

Deformación de modelos elastoplásticos mediante interacción háptica



Francisco Joaquín Rodríguez Prados

Departamento de Lenguajes y Sistemas Informáticos

Universidad de Granada

Memoria para la obtención del grado de

Doctor en Informática

Marzo 2013

Editor: Editorial de la Universidad de Granada
Autor: Francisco Joaquín Rodríguez Prados
D.L.: GR 2324-2014
ISBN: 978-84-9083-371-1

A mis padres, que han hecho esto posible,
y a mi hermana, por estar a mi lado cuando más lo necesitaba.

Agradecimientos

Ha pasado algún tiempo desde que Alejandro León, recién llegado yo de mi primera estancia en Alemania en marzo de 2006, me sugirió empezar la carrera de la tesis doctoral. No pude hacer otra cosa que seguir su consejo y ejemplo, pues ya desde mi primer año en el Colegio Mayor Isabel la Católica (del cual fue un ejemplar subdirector) me supo inculcar el entusiasmo por la investigación. Le agradezco no solo que plantara la semilla, sino también que mantuviera vivo el espíritu durante todo el proceso.

A Juan Carlos Torres, mi otro director de tesis, jefe y maestro, le debo el apoyo en momentos cruciales de este trabajo, en los que estuve casi por abandonar. A veces firme y otras veces comprensivo, ha sabido guiarme sin dudar en todos los momentos: en los más activos y en los más dispersos.

A Victoria Luzón nunca podré agradecerle el ánimo que me prestó desde que nos conocimos. Siempre tuvo una buena palabra cuando parecía que nunca se iba a acabar, siempre tuvo un buen consejo cuando no sabía dónde agarrarme.

A Aurora Valenzuela, por animarme siempre a seguir adelante y terminar, y por enseñarme a ser no sólo un mejor investigador sino también una mejor persona.

Para Paco Soler sólo tengo buenas palabras, por ser siempre un buen compañero de laboratorio y mejor amigo.

A Juan Carlos Criado por iluminarme en el camino hacia el jacobiano con su mente preclara.

A Luis Otero por su profunda y útil revisión de estilo.

Y finalmente, no puedo terminar sin agradecer de corazón a todos mis amigos que me han apoyado cuando parecía que esta senda nunca se acabaría, y sin los cuales habría sido todo muchísimo más desolador.

Abstract

Solid deformation has always been a field of great interest among the Computer Graphics community. Many works have aimed to create a natural, reliable interaction with elastic or elastoplastic solids. However, many are also the problems that appear when playing in this scenario. When designing a simulation realism is always confronted with performance, and there are indeed many aspects of solid deformation that require intensive use of the computation resources. It is therefore necessary to find a compromise between these two desired properties.

This work will target at creating a new simulation environment for deformable materials maximizing realism while maintaining interactive response times. Additionally a tactile interaction method will be proposed. Whereas other works have focused on creating a fast response using the intuitive behaviour of clay, continuum mechanics will be used here to inspire and support the force rendering algorithm and the visualization of the modeled deformation.

In this document the motivation, formalization, design process and evaluation of the proposed system will be detailed. This pages will hence explain how following the formal behaviour of an elastoplastic solid a method offering a natural, efficient, and plausible interaction with deformable materials has been built.

Índice general

Índice general	vii
Prólogo	xiii
1 Introducción	1
1.1 Motivación	2
1.2 Contexto del trabajo	2
1.2.1 Representaciones	6
1.2.1.1 Modelos bidimensionales	6
1.2.1.2 Modelos tridimensionales	8
1.2.1.3 Análisis comparativo	10
1.2.2 Modelos volumétricos discretos	12
1.2.2.1 Organización de las muestras	13
1.2.2.2 Vecindades y distancias	16
1.2.2.3 Continuidad del modelo	17
1.2.2.4 Extracción de modelos de superficie	20
1.2.2.5 Visualización de modelos volumétricos	23
1.2.3 Interacción con entornos 3D	26
1.3 Aportaciones	28
2 Trabajos previos	31
2.1 Métodos de interacción háptica	32
2.2 Modelos deformables con respuesta visual	36
2.3 Modelos deformables con respuesta visual y háptica	43

