	134	105	120
11	130	135	120	146
12	106	120	117	146
Media	88,16	101,58	82	93,58
Dev. St.	26,41	25,25	23,01	29,95

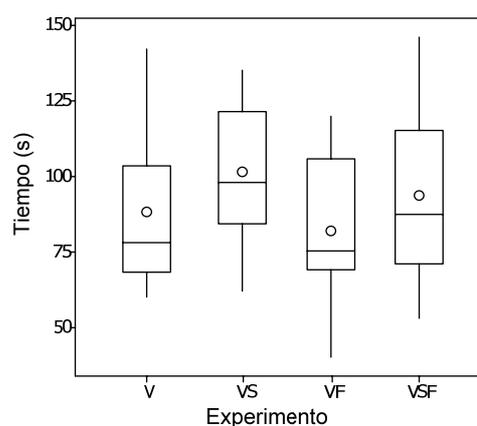


Figura 7.5: Tiempo (s) empleado por los participantes en completar la tarea en cada modalidad, y diagrama de cajas con los datos (la línea dentro del rectángulo representa el valor de la mediana, y los círculos los valores medios)

Si se comparan los valores medios obtenidos en las pruebas V y VS, los participantes tardaron 13,4 segundos más de media en realizar la prueba con sonido, mientras que en la prueba VSF los participantes necesitaron 11,5 segundos más que en la modalidad VF. Se ha realizado un test *t pareado* para cada grupo (con y sin fuerzas) y los resultados del análisis muestran que existen diferencias estadísticamente significativas entre los grupos: V-VS ($p = 0,024$), y VF-VSF ($p = 0,005$). Sin embargo, debido a que las diferencias entre los tiempos medios están por debajo de la desviación estándar en cada modalidad no se puede afirmar definitivamente que el tiempo añadido sea debido a la integración del canal auditivo. Es decir, son más significativas las disparidades encontradas entre cada usuario que entre las distintas modalidades, por lo que los resultados deben tomarse con cautela. Además, nótese que estos datos son tan sólo orientativos, puesto que en ningún caso se les mencionó a los usuarios que el objetivo de la tarea era completarla en el menor tiempo posible.

7.1.6 CONCLUSIONES

Mejorar la interacción entre los humanos y las computadoras es un desafío continuo. Hay una necesidad de incrementar la inmersión y la interacción del usuario en los sistemas de Realidad Virtual. Este hecho motiva la incorporación de nuevos canales sensoriales al proceso. Por ello, se han realizado experimentos estadísticos con un dispositivo háptico

especialmente diseñado para tareas de mantenimiento virtuales con capacidad para generar sonidos.

Se ha presentado un estudio multiusuario que analiza la influencia de la combinación de información proveniente de distintos canales sensoriales en la consecución y el realismo de una tarea virtual de accesibilidad. Los resultados demuestran que la sinergia de los canales sensoriales visual, auditivo y háptico ofrece mejores resultados en la consecución de estas tareas que en sistemas con menos canales sensoriales. Sin embargo, también se observa en los resultados que la inclusión del canal auditivo puede afectar negativamente en factores como el tiempo de ejecución de las tareas.

7.2 SUSTITUCIÓN SENSORIAL

El concepto de sustitución sensorial es definido por Bach-y-Rita et al. (1987) y posteriormente estudiada por Lenay et al. (2003) como la provisión al cerebro de información que habitualmente se encuentra en un dominio sensorial (por ejemplo información visual a través de los ojos) por medio de receptores y áreas cerebrales de otro sistema sensorial (por ejemplo información visual a través de la piel y el sistema somatosensorial).

En (Massimino 1995), el autor estudia la sustitución sensorial de la restitución de fuerza por tacto y sonido, concluyendo que se puede considerar como un método probado por el cuál se puede suministrar información cinestésica cuando la restitución tradicional de fuerzas es costosa o inviable para una tarea determinada. En particular, sus experimentos muestran que la sustitución a través de los sentidos del tacto y oído es adecuada para tareas altamente dependientes del tiempo de reacción, como por ejemplo la detección de fuerzas de contacto en colisiones.

En tareas de ensamblaje virtual, Edwards et al. (2004) investigan si la sustitución de la restitución de fuerza por estímulos auditivos adecuados mejora la manipulación y la percepción subjetiva de utilidad. Encuentran que esta sustitución puede ser una opción a tener en cuenta dependiendo del objetivo concreto fijado en la tarea (minimizar el número de colisiones, mejorar la percepción del usuario, etc.), y proponen en este sentido algunas reglas para el diseño de dispositivos. En un experimento similar (Petzold et al. 2004), los autores muestran que una combinación de estímulos visuales y auditivos es un buen sustituto del canal háptico.

En el ámbito de la medicina, Kitagawa et al. (2005) analizan el efecto de sustituir la restitución háptica por estímulos visuales y auditivos para

suministrar al cirujano una representación de la magnitud de las fuerzas resultantes de un procedimiento teleoperado de sutura. Este estudio muestra que la sustitución sensorial proporciona suficiente información como para controlar las fuerzas ejercidas. Para el mismo procedimiento y objetivo, Tavakoli et al. (2005) comparan la restitución háptica con una sustitución mediante estímulos visuales. Los resultados coinciden en que dicha sustitución es una buena solución, siempre y cuando la tarea no lleve mucho tiempo. Para tareas largas, los resultados no fueron tan buenos como los obtenidos mediante la restitución háptica, debido a la fatiga que causaban los estímulos visuales en los usuarios.

Por último, son interesantes los trabajos realizados por (Lécuyer et al. 2001, Lécuyer et al. 2000b) en el que se muestra cómo es posible sentir fuerzas utilizando dispositivos sin capacidad de restitución de fuerzas, mediante lo que los autores definen como restitución “pseudo-háptica”. Para ello, los autores utilizan dispositivos isométricos en combinación con estímulos visuales específicos que generan en el usuario, hasta cierto punto, la percepción de fuerzas.

7.2.1 SIMPLIFICACIÓN DEL DISEÑO DE UNA MUÑECA HÁPTICA

Numerosos trabajos han demostrado que la inclusión de la restitución de fuerzas en un entorno virtual aumenta el realismo y la inmersión del usuario dentro del entorno. De hecho, este tipo de feedback se considera indispensable para poder llevar a cabo tareas de mantenimiento, accesibilidad o de prototipado virtual. Aun así, en este tipo de aplicaciones se han de llevar a cabo diversas operaciones en las que es necesario sentir además de fuerzas, momentos como respuesta a la interacción con el entorno virtual tal y como ocurre en la realidad. Diversos autores respaldan la idea de que la inclusión de los pares en la respuesta háptica incrementa el realismo en las aplicaciones mencionadas anteriormente (Cohen and Chen 1999, Chen 1999, Lin and Manocha 1993).

Sin embargo, el incremento de 3 a 6 de los grados de libertad de un dispositivo háptico para proporcionar una respuesta de pares no es directo, ya que se requiere un cambio en el diseño mecánico del dispositivo. Este es el caso del dispositivo háptico LHIFAM. Aunque este dispositivo tiene 6 gdl, únicamente 3 de ellos están actuados por lo que el usuario recibe como respuesta una fuerza pero no un momento. Para poder actuar los tres grados de libertad restantes, el Departamento de Mecánica Aplicada del CEIT ha llevado a cabo un estudio de las diferentes posibilidades existentes

en la bibliografía (Martin et al. 2005) llegando a la conclusión de que ninguna de ellas satisface los requerimientos en términos de niveles altos de salida y gran espacio de trabajo simultáneamente.

Las limitaciones mecánicas inherentes al LHIFAM hacen difícil encontrar un diseño mecánico satisfactorio. Por este motivo se plantea una filosofía alternativa que se sale del puro diseño mecánico y trata de buscar soluciones más cerca de la percepción del usuario que es realmente quien va a interactuar con el dispositivo.

La solución propuesta conlleva a eliminar estratégicamente una parte del mecanismo sin eliminar completamente la capacidad de éste para transmitir la percepción de pares al usuario. Esto es posible gracias al diseño de técnicas que combinan pseudo-háptica y sustitución sensorial. El resultado proporciona mejoras en reducción de costes, simplificación cinemática y aligeramiento del dispositivo.

7.2.2 DESCRIPCIÓN DE LA ESTRATEGIA DE SUSTITUCIÓN SENSORIAL

7.2.2.1 CONCEPTO

La bondad de la solución depende de si la nueva alternativa sensorial es capaz de representar la misma calidad háptica que un sistema mecánicamente completo.

La hipótesis de partida consiste en eliminar uno de los giros (tanto de forma activa como pasiva) y tratar de restituir al usuario la percepción cinestética² perdida aplicando el concepto de los pseudo-hápticos y las ilusiones hápticas. Lo que se pretende conseguir es que la percepción del usuario pueda acercarse a los 3 gdl activos aún cuando mecánicamente el dispositivo tuviese sólo 2 gdl activos. El objetivo final es analizar la viabilidad de esta estrategia como alternativa a la complejidad en el diseño mecánico.

Para elegir en qué eje se va a bloquear el giro, se analiza la rotación de la muñeca en cada uno de los ejes. Cuando el usuario utiliza una herramienta, el movimiento rotacional se puede dividir en tres tipos de giros: yaw, pitch y roll. El primero hace referencia al giro sobre el eje Y, el segundo al giro sobre el eje X y el tercero a la rotación sobre el eje Z. En

² Por cinestesia se entiende la percepción del equilibrio y de la posición de las partes del cuerpo. Proporciona la información debida al movimiento de los miembros del cuerpo.

términos de ergonomía, los movimientos del brazo y la muñeca al hacer los giros en pitch y yaw son diferentes al que se hace en roll ya que éste último suele realizarse de forma independiente a los otros dos, por este motivo se elige el roll como giro a reemplazar (Figura 7.6).

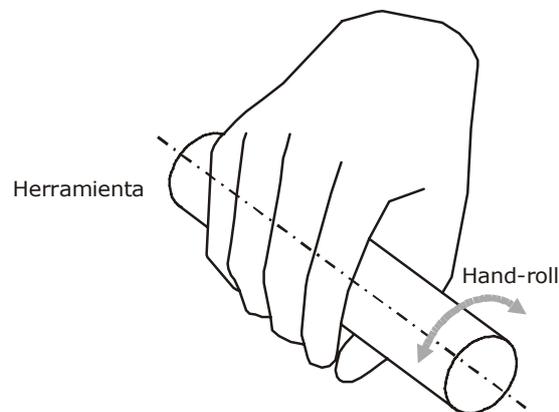


Figura 7.6: Rotación hand-roll.

Por lo tanto, el diseño alternativo para la muñeca háptica consiste en dotar al mecanismo de dos giros activos (pitch y yaw). El giro eliminado mecánicamente se sustituye por un sensor de fuerza y por una combinación de estímulos visuales y táctiles con el objetivo de recrear una ilusión cinestética en el eje de giro eliminado. Esta estrategia alternativa es lo que se ha denominado *aproximación de 2,5 gdl*.

Para conseguir transmitir la impresión de 3 gdl activos con sólo 2 gdl actuados, es necesario que la alternativa sensorial sustituya este último grado de libertad sea lo suficientemente explícita y clara para que el usuario asocie rápidamente esta respuesta con la que sentiría con el giro en roll actuado (Lécuyer et al. 2000a). La Figura 7.7 muestra un ejemplo del proceso que lleva al usuario a creer que está recibiendo una respuesta táctil mediante el dispositivo háptico utilizando para ello estímulos visuales y táctiles (en lo sucesivo, estímulos visuotáctiles).

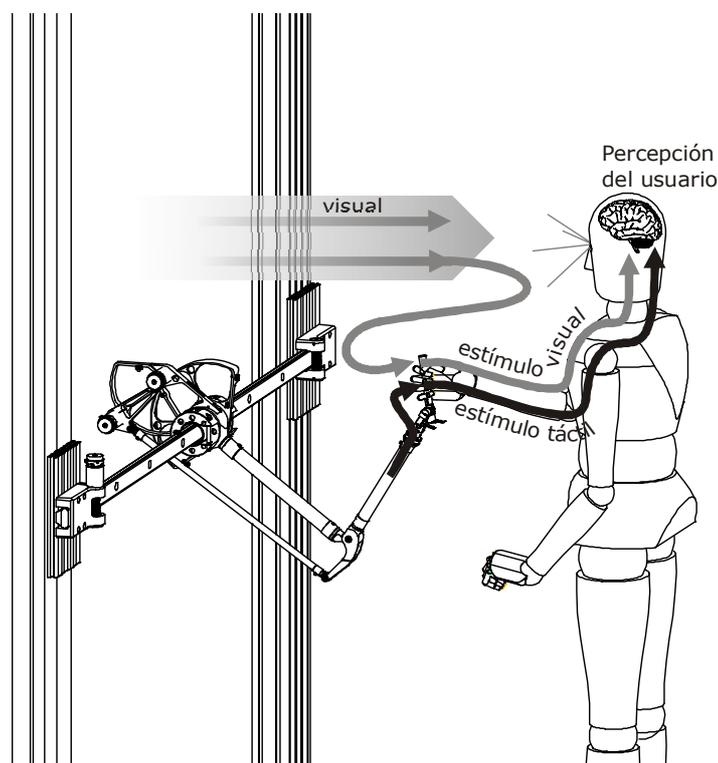


Figura 7.7: Háptico e ilusión háptica.

7.2.2.2 DISEÑO E IMPLEMENTACIÓN DEL DISPOSITIVO CON 2,5 GRADOS DE LIBERTAD

Los estímulos visuotáctiles diseñados siguen los criterios establecidos por (Massie and Salisbury 1994) que establece que un dispositivo háptico es efectivo cuando se puede sentir total libertad en el movimiento libre, y cuando los objetos virtuales se sienten rígidos.

El modo de funcionamiento del dispositivo es diferente dependiendo de si se produce una colisión con el entorno o no. Si el movimiento es libre, cuando el usuario trata de girar en el eje roll, aunque este giro no esté activo, ejerce un par (T) que es medido por el sensor de fuerza. Este giro aunque no sea real se traduce en una rotación visual de la herramienta en el entorno virtual con una velocidad ω proporcionando al usuario la sensación de que está ejerciendo un giro en un eje que realmente no está activo (Figura 7.8).