3	Formalización del problema	49
3.1	Material	51
3.1.1	Modelo de muestras	51
3.1.2	Reconstrucción del modelo	52
3.1.3	Interpretación del modelo	54
3.2	Herramienta	54
3.3	Espacio	56
4	Teoría de la elasticidad y la plasticidad	59
4.1	Deformación	60
4.2	Tensión	63
4.3	Teoría de la elasticidad lineal	68
4.3.1	Ecuación constitutiva	68
4.3.2	Planteamiento del problema elástico lineal	70
4.4	Teoría incremental de la plasticidad	71
4.4.1	En una dimensión	72
4.4.2	En tres dimensiones	76
5	Fuerza	81
5.1	El flujo de trabajo háptico	83
5.2	Adaptación del marco teórico al entorno de simulación	86
5.3	Estimación de los campos tensoriales	88
5.3.1	Aproximación del campo de deformaciones	89
5.3.2	Inferencia del campo de tensiones	90
5.4	Resultados	96
5.5	Experimentos	98
5.6	Conclusiones	101
6	Plasticidad	103
6.1	Aislamiento del efecto plástico	104
6.2	Viabilidad del modelo teórico	105
6.3	Aproximación heurística de la deformación plástica	107
6.3.1	Traducción de los desplazamientos al dominio del vo- lumen	109
6.3.1.1	Equivalencia de los desplazamientos	110

6.3.1.2	Equivalencia de las deformaciones	113
6.3.2	Caracterización de la zona elástica y la zona plástica	114
6.3.2.1	Función de fluencia y endurecimiento	115
6.3.2.2	Resolución de la carga plástica	117
6.3.3	Estimación del campo de desplazamientos	119
6.3.3.1	Análisis de los cambios en la superficie	120
6.3.3.2	Obtención de la superficie deformada	122
6.3.3.3	Primera aproximación de la deformación interna	122
6.3.3.4	Heurística de deformación: <i>algoritmo de dispersión recursiva</i>	127
6.3.3.5	Segunda aproximación de la deformación interna	130
6.3.3.6	Heurística de deformación: <i>operador de suavizado</i>	131
6.3.3.7	Tercera aproximación a la deformación interna	134
6.3.3.8	Heurística de deformación: <i>operador de desplazamiento</i>	135
6.3.3.9	Heurística de deformación: <i>operador de desplazamiento orientado</i>	137
6.3.3.10	Heurística de deformación: <i>operador de desplazamiento orientado con prioridad</i>	140
6.3.3.11	Mejora del efecto visual	142
6.4	Resultados	143
6.5	Experimentos	145
6.5.1	Direccionalidad del desplazamiento de material con los vectores de dispersión	146
6.5.2	Comparativa de los tres algoritmos propuestos	148
6.5.3	Tiempo de convergencia del operador de desplazamiento	150
6.5.4	Variación de los tiempos de relajación con el tamaño de la herramienta	153
6.5.5	Variación de los tiempos de modificación de la superficie con el tamaño de la herramienta	155

6.5.6	Cambios topológicos	159
6.6	Conclusiones	160
7	Elasticidad	163
7.1	Aislamiento del efecto elástico	164
7.2	Viabilidad del modelo teórico	164
7.3	Aproximación heurística de la deformación elástica	165
7.3.1	Superestructura de deformación elástica	166
7.3.2	Estimación de la deformación elástica	168
7.4	Resultados	171
7.5	Experimentos	178
7.5.1	Sobre el rendimiento de la deformación elástica	178
7.5.2	Sobre la unión con el método de deformación permanente	180
7.6	Conclusiones	181
8	Conclusions	185
8.1	Achievements	186
8.2	Comparison with Previous Works	188
8.3	Future Work and Outlook	190
A	Implementación	193
A.1	La aplicación <i>VolumeWorks</i>	193
A.1.1	Interfaz de usuario	194
A.1.2	Arquitectura multitarea	196
A.2	Las rutinas hápticas	198
A.3	Técnica de visualización	200
A.3.1	Implementación del método de visualización	200
A.3.2	Función de transferencia	202
A.3.3	Interpolación	203
A.4	Superestructura de deformación elástica	203
A.5	Conclusiones	206
B	Cinemática háptica	207
B.1	Introducción	208
B.2	Trabajos previos	210