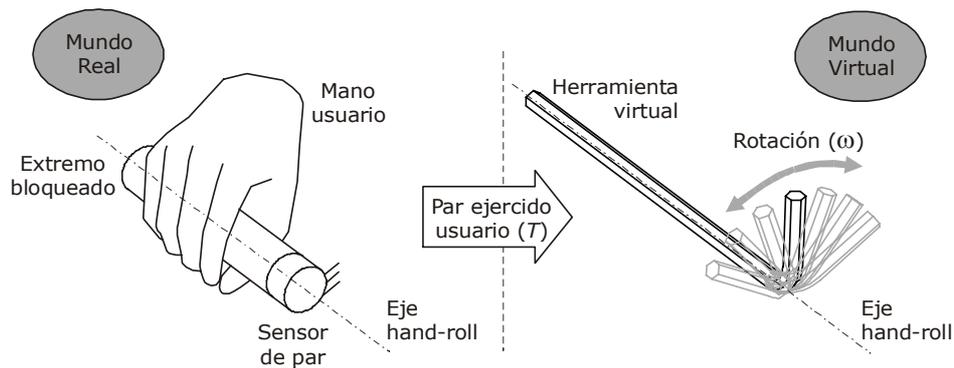


Figura 7.8: Movimiento libre en el giro hand-roll en el mundo real que se traduce al mundo virtual mediante una rotación de la herramienta virtual en este eje.

La Ecuación (7.1) muestra que la velocidad de rotación de la herramienta virtual sobre el eje hand-roll (ω) es proporcional (a) a una potencia (c) del par ejercido por el usuario y medido por el sensor (T), siendo ($\text{signo}(T)$) el signo del par medido. A medida que c se incrementa, la manipulación de precisión produce rotaciones más lentas y la manipulación de fuerza produce rotaciones más rápidas. Además, ω está limitada a un valor máximo coherente (ω_{\max}) para no perder realismo.

$$\omega = a |T|^c \text{signo}(T), \quad \text{donde } |\omega| \leq \omega_{\max} \quad (7.1)$$

Los parámetros de esta ecuación necesitan ser ajustados experimentalmente dependiendo del dispositivo háptico utilizado para poder lograr una percepción lo más parecida posible entre el gdl háptico y el gdl pseudo-háptico. Los valores utilizados en los experimentos realizados se muestran en la Tabla 7.1.

Movimiento Libre	
a	0,1 rad/(sNmb)
c	3
ω_{\max}	6 rad/s
Colisión	
p	0,5 Nm
Δ	0,006s

Tabla 7.1: Valores de los parámetros utilizados en la modalidad de 2,5 gdl.

El segundo modo de funcionamiento se da cuando se produce una colisión con el entorno virtual. En este caso, para proporcionar una mayor sensación de realismo se evita visualmente la interpenetración de la herramienta virtual en el entorno. Además, el usuario percibe un estímulo táctil para proporcionarle una sensación de rigidez del entorno virtual y evitar que siga penetrando en él. Este estímulo táctil que se percibe en el momento de la colisión (t_c) consiste en un único pulso de pequeña magnitud (p) de breve duración (Δ), generado mediante los actuadores de los gdl pitch y yaw. Así las colisiones debidas al movimiento pitch-yaw producen restitución cinestésica y las colisiones debidas a la rotación hand-roll producen restitución táctil. La Figura 7.9 muestra los dos tipos de restitución (visual y táctil) que recibe el usuario tras ejercer el giro en roll.

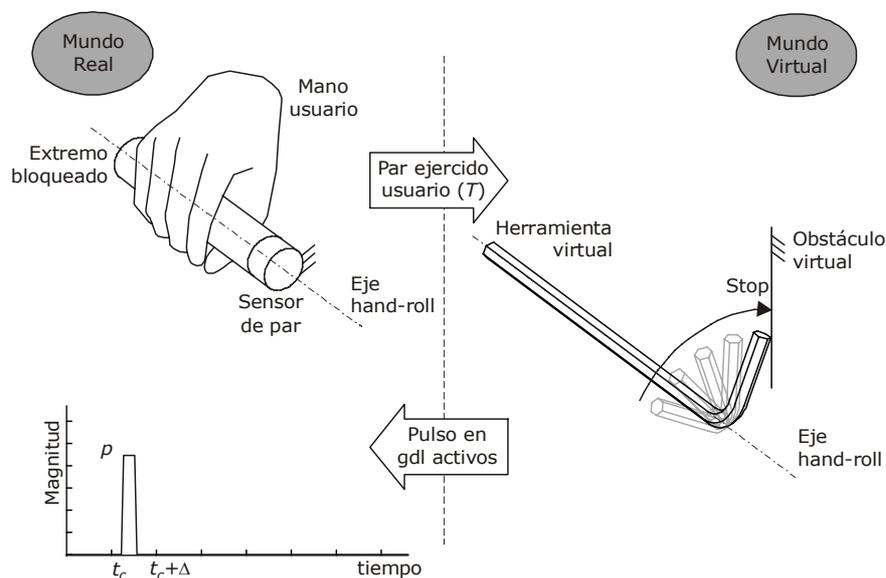


Figura 7.9: Estímulo táctil y visual proporcionado en el caso de la colisión con objetos del entorno virtual.

Las colisiones en las que se ven implicados los 3 gdl producen ambos tipos de restitución superpuestos. Debido a la breve duración y pequeña magnitud del pulso, la restitución táctil no afecta a la restitución cinestésica cuando ambas se superponen. Además, no es necesario un actuador adicional para la restitución táctil debido a que toda la restitución se lleva a cabo por los actuadores pitch-yaw. Finalmente, debido a que el usuario no es capaz de percibir la dirección del pulso, éste puede ser generado por un mismo actuador y con la misma magnitud, dirección y duración para todas

las colisiones, simplificando así todo el proceso. De nuevo, los parámetros del pulso necesitan ser ajustados experimentalmente en función del dispositivo mecánico y la aplicación.

La Figura 7.10 muestra un ejemplo de la respuesta que percibe el usuario en movimiento libre y en un estado de colisión. Para simplificarla únicamente se ha representado la colisión en el grado de libertad pseudo-háptico. Está dividida en tres partes: en la parte superior se representa el par ejercido por el usuario (T) en el tiempo t , en la zona central se representa la rotación virtual de la herramienta que viene dada por la Ecuación (7.2) y por último, la parte inferior muestra el par que ejerce el actuador de pitch (τ_p).

$$\varphi = \varphi_0 + \omega(T)t \quad (7.2)$$

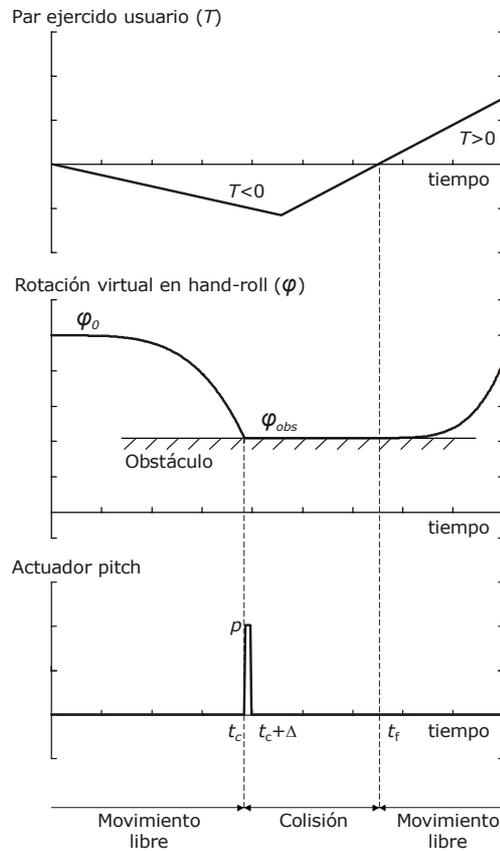


Figura 7.10: Ejemplo de interacción en espacio libre y colisión debido a rotaciones en hand-roll. De arriba abajo: Simulación del par ejercido por el usuario, rotación virtual resultante en hand-roll y estímulo táctil restituído al usuario.

En el estado inicial, el usuario percibe el movimiento libre, φ evoluciona a medida que éste ejerce un par T en el extremo al intentar girar ($T < 0$, así que φ decrece). Cuando la herramienta virtual colisiona con el entorno en el instante t_c ($\varphi \leq \varphi_{obv}$), un pulso es enviado al motor pitch y se evita la interpenetración visual entre la herramienta y el obstáculo. Cuando el usuario percibe que ha colisionado con el entorno, comienza a cambiar la dirección del par que está aplicando con su mano. Mientras el par siga siendo negativo, implica que el usuario se mantiene en un estado de colisión y como consecuencia la interpenetración visual se mantiene como se observa en la figura en el intervalo de tiempo comprendido entre $t_c + \Delta$ y t_f . Por último, en el instante de tiempo t_f se produce un cambio en el signo del par (signo (T)) lo que indica que el usuario cambia la dirección del par aplicado devolviéndole a un estado de movimiento libre o no colisión.

7.2.2.3 LIMITACIONES DE PSEUDO-HAPTICS

La propuesta planteada presenta dos limitaciones principales. La primera de ellas es que el giro pseudo-háptico proporciona la capacidad de detectar las colisiones con el entorno pero sin embargo no permite discriminar la rigidez. Debido a que mecánicamente el giro en hand-roll no es posible, la rigidez rotacional de dicho eje es percibida como infinita por el usuario. Por lo tanto, su validez dependerá de la aplicación para la que se vaya a utilizar el dispositivo háptico.

La segunda limitación se debe a que el mango del dispositivo háptico y el eje longitudinal de la herramienta virtual deben estar alineados, por lo que la solución tendrá un mejor comportamiento cuando las herramientas utilizadas sean alargadas. Antes de llegar a la propuesta final, se plantearon las posibilidades de aplicar el gdl pseudo-háptico a los giros en yaw y pitch en vez de al giro en roll, sin embargo ninguna de las opciones fue satisfactoria. Introducir un gdl pseudo-háptico implica necesariamente una pérdida de correspondencia entre los movimientos del usuario y los de la herramienta virtual. Esta pérdida no es tan severa en el eje de hand-roll, y el usuario mantiene un alto grado de realismo del sistema. Sin embargo, en cualquiera de los restantes gdl, al usuario le cuesta encontrar una relación entre los movimientos de su mano y los de la herramienta virtual.

7.2.3 VALIDACIÓN DE LA ESTRATEGIA

Para validar el concepto de un dispositivo háptico con 2,5 gdl, se ha realizado un experimento en el que se comparan tres modalidades de

restitución. El objetivo del experimento consiste en medir la efectividad de la estrategia propuesta como alertador de que se ha producido una colisión con el entorno y que éste reaccione. Las tres modalidades son las siguientes:

- Sistema de 2 gdl activos: Esta modalidad representa el peor escenario posible en la comparación. En este caso únicamente los giros pitch y yaw están actuados. La rotación hand-roll es físicamente posible, se mide pero no está actuada.
- Sistema de 2,5 gdl: propuesta a validar. Los giros pitch y yaw están actuados y el giro en roll está bloqueado pero se mide mediante un sensor de par.
- Sistema completo de 3 gdl activos y actuados. Es el mejor caso posible y por lo tanto representa el estándar a alcanzar.

7.2.3.1 DISEÑO DEL EXPERIMENTO

Para diseñar el experimento que permita validar el concepto de un sistema con 2,5 gdl se han seguido los siguientes criterios. En primer lugar, es necesario que la tarea a desempeñar por los usuarios implique giros de forma que se obtenga una restitución de pares. En segundo lugar, es necesario determinar el parámetro a utilizar para medir la efectividad del sistema como avisador de una colisión. La respuesta proporcionada por el sistema también debe proporcionar al usuario un criterio efectivo para abandonar el estado de colisión. Por último, la tarea y el entorno a utilizar en la prueba deben ser lo suficientemente sencillos para poder validar la hipótesis del experimento sin confundir a los usuarios que a priori no están familiarizados con este tipo de sistemas hápticos ni con el concepto de pseudo-háptica.

Se ha considerado que el escenario y el sistema descritos en el apartado 6.6 cumplen estos requisitos mencionados, y que por lo tanto es un entorno apropiado para validar la estrategia de 2,5 gdl.

7.2.3.2 USUARIOS Y PROCEDIMIENTO

En el experimento han tomado parte un total de 11 sujetos, 8 hombres y 3 mujeres con edades comprendidas entre los 25 y 32 años. Ninguno de los participantes presentaba anomalía alguna en la visión ni en el sentido del tacto. Además, la mayoría de los sujetos no tenía experiencia previa con dispositivos hápticos y todos ellos eran ajenos a los detalles de la hipótesis a

validar a través de las pruebas para evitar condicionar sus estrategias al llevar a cabo cada una de las pruebas y por lo tanto que pudiesen favorecer los resultados esperados.

La tarea a realizar consiste en mover la abrazadera de un extremo a otro hasta conseguir sacarla de la tubería. A lo largo de este recorrido, el participante debe evitar la colisión con el entorno, pero en caso de que se produzca, debe abandonar este estado de colisión en el menor tiempo posible con la ayuda del feedback correspondiente en cada modalidad. Los usuarios disponen de un tiempo ilimitado para realizar dicha tarea, ya que el tiempo total para completar el recorrido no se ha considerado relevante para el objetivo del experimento.

La tarea se repite tres veces una para cada modalidad diferente de restitución de par proporcionada: 3gdl, 2,5 gdl y 2 gdl. Además de los grados de libertad en los giros, en cada una de las pruebas la guía lineal permanece actuada por lo que el sistema proporciona 4, 3,5 y 3 gdl actuados respectivamente. Las diferentes modalidades de feedback se muestran en un orden aleatorio para cada usuario y con un intervalo de tiempo suficiente para evitar así un posible aprendizaje que pueda distorsionar los resultados obtenidos. Antes de comenzar el experimento, se permite a cada participante un breve entrenamiento para que se familiarice con el entorno virtual, el dispositivo háptico y la modalidad de feedback que va a utilizar. Además, se les muestra una demo de la extracción de la abrazadera antes del primer experimento para una mejor comprensión de la tarea a realizar.

El entorno virtual en el que se realizan las pruebas se ha diseñado de tal forma que resulte difícil evitar totalmente la colisión, obligando al usuario a utilizar el sentido del tacto para poder cumplir con la tarea propuesta.

El objetivo del experimento consiste en comprobar la validez de la estrategia de 2,5 gdl para avisar al usuario de que se ha producido una colisión además de proporcionar un criterio efectivo para salir de ésta. El parámetro utilizado para evaluar la eficacia de las tres modalidades de feedback es el tiempo de reacción medio a lo largo de la prueba. El tiempo de reacción hace referencia al intervalo de tiempo transcurrido desde que el usuario entra en un estado de colisión, y finaliza cuando logra salir de dicha colisión. Por lo tanto, para cada test y usuario, se mide el tiempo medio de reacción como el ratio entre el tiempo total en colisión y el número de colisiones obtenidas en el proceso de extracción. Puesto que la única diferencia entre las tres modalidades de feedback propuestas es la

restitución en el eje roll, únicamente se tienen en cuenta las colisiones debidas a esta rotación para determinar el tiempo medio de reacción para cada prueba.