B.3	Cinemática	211
B.4	Métodos para el cálculo de la inercia	212
B.4.1	Aceleración instantánea	213
B.4.2	Simplificación de trayectoria	214
B.4.3	Trayectoria promedio	215
B.4.4	Aceleración promedio	216
B.4.5	Suavizado estático de trayectoria	217
B.4.6	Suavizado dinámico de trayectoria	219
B.5	Conclusiones	220
C	Notas de Copyright	223
	Índice de símbolos	225
	Índice alfabético	229
	Bibliografía	231

Prólogo

El presente trabajo tiene una considerable carga de notación y herramientas matemáticas. La formalización de las estructuras y procedimientos que se usan, así como el análisis de la parte de la mecánica que abstrae el comportamiento de los sólidos deformables, conlleva un constante uso de definiciones de conjuntos, funciones, trigonometría, cálculo vectorial y tensorial y derivación e integración de campos escalares y vectoriales.

Si bien se requiere cierta familiaridad con las herramientas citadas para seguir con detalle el desarrollo, se han procurado explicaciones cualitativas de todos los procesos descritos, de forma que no sea estrictamente necesario el análisis formal para comprender las ideas expuestas.

Se ha procurado en cualquier caso mantener una notación homogénea para facilitar la identificación de los símbolos. Se usarán con este fin los siguientes convenios:

Escalares Las variables y constantes unidimensionales se representan con minúsculas sin negrita:

$$k \in \mathbb{R}$$

Puntos Los puntos del espacio se representarán con letras mayúsculas, sin negrita ni cursiva, como por ejemplo:

$$P \in \mathbb{R}^3$$

Vectores Se usará la notación común con una flecha a la derecha sobre un carácter en cursiva, como en la expresión:

$$\vec{F} = k\vec{x}$$

Se representará habitualmente el módulo de un vector con el mismo carácter pero con el formato de un escalar:

$$u = \|\vec{u}\|$$

Los vectores unitarios se representarán con el gorro:

$$\hat{u} = \frac{\vec{u}}{\|\vec{u}\|}$$

Siendo además usual identificar los componentes de los vectores como escalares con subíndices:

$$\vec{u} \equiv (u_x, u_y, u_z)$$

Tensores Los tensores se representarán con caracteres sin cursiva y en negrita. Todo el cálculo tensorial usado en este trabajo se realiza sobre el espacio euclídeo \mathbb{R}^3 , por lo que en toda definición se proporcionará el orden del tensor, conociéndose así de forma inmediata el número de componentes. En algunos casos pueden proporcionarse las componentes en forma matricial:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}$$

siendo sus componentes representados como escalares. Cuando se realizan productos tensoriales pueden también definirse los tensores mediante los índices de sus componentes, como en la expresión:

$$\sigma_{ij} = \mathbb{C}_{ijkl} \varepsilon_{kl}$$

Conjuntos Los conjuntos se representarán en cursiva y sin negrita, quedando clara su naturaleza en su definición con los corchetes curvos usados habitualmente en álgebra, usándose la barra vertical $|$ para separar la condición si se trata de un conjunto definido de forma

implícita:

$$Q = \{ q_i \mid i > 0 \}$$

Funciones Las funciones se notarán sin negrita y cursiva, seguido de los paréntesis habituales entre los que se encerrarán los parámetros de su dominio, como por ejemplo en la definición:

$$f(x, y, z) : \mathbb{R}^3 \rightarrow \mathbb{Z}$$

Campos Los campos se representarán con el formato que represente su magnitud asociada. Así, un campo escalar se representará con el formato de un escalar, seguido de los parámetros espaciales y/o temporales encerrados entre paréntesis. Un campo tensorial se notará por ejemplo:

$$\mathbf{t}(\mathbf{x}, t)$$

usándose habitualmente el nombre del campo para referirse a una magnitud individual del mismo (ej.: en el campo anterior \mathbf{t} será un tensor concreto para un punto \mathbf{x} en un instante t determinado).