7.2.3.3 RESULTADOS

La Figura 7.11 muestra los tiempos medios de reacción en segundos obtenidos por cada participante en cada una de las pruebas realizadas. En base a estos datos, también se ha calculado el tiempo medio para cada tipo de feedback y su desviación estándar. Estos resultados se muestran visualmente mediante un diagrama de cajas que permite una comparación a simple vista de la distribución de los valores obtenidos para cada modalidad.

Tras un análisis de los resultados obtenidos, se puede concluir que la propuesta basada en 2,5 gdl mejora considerablemente el rendimiento del sistema en comparación con los resultados obtenidos con la modalidad de 2 gdl en giros, pero no alcanza la efectividad de los 3 gdl.

Sujeto	2 gdl	2,5 gdl	3 gdl
1	2,116	0,803	0,514
2	1,736	0,446	0,251
3	1,308	1,030	0,648
4	1,419	1,282	0,286
5	1,590	1,092	0,165
6	0,979	1,009	0,544
7	0,736	0,799	0,169
8	0,946	0,691	0,209
9	2,430	0,403	0,327
10	1,988	1,045	0,393
11	1,836	0,933	0,759
Media	1,553	0,866	0,387
Desv. Est.	0,532	0,271	0,202

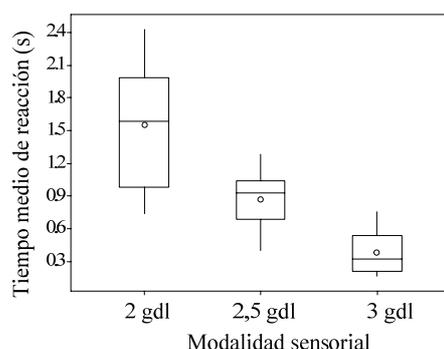


Figura 7.11: Tiempo medio de reacción (s) para cada prueba, y valor medio y desviación estándar en cada modalidad y diagrama de cajas con los datos (las líneas dentro de las cajas representan las medianas y los círculos las medias).

Como muestra el diagrama de cajas, el rango de valores de los resultados obtenidos con un sistema de 2 gdl es bastante más amplio que en las dos modalidades restantes. En esta modalidad el usuario debe detectar que se ha producido una colisión con la única ayuda de su vista y corregir su posición cuando y cómo lo considere oportuno para salir de este estado de colisión. El hecho de no recibir ningún estímulo táctil como respuesta, hace menos intuitivo para el usuario saber cuándo comienza o termina un estado de colisión, provocando una mayor variación del tiempo medio de reacción. Por el contrario, se observa que el tamaño de las cajas en

el resto de modalidades es menor, indicando una distribución más uniforme de los resultados obtenidos. De este dato, se deduce que las dos modalidades restantes proporcionan una respuesta rápida y efectiva a una colisión en la mayoría de los casos ya que sus tiempos de reacción son menores.

Para encontrar diferencias estadísticamente significativas, se ha realizado un análisis ANOVA de dos factores, sujeto y modalidad de restitución. Previamente, se ha aplicado una transformación Box-Cox para cumplir con los requisitos de normalidad y homogeneidad de la varianza de la distribución. Los resultados del test muestran que la modalidad de restitución sí tiene un efecto estadísticamente significativo sobre el tiempo medio de reacción ($F_{2,30}=34,65$, $p<0,001$), mientras que no se observa tal efecto con los sujetos ($F_{10,22}=1,19$, $p=0,354$). Se ha realizado un test post-hoc de Tukey a los resultados, donde se observa que existen diferencias significativas entre todas las modalidades de restitución.

La modalidad de 2,5 gdl reduce el tiempo medio de reacción en un 44% con respecto a la de 2 gdl, mientras que en el caso de la restitución completa de 3 gdl esta reducción es de un 75%. Esto significa que la aproximación de 2,5 gdl propuesta está aproximadamente a medio camino entre el peor caso de 2 gdl y el ideal de 3 gdl que se quería alcanzar.

Además del análisis estadístico, se han tenido en cuenta los comentarios y el procedimiento de los usuarios al realizar cada una de las pruebas. Una de las observaciones más interesantes fue que en la modalidad de 2,5 gdl, en el movimiento en espacio libre, los usuarios mencionaban que tenían la sensación de poder girar su mano en el extremo del dispositivo, cuando esta rotación no era mecánicamente posible. Se ha observado que la deformación de la piel en los puntos de contacto entre la mano y el extremo permite un cierto giro de la mano alrededor del extremo, lo que ayuda a las estrategias pseudo-hápticas a generar en el sujeto la percepción de movimiento libre. En cuanto a la percepción de los eventos de colisión en la modalidad de 2,5 gdl, los usuarios manifestaron sentir una sensación convincente y realista cuando se les presentaban los estímulos visuales y táctiles.

7.2.3.4 CONCLUSIONES

El uso de los dispositivos hápticos y todo lo relacionado con el renderizado háptico es todavía una técnica reciente pero que cada vez tiene mayor aceptación para su aplicación en diferentes áreas como la medicina o la

industria. Al tratarse de una técnica tan reciente, todavía hay muchas situaciones en las que resulta difícil simular fielmente la realidad ya sea por limitaciones del hardware o del software.

En este caso, para suplir las limitaciones del diseño mecánico del dispositivo, se ha analizado la validez de una técnica de sustitución sensorial para reemplazar el comportamiento del giro activo en roll por un gdl pseudo-háptico con una combinación de estímulos visuales y táctiles.

Para ello se ha realizado un experimento multiusuario que compara la aproximación de 2,5 gdl con estímulos visuales y táctiles, con la restitución de momentos de 2 y 3 gdl respectivamente. El estudio experimental demuestra que la solución pseudo-háptica no alcanza la efectividad de la restitución completa de pares en 3 gdl, pero mejora de forma significativa la restitución de pares con sólo 2 gdl.

Teniendo en cuenta las ventajas y limitaciones de la estrategia pseudo-háptica, se puede concluir que dependiendo de la aplicación en la que se vaya a utilizar, esta simplificación puede compensar un pequeño decremento de la efectividad en comparación con un sistema de 3 gdl en favor de una reducción de costes y un diseño más ligero. En cualquier caso, la solución estudiada introduce nuevas estrategias de diseño basadas en técnicas de sustitución sensorial.

CAPÍTULO 8

CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN

8.1 CONCLUSIONES

La presente Tesis aporta soluciones en dos áreas fundamentales de la háptica: el renderizado háptico y la percepción multisensorial.

En primer lugar se ha implementado un método de renderizado háptico capaz de calcular fuerzas y pares en tiempo real derivadas de la interacción del usuario con los objetos de los escenarios virtuales. El método calcula una respuesta estable en entornos complejos con discontinuidades, restituyendo al usuario una sensación háptica agradable y adecuada. Las principales aportaciones en este aspecto son las siguientes:

- Se ha modificado el algoritmo de colisiones desarrollado en el CEIT, para adecuarlo a la precisión en la respuesta háptica, identificando los triángulos necesarios para definir el volumen de intersección generado entre los dos objetos en contacto. Esta información es la que utilizan los algoritmos de respuesta de colisión para calcular la respuesta háptica a restituir al usuario.
- Se realiza una clasificación de los triángulos en contacto en base a áreas y zonas, que permite calcular una normal de contacto adecuada en todos los casos incluidos las colisiones complejas. Además, las variaciones de esta dirección se realizan de forma progresiva aún con la aparición de discontinuidades geométricas en los modelos del entorno virtual.

- Se han planteado diversas alternativas para el cálculo de la penetración. En primer lugar se realizan mejoras al método de Hippmann, utilizado en simulaciones dinámicas. También se plantea una solución propia que realiza un muestreo del volumen de contacto en intersección mediante rayos. Una característica significativa de este método es su alto grado de paralelización, lo que le permite poder sacar partido de la arquitectura multicore.

Por último se ha analizado una alternativa que utiliza la GPU. Este último método tiene una precisión elevada aunque su coste computacional y su bajo grado de paralelización no permiten que se obtengan buenos resultados en cuanto a la percepción debido a los retrasos. Sin embargo, su precisión permite utilizar este método como referencia al hacer la comparativa de los resultados de los diferentes métodos utilizados.

- Para validar la respuesta háptica calculada se ha realizado un análisis de los diferentes factores que afectan a la estabilidad de la respuesta háptica, tales como la normal, penetración y punto de contacto. Las variaciones bruscas tanto de la dirección de la fuerza y/o par como de su magnitud provocan inestabilidades en la respuesta final.

Estas variaciones son más frecuentes en situaciones de colisión complejas como las discontinuidades geométricas o contactos múltiples. Por este motivo, se ha realizado un estudio del comportamiento de las diferentes alternativas en estas situaciones. Los resultados muestran que con el método de rayos se obtiene una respuesta háptica estable sin variaciones bruscas ni en su módulo ni dirección, proporcionando al usuario una respuesta suave y agradable.

- Se ha realizado una comparativa de los resultados obtenidos con el método propuesto basado en rayos y el método basado en la GPU por su precisión en los cálculos. También, se ha realizado una comparativa con el método mejorado de Hippmann y el método desarrollado por Ortega. Este estudio permite demostrar que el método propuesto en este trabajo calcula una respuesta háptica con un mejor compromiso entre la estabilidad y el coste computacional, gracias en parte a su alta capacidad de paralelización.

- Para mejorar el realismo de la simulación y aumentar la inmersión del usuario dentro del entorno se ha añadido al sistema la propiedad de restitución visual. Esta propiedad aumenta la sensación de rigidez del entorno ya que visualmente se evita que el objeto móvil penetre en el objeto estático.
- Los experimentos realizados se han llevado a cabo desacoplando el problema de fuerzas y pares debido a la falta de disponibilidad de un dispositivo háptico de 6 gdl. Para intentar superar esta limitación y realizar un experimento que simulase una tarea real que además de rotaciones incluyese traslaciones, se ha implementado un sistema con 3 gdl en rotación y uno en traslación. Aunque no se consigue validar el método propuesto en un entorno completo de 6 gdl, los resultados obtenidos con 4 gdl permiten intuir que el método obtendría resultados similares con un dispositivo de 6 gdl.

Un buen método de renderizado permite que el usuario de un sistema háptico obtenga una buena información táctil de los eventos de colisión. Sin embargo, la mayoría de interacciones en entornos de la vida real implica la combinación de múltiples canales sensoriales para la ejecución de tareas. Por ello, una segunda área en la que se ha trabajado es en el estudio de la integración de distintos canales sensoriales en los sistemas hápticos:

- Se ha estudiado la influencia de combinar información proveniente de distintos canales sensoriales en la consecución y el realismo de una tarea virtual de accesibilidad. De este estudio se demuestra que la sinergia de los canales sensoriales visual, auditivo y háptico puede mejorar la eficiencia a la hora de desarrollar una tarea de accesibilidad si estos canales están debidamente sincronizados.
- Se ha analizado la validez de una estrategia de sustitución sensorial para reemplazar el comportamiento de un gdl de un dispositivo háptico. Esta técnica tiene como objetivo simplificar el diseño mecánico de una muñeca háptica por un gdl pseudo-háptico. Para suplir este gdl se ha diseñado una combinación de estímulos visuales y táctiles para dar al usuario la impresión de tener 3 gdl activos con tan sólo 2 gdl.

Se ha realizado un experimento multiusuario que compara la estrategia propuesta con la opción de tener 3 gdl mecánicamente

activos. Los resultados muestran que la solución adoptada no alcanza el realismo de los 3 gdl activos pero mejora de forma significativa la restitución de pares con sólo 2 gdl activos.

8.2 FUTURAS LÍNEAS DE INVESTIGACIÓN

Como continuación del trabajo descrito en esta Tesis, en los siguientes párrafos se proponen algunas de las posibles líneas en las que se podría seguir investigando:

Respecto al desarrollo de métodos de renderizado háptico:

- Estudiar el comportamiento del método propuesto en un dispositivo háptico de 6 gdl.

La solución propuesta para el cálculo de fuerzas y momentos estables en respuesta a una colisión, se ha estudiado de forma desacoplada en dos dispositivos hápticos diferentes por no disponer de un sistema háptico completo. Sería interesante realizar un estudio del comportamiento del método en un dispositivo de 6 gdl para demostrar que la solución aportada es general.

- Estudio de la teselación óptima a utilizar para definir los modelos del entorno.

La solución propuesta depende en gran medida de la teselación de los modelos utilizados. Cuanto menor sea el tamaño de los triángulos que definen el mallado, mayor será la precisión obtenida en el cálculo de la penetración. Sin embargo, esta teselación fina incrementa el coste computacional. Sería necesario realizar un estudio para calcular la teselación óptima de los modelos para llegar a un compromiso entre la precisión y el coste computacional que permita garantizar la estabilidad de la respuesta háptica calculada.

- Implementar el método de renderizado para que pueda ser utilizado en contactos con objetos deformables.

El método de renderizado desarrollado en este trabajo proporciona una respuesta estable tras una colisión, pero se limita a objetos virtuales rígidos. Una posible mejora consistiría en implementar el método para que pudiese calcular también contactos con objetos deformables, ampliando considerablemente

de esta manera el rango de aplicaciones en las que puede utilizarse.

- Desarrollar nuevos algoritmos de renderizado para poder crear aplicaciones con escenarios dinámicos.

En el método actual, el único objeto móvil en el escenario virtual es el controlado por el usuario. Los restantes objetos del escenario son siempre estáticos. Tanto para objetos rígidos como deformables, sería conveniente poder trabajar con escenarios dinámicos.

- Desarrollar nuevos modelos de fuerzas para que puedan sentirse texturas.

Los modelos de contacto utilizados no contemplan la restitución háptica de texturas, tan sólo pueden representar el efecto del rozamiento al recorrer una superficie. La restitución táctil de texturas en los objetos virtuales puede mejorar significativamente la sensación de realismo en tareas de exploración.

En el área de la percepción multisensorial:

- Estudiar el efecto de combinar múltiples canales sensoriales en los tiempos de ejecución de las tareas virtuales.

Una de las conclusiones de la Tesis sugiere que al emplear distintas modalidades sensoriales para representar la misma información al usuario, éste se concentra más en el evento, pero reacciona de manera más lenta. Este hecho debería analizarse más detalladamente.

- Estudiar el comportamiento del grado de libertad pseudo-háptico en el LHIFAM.

A partir del estudio realizado sobre grados de libertad pseudo-hápticos, falta por validar la solución propuesta en el LHIFAM, y comprobar que las conclusiones obtenidas siguen siendo válidas. En tal caso, se debe estudiar la aplicación de estrategias de sustitución sensorial en futuros diseños hápticos.

ANEXO A

DISPOSITIVOS HÁPTICOS

En este anexo se presentan los diferentes dispositivos hápticos que han sido utilizados para validar los resultados presentados en este trabajo.

A.1 PHANTOM PREMIUM 1.0

El Phantom es el único dispositivo comercial que se ha utilizado en este trabajo (Figura A. 1). La compañía *Sensable*³ se encarga de fabricar y comercializar este dispositivo cuyo acrónimo viene de *Personal HAptic iNTerface Mechanism*, aunque el dispositivo original fue diseñado y construido en 1990 por Massie et al. (1994).

Este dispositivo tiene seis grados de libertad sensorizados pero sólo tres de ellos están sensorizados, lo que permite únicamente captar movimientos en los tres ejes cartesianos. La Tabla A. 1 recoge sus principales características técnicas.

³ www.sensable.com



Figura A. 1: PHANTOM Premium 1.0.

Grados de libertad	
Sensorizados	6 gdl (x, y, z, Pitch, Yaw, Roll)
Actuados	3 gdl (x, y, z)
Espacio de trabajo	19,5 x 27 x 37,5 cm
Fuerza máxima (pico)	8,5 N
Fuerza máxima (continua)	1,4 N
Rigidez	3,5 N/mm
Resolución	< 0,03 mm
Fricción estática	0,04 N
Inercia aparente	< 75g

Tabla A. 1: Principales características técnicas del PHANTOM *Premium 1.0*.

A.2 PALANCA DE 3 GRADOS DE LIBERTAD

La palanca háptica desarrollada por el CEIT (Figura A. 2) está inspirada en un trabajo anterior desarrollado en Percro (Angerilli et al. 2001) para la simulación de una palanca de cambios con reflexión de fuerzas. Se trata de un dispositivo de bajo coste especialmente diseñado para ser utilizado en simuladores de conducción.

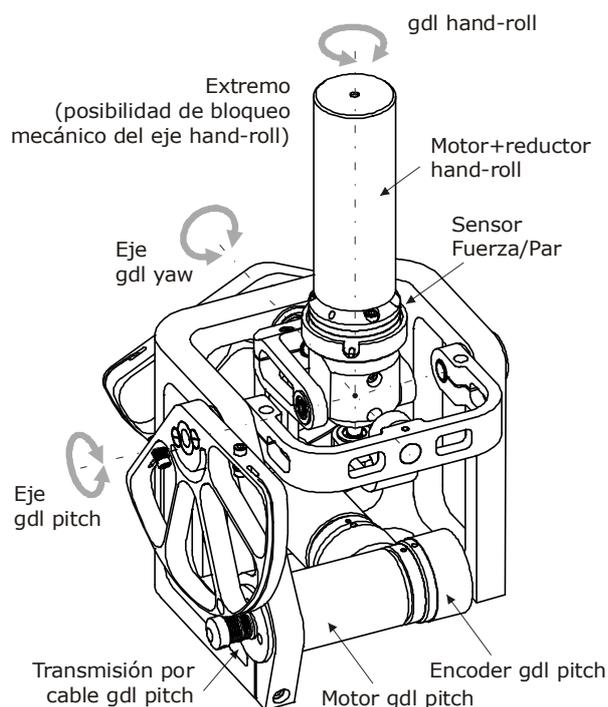


Figura A. 2: Palanca del CEIT de 3 gdl.

Este dispositivo tiene 3 gdl en giros, concretamente 2 de sus gdl (pitch/yaw) son paralelos y están actuados por motores Maxon comerciales y transmisiones de cable especialmente diseñadas. El último gdl está acoplado en serie y es actuado mediante un motor y un reductor Maxon. La rotación en cada uno de los ejes se mide mediante un encoder acoplado a cada motor.

El diseño del dispositivo permite bloquear el giro alrededor del mango (roll), convirtiéndose en un dispositivo de 2 gdl. Si se añade un sensor de fuerza/par que mida el par que ejerce el usuario cuando agarra el mango del dispositivo, se consigue un dispositivo de 2,5 gdl, ya que se mide el giro en el eje roll aunque no se encuentre mecánicamente activo.

Este dispositivo se controla mediante una tarjeta dSPACE DS1104 que lee la información de los encoders, procesa el lazo de control háptico y envía los pares a restituir a los motores.

Las características del dispositivo se muestran en la Tabla A. 2.

Grados de libertad	
Medidos	3 dof (Pitch-Yaw-Roll)
Actuados	3 dof (Pitch-Yaw-Roll)
Espacio de trabajo	
Pitch-Yaw	+/-70°
Roll	Ilimitado
Par máx. continuo	
Pitch-Yaw	0,7 Nm
Roll	0,2 Nm
Par máx. pico	
Pitch-Yaw	1,6 Nm
Roll	0,9 Nm
Actuadores	
Pitch-Yaw	DC MaxonRE35, 90 W
Roll	DC MaxonRE-max21, 6 W
Encoders	
Pitch-Yaw	QuantumDevices QD145, 5000
Roll	MaxonMR TypeM, 512 ppr
Sensor Fuerza/Par	
Modelo	ATI Mini 80/4
Rango de fuerza	80 N (x,y), 240 N (z)
Rango de par	4 Nm
Resolución fuerza	0.08 N (x,y), 0.24 N (z)
Resolucion par	2.0 Nmm

Tabla A. 2: Especificaciones de la palanca del CEIT de 3 gdl.

A.3 GUÍA LINEAL

Para proporcionar un gdl en traslación a la palanca háptica del apartado anterior, se utiliza una guía lineal diseñada y desarrollada por el Ceit (Savall et al. 2008). Su diseño se muestra en la Figura A. 3.

Esta guía está actuada por un motor comercial Maxon y una transmisión por cable especialmente diseñada que convierte el movimiento rotacional del motor de la palanca en un desplazamiento traslacional de la parte móvil de la guía.

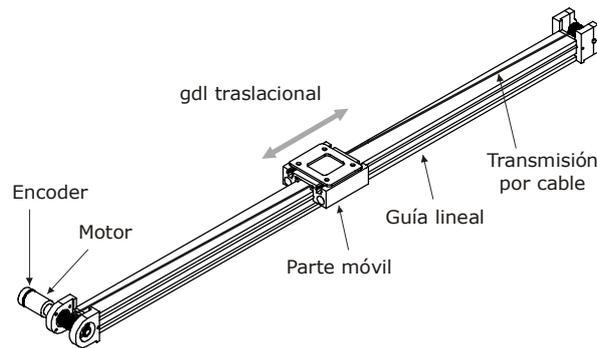


Figura A. 3: Guía lineal.

La Tabla A. 3 muestra las especificaciones de la guía lineal.

Grados de libertad	
Entradas sensorizadas	1 gdl (traslacional)
Salidas actuadas	1 gdl (traslacional)
Espacio de trabajo	1,018 m
Fuerza máx. continua	6,5 N
Fuerza máx. pico	21,8 N
Actuador	DC Max onRE35, 90W + Transmisión por cable
Encoder	HP HEDS-5500, 500 ppv

Tabla A. 3: Especificaciones de la guía lineal

A.4 LHIFAM

El reto principal del diseño de este dispositivo háptico ha sido el conseguir un espacio de trabajo con unas dimensiones tan grandes como una turbina aeronáutica. Esto se ha logrado gracias a que el dispositivo está montado sobre una guía que coincide con el eje longitudinal de lo que sería el motor.

El LHIFAM (Figura A. 4a) posee 6 gdl pero sólo 3 de ellos están actuados. El desplazamiento a lo largo del eje del cilindro del motor se consigue usando la guía lineal mientras que para conseguir los otros dos grados de libertad se monta sobre esa guía un mecanismo paralelogramo. En la Figura A. 4b se puede ver el dispositivo sobre su guía y al usuario realizando una operación de desensamblaje sobre el modelo virtual.

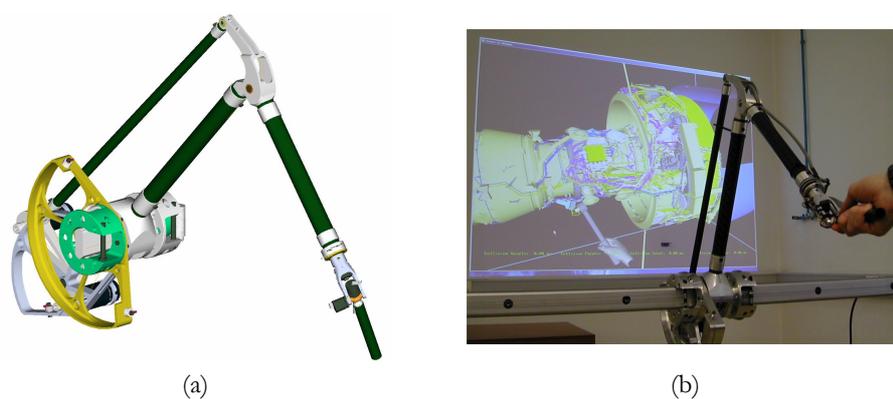


Figura A. 4: Modelo CAD del LHIFAM (a) y usuario manejando el LHIFAM con una maqueta virtual (b).

El espacio de trabajo corresponde con un sector cilíndrico aproximado al área de trabajo ocupada por una maqueta aeronáutica. Las dimensiones de este espacio cilíndrico son: radio interno, 242 mm; radio externo, 742 mm; longitud de 1500 mm y ángulo de 120° . Dicho espacio de trabajo se muestra en la Figura A. 5.

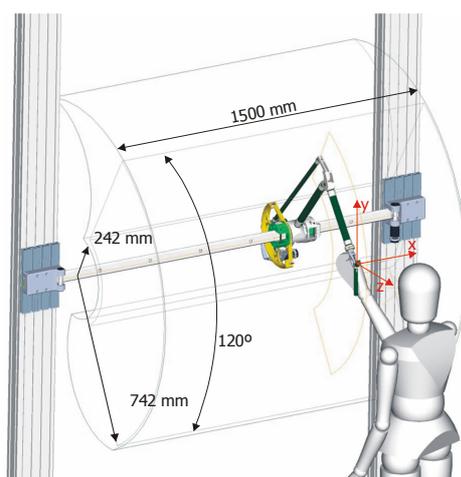


Figura A. 5: Dimensiones del espacio de trabajo del LHIFAM.

La ventaja del tamaño del dispositivo en cuanto al espacio de trabajo tiene como inconveniente la inercia que aparece cuando se intenta mover el háptico a lo largo de la guía lineal. Para compensar esa inercia se hace uso

de un sensor de fuerza gracias al cual se puede medir la fuerza que intenta hacer el usuario y de este modo ayudarle en esa traslación.

Para poder reproducir diferentes operaciones de mantenimiento y chequear distintas situaciones desde un punto de vista ergonómico, la interfaz háptica está montada en una estructura en la que se puede graduar la altura del dispositivo y además el sector cilíndrico puede ser reubicado en caso necesario. Básicamente se quiere simular las posiciones que se muestran en la Figura A. 6. Las especificaciones técnicas de este dispositivo háptico se enumeran en la Tabla A. 4.

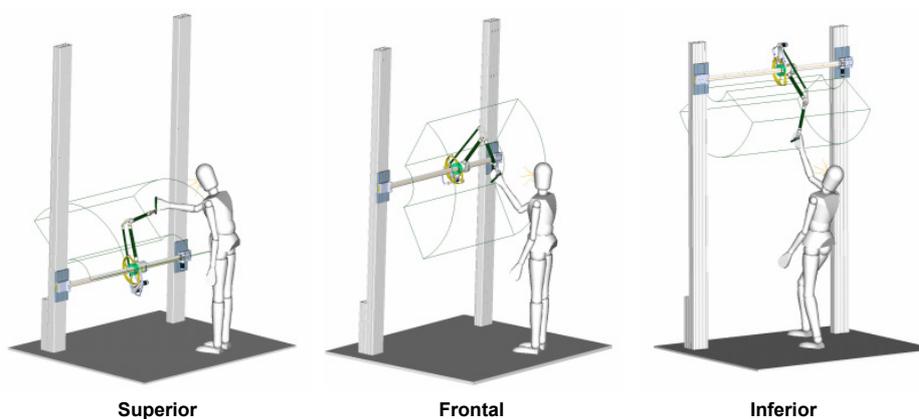


Figura A. 6: Diferentes reubicaciones del espacio de trabajo.

Grados de libertad	
Sensorizados	6 gdl (x, y, z, Pitch, Yaw, Roll)
Actuados	3 gdl (x, y, z)
Espacio de trabajo	120° x 500 x 1500 mm
Resolución en traslación eje X	0.004 mm
Resolución en traslación eje Y	0.025 mm
Resolución en traslación eje Z	0.012 mm
Resolución en rotaciones	0.0439°
Fuerza máxima eje X (pico)	42.5 N
Fuerza máxima eje Y-Z (pico)	72 N
Fuerza máxima eje X (continua)	10 N
Fuerza máxima ejes Y-Z (continua)	16 N
Inercia aparente eje X (sin compensar)	5 kg
Inercia aparente eje X (compensado)	0.90 kg
Inercia aparente eje Y-Z	< 1 kg m ²

Tabla A. 4: Especificaciones del LHIFAM

ANEXO B

PUBLICACIONES

En este anexo se presentan las publicaciones generadas hasta el momento gracias al estudio e investigación de la presente Tesis. A continuación se enumeran y se presenta la primera página de cada publicación.

Borro, D., Hernantes, J., Mansa, I., Amundarain, A., García-Alonso, A., and Matey, L., "Virtual Maintenance for Dense Environments", *Proceedings of the Laval Virtual, 7th International Conference on Virtual Reality 2005 (VRIC 2005)*, pp. 141-148. Laval, France. April 20-24, 2005.

Borro, D., Hernantes, J., García-Alonso, A. and Matey, L., "Collision problem: Characteristics for a Taxonomy", *Proceedings of the 9th International Conference Information Visualisation (IV'05)*, pp. 410-415. London, England. July 6-8, 2005.

Hernantes, J., Borro, D., Matey, L., and García-Alonso, A., "Analysis of Collision Detection in Dense Geometry Sets", *Proceedings of the 10th International Fall Workshop Vision, Modeling, and Visualization (VMV 2005)*, pp. 99-106. Erlangen, Germany. November 16-18, 2005.

Hernantes, J., Díaz, I., Álvarez, H., and Borro, D., "Stable Collision Response Method in Complex Interactions", *Poster Proceedings of the Third Workshop in Virtual Reality Interactions, and Physical Simulations (VRIPHYS'06)*, pp. 123-124. Móstoles, Madrid, Spain. November 6-7, 2006.

Díaz, I., Hernantes, J., Mansa, I., Lozano, A., Borro, D., Gil, J.J., and Sánchez, E., "Influence of Multisensory Feedback on Haptic

- Accessibility Tasks", *Virtual Reality, Special Issue on Multisensory Interaction in Virtual Environments*, Vol. 10, No. 1, Pp. 31-40. May 2006.
- Hernantes, J., Borro, D., and García-Alonso, A., "Stable Haptic Response for Complex Interactions", *Proceedings of the 2nd International Conference on Computer Graphics Theory and Applications (GRAPP'07)*, pp. 146-153. Barcelona, Spain. March 8-11, 2007.
- Martín, J., Savall, J., Díaz, I., Hernantes, J., and Borro, D. (2008). "Evaluation of Sensory Substitution to Simplify the Mechanical Design of a Haptic Wrist," *Proceedings of the IEEE International Workshop on Haptic Audio Visual Environments and Games*, pp. 132-136, Ottawa, Canada, October 17-18.
- Martín, J., Savall, J., Díaz, I., Hernantes, J., and Borro, D. (2009). "Haptic Wrists: An Alternative Design Strategy Based on User Perception". *Journal of Computing & information Science in Engineering (in press)*.
- Hernantes, J., Díaz, I., Borro, D., and Gil, J.J., "Effective Haptic Rendering Method for Complex Interactions", **submitted to** *Computer Graphics International 2009 (CGI'09)*. Victoria, British Columbia, Canada. May 26-29, 2009.

Virtual maintainability for dense environments

Borro, D.¹, Hernantes, J.¹, Mansa, I.¹, Amundarain, A.¹, García-Alonso, A.², and Matey, L.¹

¹CEIT and Tecnun (University of Navarra), Manuel Lardizabal 15, 20018 San Sebastián, Spain

²University of the Basque Country, Manuel Lardizabal 1, 20018 San Sebastián, Spain

Abstract

REVIMA (Virtual Reality for Maintainability) integrates different research areas: hardware, real time image generation, collisions, and control. It is a virtual reality system for maintainability simulation in Aeronautics based on a new haptic device. Through this system we track hand movements and provide force feedback within the dense geometric models that describe aircraft engines.

Keywords: Collision Detection, haptics, maintenance.

1. Introduction

This paper summarizes the latest developments in our work dealing with VR systems for maintenance simulation in dense environments: geometric models build up by a large amount of polygons –up to 2M–, contained in a “small” volume, like an aircraft engine. This paper provides a full view of the whole system, while more depth studies of specific problems will be referred to as the corresponding topic appears along these pages. It has been written as a full-contained document, so readers will be saved the effort of consulting previously published material [1, 2], although there can be found background complementary reading.

In the field of Aeronautics the term Maintainability is defined as “the ability of an element to keep in service or to be returned to the adequate status in order to develop its function, after being maintained at conditions previously established, using the personnel, the means and the adequate procedures”[3]. One of the most relevant aspects about maintainability is man and tool accessibility analysis, in order to analyze assembly-disassembly sequences and times.

Design based on electronic mock-up is widely used in engine design in the aeronautic industry. Those elements are installed by means of a workstation network. This allows a group of designers to work quasi-concurrently over an assembly. This technology is known in the industry as DMU/DPA (Digital Mock-Up / Digital Pre-Assembly).

DMU/DPA technology has overcome the need of a hard mock-up for design purposes, extremely decreasing the time to market and saving costs. However, nowadays the use of a physical mock-up is mandatory to evaluate engine maintainability during the developmental stage. Although these mock-ups can be used for other applications, like

training or demos, the ultimate purpose of the construction is to check the maintainability. The expenses derived from these mock-ups led ITP to research an alternative, using haptic devices.

The whole system has been built from scratch at the CEIT Applied Mechanical Department. This is a multidisciplinary development that includes, among others, the following disciplines: mechanical design, control theory, computer graphics, computational geometry and human-computer interaction.

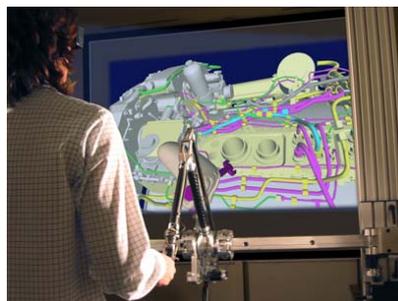


Figure 1 Virtual maintainability operation on a CAD aircraft engine.

The project required the development of a new tracking system with force feedback: the Large Haptic Interface for Aeronautic Maintainability (LHifAM) [4]. Figure 1 shows an example of a virtual operation, and offers an insight into the integration of the haptic interface and the visualization system.

The research involved in REVIMA includes two main issues: mechanical design and software development. One specific objective of our mechanical design is that the workspace of the device matches that of an aircraft engine, while

Collision problem: Characteristics for a Taxonomy

D. Borro, J. Hernantes	A. García-Alonso	L. Matey
Ceit	EHU	Tecnun (University of Navarra)
Manuel de Lardizábal 15	Manuel de Lardizábal 1	Manuel de Lardizábal 13
20018 San Sebastián, Spain	20018 San Sebastián, Spain	20018 San Sebastián, Spain

{dborro@ceit.es, jhernantes@ceit.es, alex.galonso@ehu.es, lmatey@tecnun.es}

Abstract

The Collision problem appears within many fields. The specific characteristics that can be identified in different problems lead to the broad set of specialized algorithms that appear in the literature. This paper deals with the first step needed to address the Collision Problem Taxonomy challenge: a survey that compiles and suggests a set of characteristics that could be used to discriminate collision problems, i.e. to generate the taxonomy.

1. Introduction

The Collision problem appears within many fields of interest like Geovisualization, Urban Walkthroughs, Robotics or Simulation. Surveys within this topic sometimes compile algorithms, others use applications as the compiling criterion. This survey has a different goal: it is the first step needed to address the Collision Problem Taxonomy challenge: a survey that compiles the characteristics that will be used to discriminate collision problems, i.e. to generate the taxonomy.

Although surveys and technical papers make reference to characteristics, there is a lack of systematization. Lin and Gottschalk [1] came near to this objective when they commented some characteristics: queries (see Section 2), pair vs. nbody (Section 3), static vs. dynamic (Section 4) and rigid vs. deformable.

A taxonomy is required because visualization systems are increasing the scope of problems they must deal with. For instance, geographical visualization is not limited to high altitude flights where the collision problem can nearly be neglected; low flying requires the integration of specific collision algorithms [2]. Even more, geographical visualization data is not restricted to height-maps and terrain textures (see Figures 1 and 2). Now urban models “emerge” from the terrain and each day more demanding links to data bases increase the range of models and information that must be displayed: even physical interaction through haptics. So, visualization systems, in this and in many different areas, must be able to distinguish the characteristics of the data

set within the actual navigation range. Also they should identify the characteristics of the current task requested by the user. With this knowledge, applications should apply the most appropriated algorithms.

Surveys use different classification criteria [1, 3, 4, 5], for instance, some gather algorithms for non-polygonal models [1], while others consider simple primitives [3]. Although some technical papers are not general surveys, they provide a valuable compilation of previous work [6, 7, 8, 9].

Before entering the goal of the paper, applications will be summarized.

Robotics has studied the collision problem in detail [10]. Some problems can be pre-processed, so they are no real time systems, but in most cases the problem must be solved on-line. Physical Simulation covers many engineering analysis like automobile crash simulations [11] or mechanism analysis [12].

Collisions have been often considered by Virtual Reality applications. Some simulate physical environments so that operators can visualize, explore and interact with objects of the virtual environment [13, 14]. Simple VR-CAD applications can make measures and interference analysis among objects [15]. There are applications that deal with virtual prototypes to verify assemblies or carry out maintenance tasks [16]. Collision response is used to control force feedback devices, in order to provide realistic tactile sensations [17, 18, 19].

Animation often addresses collisions, as in cloth [20, 21] and herd-flock problems [22, 23]. Collisions also play a role for achieving realism in games [24].

Each one of the following sections will deal with one characteristic but before going on, one question should be considered: whether the geometric model used to describe the scenario is a characteristic of a given collision problem. In first place, one distinction must be proposed: which geometric model is the native “source” and which one has been derived from the native one as an algorithmic requirement to solve the collision problem. A derived model is more related to the algorithms, i.e. the ways to solve the problem, than to the characteristics that describe the problem. An analysis of

Analysis of Collision Detection in Dense Geometry Sets

Josune Hernantes¹, Diego Borro¹, Luis Matey¹, Alejandro García-Alonso²

¹CEIT and TECNUN (University of Navarra) ²EHU
Manuél de Lardizabal 15, 20018 San Sebastián, Spain

Email: {jhernantes,dborro,lmatey}@ceit.es alex.galonso@ehu.es

Abstract

This paper analyses the Collision Detection problem in Dense Geometry sets, such as those that are found in mechanical applications, especially in Virtual Reality applications for the evaluation of machine designs using large haptic systems. Studies have been carried out with VFM and a well-known public collision method: V-Collide. VFM collision method based on voxels is briefly described. VFM has a very good behavior in static environments with a high density of triangles and composed of millions of them. We apply this method for maintainability simulations in the aeronautics industry having high frame rate even when it is used for computing force feedback with haptic devices. However, we know every collision technique depends on the specific problem we want to solve to, so experiments (in this kind of environments) have been made in order to check the advantages and weaknesses compared to V-Collide.

1 Introduction

The Collision detection problem is found in many different fields from Robotics to Computer Graphics and Computational Geometry. So, a great number of problems, applications, techniques and solutions appear in the bibliography.

In this way, we can find different methods depending on the problem, kind of application or environment. One technique giving good results in a particular application can have a bad behavior in another environment. That is why it is difficult to use a general method and it is often necessary to find the best technique for a specific problem.

In this paper we summarize our collision detection method (VFM) based on voxels describing the characteristics and the algorithm steps. After that,

we analyze the results obtained with VFM and V-Collide, one of the most used collision methods.

Our method is oriented towards static environments with a huge number of polygons, such as the mechanical systems used in the aeronautic industry. These environments are not large scenes like buildings or factories. They are compact systems that pack millions of polygons within a volume of a few cubic meters, what we call dense geometry sets.

In this kind of environments, the user moves a virtual tool and the collisions are detected between the tool and the mechanical model in order to perform maintainability tasks or accessibility analysis. Interference information is required to compute the forces that a haptic device must provide to the user while the virtual maintenance tests are performed.

The application has the following requirements: fast computation and precision. Fast computation means that the ultimate goal should be to match the haptic control frequency (at least 1 Khz). Precision means using models composed of a high number of triangles: models provided by industry are made of a few million polygons. Four years ago the goal was to manage models of nearly 2M, now designers want to test even larger models (4M and up).

Polygonalizations are automatically created by CAD systems and very thin and long triangles often appear. These slender triangles have the advantage that reduce the global amount of polygons, and so memory consumption but, as it will be commented later on, slender triangles seem to be ill fitted for some algorithms.

We are interested in checking collision methods because it is worth knowing which one is better for a specific problem. In this way, it would be possible to choose the best technique for each application. We use V-Collide as a reference due to its easy integration with any application, its

3rd Workshop in Virtual Reality Interactions and Physical Simulation "VRIPHYS" (2006)
C. Mendoza, I. Navazo (Editors)

Stable Collision Response Method in Complex Interactions

J. Hernantes¹, I. Diaz¹, H. Alvarez² & D. Borro¹

¹CEIT and TECNUN (University of Navarra). Manuel Lardizabal 15, 20018 San Sebastian, Spain

²University of the Basque Country, Manuel Lardizabal 1, 20018 San Sebastian, Spain

Abstract

Haptic technology is quite recent and therefore in many cases it is difficult to simulate real contacts or interactions with a high sensation of realism. Collision response methods that calculate the force-feedback tend to cause haptic instabilities when the normal direction changes abruptly. In consequence, collision events in sharp corners are often difficult to render properly by means of haptic devices. This paper describes a collision response method that first distinguishes the nature of the corners -concave or convex- and then provides an adequate solution in each case to avoid haptic instabilities.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Information Interfaces and Presentation]: User Interfaces — Haptic I/O, I.3.4 [Computer Graphics]: Three-Dimensional Graphics and Realism — Virtual reality

1. Introduction

Humans are able to perceive the environment using all their senses. Usually sight is the predominant sense, although some of the other senses are also needed to perform most tasks. For instance, maintainability studies need accessibility tests to verify whether each part of the mock-up is accessible or not. Obviously, a visual test is not enough to detect possible inaccessible parts. Providing users with the natural ability to use all their senses in a simulation environment is an important goal in the Virtual Reality research area. Within this context, haptic devices are used to provide us with force feedback in domains where it is needed, enhancing considerably interactivity. The process of computing and generating forces in response to user interactions with virtual objects is known as haptic rendering [SBM*95]. Three main modules can be identified in a typical haptic rendering algorithm: collision detection, collision response and control modules. This paper focuses specifically on the second module. Previous works dealt with the voxel-based collision detection [BGAM04] and control algorithms [GAGB05].

2. Description of Collision Response Algorithm

When two objects collide, the group of collided triangles of the static object is defined as the collision area. The method proposed subdivides this area into smaller areas called contact zones in order to facilitate the computation of the colli-

sion final haptic force. Each contact zone is represented by its resultant penetration value and normal vector. The final haptic force is computed taking into account the information of each contact zone.

The subdivision of the collision area into different contact zones is achieved taking into account the proximity and distribution of the static colliding triangles in the 3D space. After that, the method subdivides each contact zone according to the angle between normal vectors of the triangles grouped in each zone. The normal of each contact zone is computed as the average normal vectors of its faces.

Next, we compute the exact penetration depth using a perpendicular contact plane for each contact zone and subset of the vertices which define the tool model and are contained inside a sphere with radius r .

The final step is to obtain a proper resultant normal vector and a penetration depth value based on the information of each contact zone. Two different strategies have been developed.

2.1. Method based on penetration weights

The idea is to compute the resultant normal direction in terms of a weighted average, where the weights depend on the computed penetration values (Figure 1a). This is a valid strategy for convex corners. However, applying it to concave corners, we observe that the normal vector which belongs to the second contact zone will have a higher influence in the

Virtual Reality (2006)
DOI 10.1007/s10055-006-0028-4

ORIGINAL ARTICLE

Iñaki Díaz · Josune Hernantes · Ignacio Mansa
Alberto Lozano · Diego Borro · Jorge Juan Gil
Emilio Sánchez

Influence of multisensory feedback on haptic accessibility tasks

Received: 23 December 2005 / Accepted: 3 April 2006
© Springer-Verlag London Limited 2006

Abstract Environments of a certain nature, such as those related to maintenance tasks can benefited from haptic stimuli by performing accessibility simulation in a realistic manner. Accessibility is defined as the physical feasibility of accessing an element of a 3D model avoiding undesirable collisions. This paper studies the benefits that multisensory systems can provide in performing this kind of tasks. The research is specially focused on the improvements provided by auditory feedback to the user's performance. We have carried out a user study where participants had to perform an accessibility task with the aid of different combinations of sensorial stimuli. A large haptic interface for aeronautic maintainability has been extended with real-time sound generation capabilities to study this issue. The results of these experiments show that auditory stimuli provide with useful cues to the users helping them to correct trajectories and hence improving their performance.

Keywords Haptics · Accessibility · Force feedback · Auditory · Virtual environments · Synchronization · Multisensory interaction · Multimodal

1 Introduction

Humans are able to perceive the environment using all their senses. Usually sight is the predominant sense, although some of the other senses are also needed to perform most tasks. Sometimes, it is necessary to

perceive the environment in more detail and all our senses are unconsciously used to obtain the information we need. For instance, maintainability studies need of accessibility testing to verify whether each part of the model is accessible or not. Obviously, a visual test is not enough to detect possible inaccessible parts. In addition to this, the worker has to explore and manipulate the model or different parts of it to complete the assessment, thus more than one sense is necessary.

Providing users with the natural ability to use all their senses in a simulation environment is an important goal in the Virtual Reality research area. Within this context, haptic devices are used to provide us with force feedback in domains where it is needed, such is the case of the accessibility studies. Using these devices, interactivity is enhanced. Furthermore, the simulation efficiency can be improved by making users aware of other physical characteristics of the objects, such as the weight or surface smoothness. However, haptic technology is quite recent and as a result the tactile realism obtained is not as good as it would be desirable. In addition, rigid bodies cannot be haptically simulated as rigid as they appear in reality due to mechanical and control constraints (Gil et al. 2004; Colgate and Schenkel 1994) and the sensation of real surface textures in virtual environments still remains very complex. In an attempt to improve the overall perception, we can help users by adding sound to haptic systems. In reality, if we hear a person hitting an object with a hammer, we can deduce how much strength was used without really having to experience the impact force. Thus, adding sound to haptic applications seems like a valid hypothesis in improving the haptic perception.

The research in this paper is intended to validate this hypothesis for a specific problem with haptic interaction: accessibility. This problem relates to the accuracy to guide the virtual haptic representation through a three-dimensional (3D) model. Users must avoid colliding with the virtual model. However, when the collision is unavoidable, users will receive adequate feedback (visual, auditory and haptic) to be able to correct the

I. Díaz · J. Hernantes · I. Mansa · A. Lozano · D. Borro
J. J. Gil · E. Sánchez
CEIT, Paseo Manuel Lardizábal, 15,
20018 San Sebastián, Spain

I. Díaz · J. Hernantes · I. Mansa · A. Lozano · D. Borro
J. J. Gil (✉) · E. Sánchez
TECNUN, University of Navarra,
Paseo Manuel Lardizábal, 13, 20018 San Sebastián, Spain
E-mail: jjgil@ceit.es
Tel.: +34-943-212800
Fax: +34-943-213076

STABLE HAPTIC RESPONSE FOR COMPLEX INTERACTIONS

J. Hernantes, D. Borro

CEIT and TECNUN (University of Navarra), Manuel Lardizabal 15, 20018 San Sebastian, Spain
jhernantes@ceit.es, dborro@ceit.es

A. García-Alonso

University of Basque Country, Manuel Lardizabal 1, 20018 San Sebastian, Spain
alex.galonso@ehu.es

Keywords: Haptic Rendering, Collision Response, Virtual Environment.

Abstract: Haptic technology is quite recent and therefore in many cases it is difficult to simulate real contacts or interactions with a high sensation of realism. Collision response methods that calculate the force-feedback tend to cause haptic instabilities when the normal direction changes abruptly. In consequence, collision or contact events are often difficult to render properly in sharp corners by means of haptic devices. This paper describes a collision response method which not only provides users with a stable force feedback, but also a comfortable and convincing haptic interaction. The experimental results show that this approach leads to a smoother force evolution which manages to avoid discontinuities and enhances the quality in the interaction with corners.

1 INTRODUCTION

Humans are able to perceive the environment using all their senses. Usually sight is the predominant sense, although some of the other senses are also needed to perform most tasks. Sometimes, it is necessary to perceive the environment in more detail and all our senses are unconsciously used to obtain the information we need. For instance, maintainability studies need accessibility tests to verify whether each part of the mock-up -static object- is accessible or not. Obviously, a visual test is not enough to detect possible inaccessible parts or manipulate different parts of a virtual model in order to complete an assessment task.

Providing users with the natural ability to use all their senses in a simulation environment is an important goal in the Virtual Reality research area. Within this context, haptic devices are used to provide us with force feedback in domains where it is needed, considerably enhancing interactivity.

Following with the example of virtual simulation of maintainability tasks, an operator moves a virtual tool or mobile object such as a screwdriver using the haptic device, and collides with the different parts that constitute an engine. The haptic forces restored in the collision event should make the operator feel

the virtual objects like real rigid objects, and prevent any interpenetration with the environment.

This paper focuses on the problems that virtual corners cause in haptic interactions, in which the force direction changes suddenly causing instabilities in the haptic system. The proposed algorithm manages a proper resultant penetration and normal direction of the collision. In addition to stability and time performance, we have paid particular attention to provide users with a comfortable algorithm to interact with.

Some experiments have been performed to analyze the quality of the proposed method using a haptic device called LHfAM (Savall et al., 2004), which only provides force feedback in three translational degrees of freedom. However, it can be used with any commercial haptic device. The results show that this algorithm avoids abrupt changes in the computed haptic force obtaining a more continuous force. As a result, haptic stability is improved in complex intersection of surfaces.

The article is organized as follows. Firstly, we present the state of the art of the collision response methods. Section 3 describes the specific problem involving the computation of force feedback. After that, the description of the proposed collision response algorithm is presented in Section 4. Section

HAVE 2008 – IEEE International Workshop on
Haptic Audio Visual Environments and their Applications
Ottawa – Canada, 18-19 October 2008

Evaluation of Sensory Substitution to Simplify the Mechanical Design of a Haptic Wrist

Javier Martín
Applied Mechanics Department, CEIT
San Sebastián, Spain
jmamezaga@ceit.es

Joan Savall, Iñaki Díaz, Josune Hernantes, Diego Borro
Tecnun (University of Navarra), and
Applied Mechanics Department, CEIT
San Sebastián, Spain
{jsavall, idiaz, jhernantes, dborro}@ceit.es

Abstract— The design of a new 3 degrees-of-freedom (DOF) haptic wrist for our 3-DOF force feedback haptic device is presenting a great design challenge. Demanding target specifications make it difficult to find a satisfactory mechanical design solution. This paper studies whether sensory substitution can lead to the possibility of simplifying the mechanics while preserving haptic performance. The proposed approach consists of substituting the haptic actuation of one of the 3 rotational DOF by specially designed visual or auditory cues. We have carried out a user study where participants are asked to perform an accessibility task with the aid of different sensorial stimuli. The results of these experiments show that the performance obtained by visual or auditory substitution can be satisfactory for accomplishing specific tasks, although it does not meet the performance obtained with a full haptic wrist.

Keywords- Haptic interfaces, sensory substitution, multisensory

I. INTRODUCTION

Providing users with the natural ability to use all their senses in a simulation environment is an important goal in the Virtual Reality research area. Within this context, haptic devices are used to provide us with force feedback in domains where it is needed. Over the past years haptic interfaces have been successfully integrated into a wide range of fields such as engineering [1] or surgery [2, 3]. These devices allow users to interact with a remote or virtual environment using the sense of touch.

LHIFAM (Large Haptic Interface for Aeronautics Maintainability) [4, 5] is a floor-grounded device developed at CEIT (Centro de Estudios e Investigaciones Técnicas de Guipúzcoa). It measures 6 DOF translation and orientation and provides 3 DOF force feedback, allowing the user to move within a human-size workspace. Currently it is being augmented to 6 active DOF in order to provide both force and torque feedback. However, required high output torques and wide rotational workspace make it difficult to find a satisfactory design solution [6, 7].

The research in this paper is intended to analyze whether sensory substitution for torque feedback can simplify the mechanics while preserving task performance. The idea is to substitute the torque feedback of one DOF by appropriate

visual or auditory stimuli and evaluate its performance. If successful, the proposed approach enables simpler kinematics, less friction, lightweight designs and lower costs.

A user study has been carried out to evaluate how far these sensorial modalities can be from an active haptic DOF to provide the users with valuable collision information. The virtual environment designed for the experiments simulates an accessibility task, where haptic feedback is essential for the operator to perform the task properly.

This paper is organized as follows: Section II summarizes previous work on sensory substitution in virtual haptic environments. Section III describes the sensory substitution strategy proposed for the haptic wrist, and Section IV presents the experiment carried out to support this research, along with yielded results. Finally, conclusions and future work are presented in Section V.

II. RELATED WORK

With the introduction of haptic interfaces in virtual reality systems, several studies have evaluated the benefits and backwards of multisensory integration. A good introduction to multisensory interaction is presented in [8].

In the sensory substitution field, [9] analyzed the effects of substituting direct haptic feedback with visual and auditory cues to provide surgeons with a representation of force magnitude in teleoperated suture-manipulation procedures. They showed that sensory substitution is capable of providing sufficient feedback for the user to control these robotically-applied forces.

In [10], the authors carried out a user-study to evaluate the influence of sensory augmentation and sensory substitution in telepresent tasks. Their results showed that the use of haptic feedback improved significantly operation effectiveness. In cases in which haptic feedback was not provided, the best results were achieved when auditory and visual force information was applied together.

In virtual assembly tasks, [11] investigated whether the substitution of force feedback with auditory cues improved manipulation performance and subjective perception of usability. They found that, depending on the specific goal

Haptic Wrists: An Alternative Design Strategy Based on User Perception

Javier Martín¹, Joan Savall², Iñaki Díaz², Josune Hernantes² & Diego Borro²

¹Centro de Estudios e Investigaciones Técnicas (CEIT), Department of Applied Mechanics, Paseo Manuel Lardizábal 15, 20018, San Sebastian, Spain

²Centro de Estudios e Investigaciones Técnicas (CEIT), Department of Applied Mechanics, and Tecnun (University of Navarra), Paseo Manuel Lardizábal 13, 20018, San Sebastian, Spain

Abstract

A new 3 degrees-of-freedom (DOF) torque feedback wrist is being developed to be added to an existing 3 DOF force feedback haptic device. It is difficult to find a satisfactory solution to the mechanical design problem, mainly because of the required large rotational workspace and severe weight constraints. This work proposes an alternative design strategy based on user perception which allows simplification of the mechanics. The proposed approach consists of substituting the last rotational DOF of the wrist with a pseudo-haptic DOF. Thanks to specially designed visuotactile cues, the pseudo-haptic DOF is integrated with the active DOF into the same device, being able to generate free motion and collision detection perception to the user. This approach provides for simpler kinematics, lightweight designs, lower inertias and less friction, which are key advantages for the inclusion of torque feedback into force feedback devices.

1 Introduction and motivation

Haptic interfaces are both i) input devices which capture the movements of a user and ii) feedback devices which allow the user to physically interact with a virtual or remote environment. Some haptic applications require a large-scale workspace. One example is the Large Haptic Interface for Aeronautics Maintainability (LHifAM) developed at CEIT [1, 2]. LHifAM measures 6 DOF translation and orientation and provides 3 DOF force feedback, allowing the user to interact with full-scale 3D virtual aircraft engines for maintainability analysis purposes.

Currently LHifAM is being augmented to 6 active DOF in order to provide both force and torque feedback. This demands a lightweight torque feedback wrist providing high output torques within a large rotational workspace. A state-of-the-art in mechanisms for haptic torque feedback was presented by the authors [3], concluding that none of the reviewed wrists suits our needs. An in-depth study of the challenges presented by torque feedback was also carried out by the authors [4] showing that, design-wise, there is no way to simultaneously guarantee a large rotational workspace with high quality performance. In order to avoid the severe mechanical limitations, this work moves the scope of the design solutions closer to the user's perception through pseudo-haptic and sensory substitution techniques. This allows simplification of the mechanics.

The paper is organized as follows: Section 2 summarizes previous work on pseudo-haptics and sensory substitution. The concept, implementation, applicability and limitations of the proposed approach are explained in Section 3. Finally, conclusions and future work are presented in Section 4.

2 Related work

The first choice for simplifying the 6 DOF mechanics is to simulate torque with the pure force mechanism itself, without adding any wrists. In that respect, Lécuyer et al. [5] propose an interaction paradigm called the "Virtual Haptic Sphere" which allows one to use the PHANToM™ alternately as a 3 DOF input/output device in translation and as a 2 DOF input/output device in rotation.

Another work [6] argues that even without true torque feedback exerted by the haptic device, the user can voluntarily supply the sense of torque, aided by visualization and previous experience with the task being performed. This is emphasized by [7] with further experiments displaying haptic information with both torque and visual feedback. His work poses the following question: "Can the

<p>Noname manuscript No. (will be inserted by the editor)</p>

Effective Haptic Rendering Method for Complex Interactions

Josune Hernantes · Iñaki Díaz · Diego Borro · Jorge Juan Gil

Received: date / Accepted: date

Abstract The real-time computation of the forces and torques applied to users as a result of their actions in a Virtual Reality application is a complicated and essential task in order for haptic interfaces to be effective and applicable in many fields. This paper presents a haptic rendering method that allows for realistic and stable interaction with virtual environments. The method ensures an improved haptic response by seeking a compromise between accuracy and computational cost. This research has been especially focused on complex virtual contacts, such as multiple simultaneous collisions or geometrical discontinuities, in order to develop an effective method. The performance of the proposed algorithms has been validated with two different virtual scenarios.

Keywords Haptic rendering · Collision detection · Virtual Reality

1 Introduction

The development of haptic technology is allowing the introduction of Virtual Reality systems as teaching and

Josune Hernantes
CEIT and TECNUN (University of Navarra)
Paseo Manuel Lardizábal 15, E-20018 San Sebastián, Spain
Tel.: +34-943212800
Fax: +34-943213076
E-mail: jhernantes@ceit.es

Iñaki Díaz
E-mail: idiaz@ceit.es

Diego Borro
E-mail: dborro@ceit.es

Jorge Juan Gil
E-mail: jjgil@ceit.es

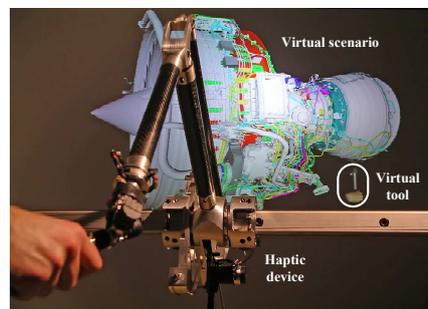


Fig. 1 Haptic interaction with a virtual environment

working tools into many fields such as engineering [1, 2] or surgery [3, 4].

Haptic devices allow users to interact with a certain environment, either remote or virtual, through the sense of touch, considerably enhancing interactivity. A haptic device is a mechanism that allows users to control the movements of a virtual tool or a real robot and receive tactile and kinesthetic information from the working environment (Fig. 1).

The usability of these systems is conditioned by the quality of the haptic feedback applied to the user. Technologically, the computation of appropriate and realistic haptic stimuli continues to be a complicated issue. The human sensory-motor system demands a fast update rate (at least 1 kHz) for the haptic stimuli applied to the user in order to avoid instabilities in the system and to present rigid objects with reasonable stiffness [5]. However, this update rate is often difficult to reach by haptic rendering methods, especially when working in complex environments. One possible solution is to

ANEXO C

REFERENCIAS

- Adachi, Y., Kumano, T., and Ogino, K., "Intermediate Representation for Stiff Virtual Objects," *Proceedings of the IEEE Virtual Reality Annual International Symposium*, pp. 203-210, 1995.
- Amundarain, A., Aguinaga, I., Borro, D., García-Alonso, A., and Matey, L., "Realidad virtual para el mantenimiento de motores de avión," *Proceedings of the CEIG (Congreso Español de Informática Gráfica)*, pp. 233-246. La Coruña, Spain, 2003.
- Angerilli, M., Frisoli, A., Salsedo, F., Marcheschi, S., and Bergamasco, M., "Haptic Simulation of an Automotive Manual Gearshift," *Proceedings of the International Workshop on Robot and Human Interactive Communication*, 2001.
- Avanzini, F. and Crosato, P., "Haptic-auditory rendering and perception of contact stiffness," *Haptic and Audio Interaction Design, LNCS*, vol. 4129, pp. 24-35, 2006.
- Ayala, D., Brunet, P., Juan, R., and Navazo, I., "Object representation by means of nonminimal division quadtrees and octrees," *ACM Transactions on Graphics*, vol. 4, pp. 41-59, 1985.
- Bach-y-Rita, P., Webster, J. G., Tompkins, W. J., and Crabb, T., "Sensory Substitution for Space Gloves and for Space Robots," *Proceedings of the Workshop on Space Telerobotics*, pp. 51-57. Pasadena, California, 1987.
- Basdogan, C., De, S., Kim, J., Muniyandi, M., Kim, H., and Srinivasan, M. A., "Haptics in Minimally Invasive Surgical Simulation and Training," *IEEE Computer Graphics and Applications*, vol. 24, pp. 56-64, 2004.
- Basdogan, C., Ho, C.-H., Srinivasan, M. A., Small, S., and Dawson, S., "Force Interactions in Laparoscopic Simulations: Haptic Rendering of Soft

- Tissues," *Proceedings of the Medicine Meets Virtual Reality VI Conference*, pp. 385-391. San Diego, CA, 1998.
- Basdogan, C. and Srinivasan, M. A., "Haptic Rendering in Virtual Environments," in *Handbook of Virtual Environments: Design, Implementation, and Applications (Human Factors and Ergonomics)*, K. M. Stanney, Ed. London: Lawrence Earlbaum Inc., 2002, pp. 117-134.
- Borro, D., "Colisiones en Estudios de Mantenibilidad con Restitución de Esfuerzos sobre Maquetas Digitales Masivas y Compactas," in *Department of Applied Mechanics*. San Sebastián: University of Navarre, 2003.
- Borro, D., García-Alonso, A., and Matey, L., "Approximation of Optimal Voxel Size for Collision Detection in Maintainability Simulations within Massive Virtual Environments," *Computer Graphics Forum*, 2004a.
- Borro, D., Hernantes, J., García-Alonso, A., and Matey, L., "Collision problem: Characteristics for a Taxonomy," *Proceedings of the 9th International Conference Information Visualisation (IV'05)*, pp. 410-415. London, England, 2005.
- Borro, D., Savall, J., Amundarain, A., Gil, J. J., García-Alonso, A., and Matey, L., "A Large Haptic Device for Aircraft Engines Maintainability," *IEEE Computer Graphics and Applications*, vol. November, 2004b.
- Buckley, C. E. and Leifer, L. J., "A proximity metric for continuum path planning," *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pp. 1096-1102, 1985.
- Burdea, G., "Haptic Interfaces for Virtual Reality," *Proceedings of the International Workshop on Virtual Reality and Prototyping*, pp. 87-96. Laval, France, 1999.
- Cameron, S. A., "Enhancing GJK: Computing Minimum and Penetration Distances between Convex Polyhedra," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1997.
- Cameron, S. A. and Culley, R. K., "Determining the Minimum Translation Distance between Two Convex Polyhedra," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 591-596. San Francisco, USA, 1986.
- Cohen, A. and Chen, E., "Six Degree-of-Freedom Haptic System as a Desktop Virtual Prototyping Interface," *Proceedings of the Winter Annual Meeting, Dynamics Systems and Control*, pp. 401-402. Nashville, Tennessee, 1999.
- Cohen, J. D., Lin, M. C., Manocha, D., and Ponamgi, M. K., "I-Collide: An Interactive and Exact Collision Detection System for Large-Scale Environments," *Proceedings of the ACM Interactive 3D Graphics Conference*, pp. 189-196, 1995.

- Colgate, J. and Brown, J., "Factors affecting the z-width of a haptic display," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3205–3210. San Diego, California, USA, 1994.
- Colgate, J., Stanley, M., and Brown, J., "Issues in the haptic display of tool use," *IEEE Intelligent Robot and Systems*, vol. 3, pp. 140-145, 1995.
- Constantinescu, D., Salcudean, S. E., and Croft, E. A., "Haptic rendering of rigid contacts using impulsive and penalty forces," *IEEE Transactions on Robotics and Automation*, vol. 21, pp. 309-323, 2005.
- Chang, B. and Colgate, J. E., "Real-time impulse-based simulation of rigid body systems for haptic display," *Proceedings of the ASME International Mechanical Engineering Congress and Exhibition*, pp. 1-8, 1997.
- Chen, E., "Six Degree-of-Freedom Haptic System For Desktop Virtual Prototyping Applications," *Proceedings of the International Workshop on Virtual Reality and Prototyping*, pp. 97-106. Laval, France, 1999.
- Choi, Y.-K., Li, X., Rong, F., Wang, W., and Cameron, S., "Computing the Minimum Directional Distance between Two Convex Polyhedra," University of Hong Kong TR-2006-1, 2006.
- DiFilippo, D. and Pai, D. K., "The AHI: An Audio and Haptic Interface for Contact Interactions," *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, pp. 149-158. San Diego, California, USA, 2000.
- DiFranco, D. E., Beauregard, G. L., and Srinivasan, M. A., "The effect of auditory cues on the haptic perception of stiffness in virtual environments," *Proceedings of the 6th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 17-22, 1997.
- Dobkin, D., Hershberger, J., Kirkpatrick, D., and Suri, S., "Computing the Intersection-Depth of Polyhedra," *Algorithmica*, vol. 9, pp. 518-533, 1993.
- Edwards, W., Barfield, W., and Nussbaum, A., "The Use of Force Feedback and Auditory Cues for Performance of an Assembly Task in an Immersive Virtual Environment," *Virtual Reality*, vol. 7, pp. 112-119, 2004.
- Foley, J. D., Dam, A. V., Feiner, S. K., and Hughes, J. F., *Computer Graphics: Principles and Practice*, Second ed: Addison-Wesley, 1990.
- García-Alonso, A., Serrano, N., and Flaquer, J., "Solving the Collision Detection Problem," *IEEE Computer Graphics and Applications*, vol. 14, pp. 36-43, 1994.
- Gil, J. J., "Control de Dispositivos Físicos de Gran Espacio de Trabajo para la Interacción Táctil con Entornos Virtuales," in *Applied Mechanics*. San Sebastian, Spain: Tecnun, University of Navarre, 2003, pp. 195.

- Gil, J. J., Sánchez, E., Hulin, T., Preusche, C., and Hirzinger, G., "Stability boundary for haptic rendering: Influence of damping and delay," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 124-129. Roma, Italy, 2007.
- Gilbert, E. G., Johnson, D. W., and Keerthi, S. S., "A Fast Procedure for Computing the Distance between Complex Objects in Three Dimensional Space," *IEEE Journal of Robotics and Automation*, vol. 4, pp. 193-203, 1988.
- Gillespie, R. B. and Cutkosky, M. R., "Stable user-specific haptic rendering of the virtual wall," *Proceedings of the ASME International Mechanical Engineering Congress and Exposition*, pp. 397-406. Atlanta, GA, USA, 1996.
- Gottschalk, S., Lin, M. C., and Manocha, D., "OBBTree: A Hierarchical Structure for Rapid Interference Detection," *Proceedings of the ACM SIGGRAPH - Computer Graphics*, pp. 171-180, 1996.
- Gouraud, H., "Continuous shading of curved surfaces," *IEEE Transactions on Computers*, vol. 6, pp. 623-629, 1971.
- Grand Le, S., "Broad-Phase Collision Detection with CUDA," in *GPU Gems 3*, H. Nguyen, Ed.: Addison-Wesley, 2007, pp. 697-721.
- Gregory, A., Lin, M. C., Gottschalk, S., and Taylor, R., "Fast and accurate collision detection for haptic interaction using a three degree-of-freedom force-feedback device," *Computational Geometry - Theory & Applications*, pp. 69-89, 2000a.
- Gregory, A., Lin, M. C., Gottschalk, S., and Taylor, R., "H-Collide_ A Framework for Fast and Accurate Collision Detection for Haptic Interaction," *Proceedings of the IEEE Virtual Reality Conference*, 1999.
- Gregory, A., Mascarenhas, A., Ehmann, S., Lin, M., and Manocha., D., "Six Degree-of-Freedom Haptic Display of Polygonal Models," *Proceedings of the Conference on Visualization*, pp. 139-146, 2000b.
- Guest, S., Catmur, C., Lloyd, D., and Spence, C., "Audiotactile interactions in roughness perception," *Experimental Brain Research*, vol. 146, pp. 161-171, 2002.
- Guest, S. and Spence, C., "What Role does Multisensory Integration Play in the Visuotactile Perception of Texture?" *International Journal of Psychophysiology*, vol. 50, pp. 63-80, 2003.
- Harada, T., "Real-Time Rigid Body Simulation on GPUs," in *GPU Gems 3*, H. Nguyen, Ed.: Addison-Wesley, 2007, pp. 611-632.

- Hasegawa, S. and Sato, M., "Real-time Rigid Body Simulation for Haptic Interactions Based on Contact Volume of Polygonal Objects," *Computer Graphics Forum*, vol. 23, pp. 529-538, 2004.
- Hayward, V., "Fast Collision Detection Scheme by Recursive Decomposition of a Manipulator Workspace," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1044-1049. San Francisco, CA, 1986.
- Held, M., Klosowski, J. T., and Mitchell, J. S. B., "Evaluation of Collision Detection Methods for Virtual Reality Fly-Throughs," *Proceedings of the Conference on Computer Geometry*, pp. 205-210. Québec City, Québec, Canada, 1995.
- Hippmann, G., "An Algorithm for Compliant Contact Between Complexly Shaped Surfaces in Multibody Dynamics," *Proceedings of the International Conference on Advanced in Computational Multibody Dynamics*. Lisbon, Portugal, 2003.
- Ho, C.-H., Basdogan, C., and Srinivasan, M. A., "Ray-based haptic rendering: force and torque interactions between a line probe and 3D objects in virtual environments," vol. 19, pp. 668-683, 2000.
- Hubbard, P. M., "Approximating Polyhedra with Spheres for Time-Critical Collision Detection," *ACM Transactions on Graphics*, vol. 15, pp. 179-210, 1996.
- Hudson, T. C., Lin, M. C., Cohen, J., Gottschalk, S., and Manocha, D., "V-Collide: Accelerated Collision Detection for VRML," *Proceedings of the VRML (Virtual Reality Modeling Language)*, pp. 119-125, 1997.
- Jain, A. K., Murty, M. N., and Flynn, P. J., "Data clustering: A review," *ACM Computing Surveys*, vol. 31, pp. 264-323, 1999.
- Jiménez, J. J., Segura, R. J., Feito, F. R., and Ogáyar, C. J., "Particle Oriented Collision Detection using Simplicial Coverings and Tetra-Trees," *Computer Graphics Forum*, vol. 25, pp. 53-68, 2006.
- Jiménez, P., Thomas, F., and Torras, C., "3D collision detection: a survey," *Computers & Graphics*, vol. 2, pp. 269-285, 2001.
- Johnson, D. and Cohen, E., "Spatialized normal cone hierarchies," *Proceedings of the ACM Symposium on Interactive 3D Graphics*, pp. 129-134, 2001.
- Johnson, D. E. and Willemsen, P., "Six degree of freedom haptic rendering of complex polygonal models," *Proceedings of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS'03)*, pp. 229-235, 2003.

- Kim, L., Kyrikou, A., Sukhatme, G. S., and Desbrun, M., "An Implicit-Based Haptic Rendering Technique," *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*. EPFL, Lausanne, Switzerland, 2002a.
- Kim, Y. J., Lin, M. C., and Manocha, D., "DEEP: Dual-space Expansion for Estimating Penetration depth between convex polytopes," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. Washington D.C., USA, 2002b.
- Kim, Y. J., Otaduy, M. A., Lin, M. C., and Manocha, D., "Fast Penetration Depth Computation for Physically-based Animation," *Proceedings of the ACM Symposium on Computer Animation*, 2002c.
- Kim, Y. J., Otaduy, M. A., Lin, M. C., and Manocha, D., "Six-Degree-of-Freedom Haptic Rendering Using Incremental and Localized Computations," *Presence - Teleoperators and Virtual Environments*, vol. Vol 12, pp. 277-295, 2003.
- Kipfer, P., "LCP Algorithms for Collision Detection Using CUDA," in *GPU Gems 3*, H. Nguyen, Ed.: Addison-Wesley, 2007, pp. 723-740.
- Kitagawa, M., Dokko, D., Okamura, A. M., and Yuh, D. D., "Effect of Sensory Substitution on Suture-Manipulation Forces for Robotic Surgical Systems," *Journal of Thoracic and Cardiovascular Surgery*, vol. 129, pp. 151-158, 2005.
- Lawn, M., Takeda, T., and Yamada, T., "Development of a 5 Axis Upper Limb Force Display Operating in a VR Environment and Training," *Proceedings of the ICAT'97*, pp. 99-105. Tokyo, Japon, 1997.
- Laycock, S. D. and Day, A. M., "A survey of Haptic Rendering Techniques," *Computer Graphics Forum*, vol. 26, pp. 50-65, 2007.
- Lécuyer, A., Burkhardt, J. M., Coquillart, S., and Coiffet, P., ""Boundary of Illusion": an Experiment of Sensory Integration with a Pseudo-Haptic System," *Proceedings of the International Conference on Virtual Reality*, pp. 115-122. Yokohama, Japan, 2001.
- Lécuyer, A., Coquillart, S., and Coiffet, P., "Simulating Haptic Information with Haptic Illusions in Virtual Environments," *Proceedings of the NATO RTA/Human Factors & Medicine Panel Workshop*. The Hague, The Netherlands, 2000a.
- Lécuyer, A., Coquillart, S., Kheddar, A., Richard, P., and Coiffet, P., "Pseudo-Haptic Feedback: Can Isometric Input Devices Simulate Force Feedback?" *Proceedings of the International Conference on Virtual Reality*, pp. 83-90, 2000b.
- Lecuyer, A., Mégard, C., Burkhardt, J. M., Lim, T., Coquillart, S., Coiffet, P., and Graux, L., "The effect of haptic, visual and auditory feedback on an insertion task on a 2-screen workbench," *Proceedings of the Immersive Projection Technology Symposium (IPT)*, 2002.

- Lenay, C., Gapenne, O., Hanne-ton, S., Marque, C., and Genouëlle, C., "Sensory substitution, limits and perspectives," in *Touch for Knowing*: John Benjamins Publishers, Amsterdam, 2003, pp. 275-292.
- Lin, M. C., "Efficient Collision Detection for Animation and Robotics," in *Department of Electrical Engineering and Computer Science*. Berkeley: University of California, 1993.
- Lin, M. C. and Canny, J. F., "A Fast Algorithm for Incremental Distance Calculation," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. Sacramento, California, 1991.
- Lin, M. C. and Gottschalk, S., "Collision Detection between geometric models: a survey," *Proceedings of the IMA Conference on Mathematics of Surfaces VIII*, pp. 37-56. Birmingham, UK, 1998.
- Lin, M. C. and Manocha, D., "Collision and Proximity Queries," in *Handbook of Discrete and Computational Geometry*, J. E. Goodman and J. O'Rourke, Eds., Second Edition ed: CRC Press, 2004.
- Lin, M. C. and Manocha, D., "Interference detection between curved objects for computer animation," *Models and Techniques in Computer Animation*, pp. 43-57, 1993.
- Lin, M. C., Manocha, D., Cohen, J., and Gottschalk, S., "Collision Detection: Algorithms and Applications," *Proceedings of the Algorithms for Robotics Motion and Manipulation*, pp. 129-142, 1996.
- Mansa, I., Amundarain, A., Hernantes, J., García-Alonso, A., and Borro, D., "Occlusion Culling for Dense Geometric Scenarios," *Proceedings of the Laval Virtual 2006 (VRIC 2006)*. Laval, France, 2006.
- Mark, W. R., Randolph, S. C., Finch, M., Verth, J. M. V., and II, R. M. T., "Adding Force Feedback to Graphics Systems: Issues and Solutions," *Proceedings of the ACM SIGGRAPH - Computer Graphics*, pp. 447-452. New Orleans, USA, 1996.
- Martin, J. and Savall, J., "Mechanisms for Haptic Torque Feedback," *Proceedings of the WorldHaptics'05*, pp. 611-614. Pisa, Italy, 2005.
- Massie, T. H. and Salisbury, J. K., "The PHANTOM Haptic Interface: A Device for Probing Virtual Objects," *Proceedings of the Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. Chicago, IL, 1994.
- Massimino, M. J., "Improved Force Perception Through Sensory Substitution," *Control Engineering Practice*, vol. 3, pp. 215-222, 1995.

- Mc Gee, M. R., Gray, P. D., and Brewster, S. A., "The Effective Combination of Haptic and Auditory Textural Information," *Proceedings of the First Workshop on Haptic Human-Computer Interaction*, pp. 118-126, 2000.
- McGee, M. R., Gray, P. D., and Brewster, S. A., "Feeling rough: Multimodal perception of virtual roughness," *Proceedings of the Eurohaptics*, pp. 29-33. Birmingham, UK, 2001.
- McNeely, W. A., Puterbaugh, K. D., and Troy, J. J., "Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling," *Proceedings of the ACM SIGGRAPH - Computer Graphics*, pp. 401-408. Los Angeles, California, USA, 1999.
- McNeely, W. A., Puterbaugh, K. D., and Troy, J. J., "Voxel-Based 6-DOF Haptic Rendering Improvements," *Haptics-e*, vol. 3, 2006.
- Mirtich, B., "V-Clip: Fast and Robust Polyhedral Collision Detection," *ACM Transactions on Graphics*, vol. 17, pp. 177-208, 1998.
- Mirtich, B. and Canny, J., "Impulse-based dynamic simulation," *Proceedings of the Workshop on Algorithmic Foundations of Robotics*, pp. 407-418, 1994.
- Möller, T., "A Fast Triangle-Triangle Intersection Test," *Journal of Graphics Tools (JGT)*, vol. 2, pp. 25-30, 1997.
- Möller, T. and Trumbore, B., "Fast, minimum storage ray-triangle intersection," *Journal of Graphics Tools (JGT)*, vol. 2, pp. 21-28, 1997.
- Moore, M. and Wilhelms, J., "Collision Detection and Response for Computer Animation," *Proceedings of the ACM SIGGRAPH - Computer Graphics*, pp. 289-298. Atlanta, USA, 1988.
- Morgenbesser, H. B. and Srinivasan, M. A., "Force Shading for Haptic Shape Perception," *Proceedings of the The International Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, pp. 407-412, 1996.
- Naylor, B., Amanatides, J., and Thibault, W., "Merging BSP Trees Yields Polyhedral Set Operations," *Proceedings of the ACM SIGGRAPH - Computer Graphics*, pp. 115-124. Dallas, USA, 1990.
- Nelson, D., Johnson, C., and Cohen, E., "Haptic rendering of surface-to-surface sculpted model interaction," *Proceedings of the 8th Symposium of Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 1999.
- Niemeyer, G. and Probal, M., "Dynamic Proxies and Haptic Constraints," 2004.
- O'Sullivan, C., Dingliana, J., Ganovelli, F., and Bradshaw, G., "Collision Handling for Virtual Environments," *Proceedings of the Eurographics Tutorials*. Manchester, UK, 2001.

- Ortega, M., Redon, S., and Coquillart, S., "A Six Degree-of-Freedom God-Object Method for Haptic Display of Rigid Bodies," *Proceedings of the IEEE International Conference on Virtual Reality*, pp. 191-198. Alexandria, Virginia, USA, 2006.
- Otaduy, M. A., "6-DOF Haptic Rendering Using Contact Levels of Detail and Haptic Textures," in *Computer Science*. Chapel Hill: University of North Carolina, 2004, pp. 197.
- Otaduy, M. A. and Lin, M. C., "Sensation Preserving Simplification for Haptic Rendering," *Proceedings of the SIGGRAPH*, pp. 543-553. San Diego, CA, 2003.
- Otaduy, M. A. and Lin, M. C., "Stable and Responsive Six-Degree-of-Freedom Haptic Manipulation Using Implicit Integration," *Proceedings of the World Haptics Conference*, pp. 247-256. Pisa, Italy, 2005.
- Owens, J. D., Luebke, D., Govindaraju, N. K., Harris, M., Krüger, J., Lefohn, A. E., and Purcell, T. J., "A Survey of General-Purpose Computation on Graphics Hardware," *Proceedings of the Eurographics*, pp. 21-51, 2005.
- Pai, D. K., "Multisensory interaction: Real and Virtual," in *Robotics Research: The Eleventh International Symposium*, vol. 15, P. Dario and R. Chatila, Eds.: Springer-Verlag GmbH, 2005, pp. 489-498.
- Patoglu, V. and Gillespie, R., "Haptic rendering of parametric surfaces using a feedback stabilized extremal distance tracking algorithm," *Proceedings of the 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 391-399. Chicago, 2004.
- Petzold, B., Zaeh, M. F., Faerber, B., Deml, B., Egermeier, H., Schilp, J., and Clarke, S., "A study on visual, auditory, and haptic feedback for assembly tasks," *Presence: Teleoperators and Virtual Environments*, vol. 13, pp. 16-21, 2004.
- Phong, B., "Illumination for computer generated pictures," *Communications of the ACM*, vol. 6, pp. 311-317, 1975.
- Ponamgi, M. K., Manocha, D., and Lin, M. C., "Incremental algorithms for collision detection between solid models," *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, pp. 51-64, 1997.
- Redon, S. and Lin, M. C., "A Fast Method for Local Penetration Depth Computation," *Journal of Graphics Tools (JGT)*, vol. 11, pp. 37-50, 2005.
- Renz, M., Preusche, C., Pötke, M., Kriegel, H.-P., and Hirzinger, G., "Stable Haptic Interaction with Virtual Environments Using an Adapted Voxmap-PointShell Algorithm," *Proceedings of the Eurohaptics*. Birmingham, UK, 2001.

- Ritter, J., "An Efficient Bounding Sphere," in *In Graphics Gems I*, A. S. Glassner, Ed.: Academic Press, Inc, 1990, pp. 301-303.
- Ruspini, D. C. and Khatib, O., "Haptic Display for Human Interaction with Virtual Dynamic Environments," *Robotic Systems*, vol. 18, pp. 769-783, 2001.
- Ruspini, D. C., Kolarov, K., and Khatib, O., "The Haptic Display of Complex Graphical Environments," *Proceedings of the SIGGRAPH 97*, 1997.
- Salcudean, S. E. and Vlaar, T., "On the emulation of stiff walls and static friction with a magnetically levitated input/output device," *ASME Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 127-132, 1994.
- Salisbury, J. K., Brock, D. L., Massie, T. H., Swarup, N., and Zilles, C., "Haptic Rendering: Programming Touch Interaction with Virtual Objects," *Proceedings of the 1995 Symposium on Interactive 3D Graphics*, pp. 132-130. Monterey, CA, USA, 1995a.
- Salisbury, J. K., Conti, F., and Barbagli, F., "Haptic Rendering: Introductory Concepts," *IEEE Computer Graphics and Applications*, vol. 24, pp. 24-32, 2004.
- Salisbury, K., Brock, D., Massie, T., Swarup, N., and Zilles, C., "Haptic Rendering: Programming Touch Interaction with Virtual Objects," *Proceedings of the ACM Symposium on Interactive 3D Graphics*, pp. 123-130. Monterey, California, USA, 1995b.
- Savall, J., Borro, D., Gil, J. J., and Matey, L., "Description of a Haptic System for Virtual Maintainability in Aeronautics," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2887-2892. EPFL, Lausanne, Switzerland, 2002.
- Savall, J., Martín, J., and Avello, A., "High Performance Linear Cable Transmission," *Journal of Mechanical Design*, vol. 130, 2008.
- Shimoga, K., "Finger Force and Touch Feedback Issues in Dexterous Telemanipulation," *Proceedings of the NASA-CIRSE International Conference on Intelligent Robotic Systems for Space Exploration*, NASA. Greenbelt, MD, USA, 1992.
- Siira, J. and, D. K. P., "Haptic Textures-A Stochastic Approach," *Proceedings of the IEEE International Conference on Robotics and Automation*. Minneapolis MN, 1996.
- Srinivasan, M. A. and Basdogan, C., "Haptics in Virtual Environments: Taxonomy, Research Status, and Challenges," *Computers & Graphics*, vol. 21, pp. 393-404, 1997.
- Srinivasan, M. A., G.L. B., and Brock, D. L., "The Impact of Visual Information on the Haptic Perception of Stiffness in Virtual Environments," *Proceedings of*

- the ASME Dynamic Systems and Control Division*, pp. 555-559. Atlanta, USA, 1996.
- Tavakoli, M., Patel, R., and Moallem, M., "Haptic feedback and sensory substitution during telemanipulated suturing," *Proceedings of the World Haptics*, pp. 543 – 544, 2005.
- Thompson II, T. V. and Cohen, E., "Direct Haptic Rendering of Complex Trimmed NURBS Models," *Proceedings of the Symposium on Haptic Interfaces ASME International ME'99 Congress and Exposition*. Nashville, Tennessee, 1999.
- Thompson II, T. V., Nelson, D. D., Cohen, E., and Hollerbach, J. M., "Maneuverable NURBS Models Within A Haptic Virtual Environment," *Proceedings of*, pp. 37-44. Dallas, USA, 1997.
- Turnbull, C. and Cameron, S., "Computing distances between NURBS-defined convex objects," *Proceedings of the IEEE Transaction on Robotics and Automation*, pp. 3685-3690, 1998.
- Van Den Bergen, G., "Collision Detection in Interactive 3D Computer Animation," in *Departamento de Matemáticas y Ciencias de la Computación*. Eindhoven, Germany: Universidad de Tecnología de Eindhoven, 1999a.
- Van Den Bergen, G., "Efficient Collision Detection of Complex Deformable Models using AABB Trees," *Journal of Graphics Tools (JGT)*, vol. 4, pp. 1-13, 1997.
- Van Den Bergen, G., "A Fast and Robust GJK Implementation for Collision Detection of Convex Objects," *Journal of Graphics Tools (JGT)*, vol. 4, pp. 7-25, 1999b.
- Van Den Bergen, G., "Proximity Queries and Penetration Depth Computation on 3d Game Objects," *Proceedings of the Game Developers Conference*, 2001.
- van den Doel, K. and Pai, D. K., "The Sounds of Physical Shapes," *Presence: Teleoperators and Virtual Environments*, vol. 7, pp. 382-395, 1998.
- Vanecek, G. J., "Back-face Culling applied to Collision Detection of Polyhedra," *Journal of Visualization and Computer Animation*, vol. 5, pp. 55-63, 1994.
- Volino, P. and Magnenat-Thalmann, N., "Interactive Cloth Simulation: Problems and Solutions," *Proceedings of the JWS*, 1997.
- Walker, S. P. and Salisbury, J. K., "Large Haptic Topographic Maps: MarsView and the Proxy Graph Algorithm," *Proceedings of the Siggraph*, pp. 83-92. San Diego, 2003.

- Wan, M. and McNeely, W. A., "Quasi-Static Approximation for 6 Degrees-of-Freedom Haptic Rendering," *Proceedings of the IEEE Visualization Conference (VIS'03)*. Seattle, Washington, USA, 2003.
- Zachmann, G., "Real-time and Exact Collision Detection for Interactive Virtual Prototyping," *Proceedings of the ASME Design Engineering Technical Conferences*. Sacramento, California, USA, 1997.
- Zhang, L., Kim, Y. J., Varadhan, G., and Manocha, D., "Generalized penetration depth computation," *Proceedings of the ACM Symposium on Solid and Physical modeling*, pp. 173-184. New York, USA, 2006.
- Zilles, C. B. and Salisbury, J. K., "A Constraint-based God-object Method For Haptic Display," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Human Robot Interaction, and Cooperative Robots*, pp. 146-151, 1995.