Cualquier otra notación específica se definirá claramente en el texto. Como referencia y ayuda para la comprensión del texto, se puede encontrar un índice de nomenclatura al final del libro, donde se pueden consultar los símbolos usados y la página en la que se puede encontrar su definición.

Capítulo 1

Introducción

Esta tesis responde a una problemática concreta que apareció cuando, en uno de los proyectos del Grupo de Investigación en Informática Gráfica de la Universidad de Granada, se pretendió dotar de interacción táctil a la herramienta de modelado en la que se estaba trabajando.

El *modelado virtual*, tal y como se planteó en aquel proyecto, es un conjunto de técnicas que facilitan tareas o ayudan al usuario (o artista, o modelador) durante el proceso creativo, ya sea apoyando en alguna tarea concreta u ofreciendo una simulación total o parcial del mismo. En concreto la herramienta que se desarrollaba en el seno de aquel proyecto trataba de optimizar la simulación del trabajo de escultura en materiales arcillosos, trabajo que dio sus frutos en forma de interesantes resultados y publicaciones científicas [LTV08].

Quedó pendiente de resolver, no obstante, la inclusión de un método realista de interacción táctil, pues la dirección principal de aquella investigación era la optimización de las operaciones con la arcilla y no trató con profundidad esta cuestión. La revisión de los trabajos relacionados con esta problemática pone de manifiesto la falta de soluciones lo suficientemente realistas o lo suficientemente fieles a la interacción real con un bloque de arcilla. Esto es lo que motiva el inicio de este trabajo, que llevará en última instancia al diseño de un nuevo método adaptado a las necesidades que requiere la simulación anteriormente descrita.

En este trabajo se describe el camino recorrido en el proceso de creación

de una nueva solución para la interacción con **sólidos deformables**.

Empezando por los conceptos y definiciones básicas sobre las que está construido, y pasando por la formalización de los algoritmos y datos que se manejan, a lo largo de este documento se mostrarán los pormenores del diseño, implementación y prueba del fruto de esta tesis doctoral.

1.1 Motivación

El objetivo que se persigue es crear una técnica *eficaz y creíble* para deformar un sólido deformable con una respuesta interactiva. Teniendo siempre presente la limitación de recursos disponibles en el entorno de simulación, debe llegarse a un compromiso entre realismo y eficiencia.

La motivación es común tanto para la visualización de la deformación como para la dimensión táctil de la simulación. Especialmente delicada será la construcción del método táctil, el cual, por lo exigente de su frecuencia de actualización, requerirá un especial cuidado en la *optimización de los recursos*. El sistema que genere el tacto deberá proporcionar una sensación natural y robusta, de forma que el usuario pueda tocar y deformar el modelo con el suficiente realismo sin que haya lugar a paradas o discontinuidades anormales.

En cualquiera de los casos la simulación deberá proporcionar un alto sentido de la *interactividad*, de forma que se minimice el tiempo de respuesta necesario para que el sistema provea los cambios correspondientes ante cualquier acción del usuario, obteniéndose así una sensación global de agilidad e inmediatez.

Deben cumplirse estrictamente estos requisitos con vistas a que las técnicas aquí desarrolladas permitan su inclusión en aplicaciones prácticas futuras, creando un método de simulación útil que no se quede en la mera comprobación de los conceptos desarrollados.

1.2 Contexto del trabajo

Los gráficos generados por ordenador se han convertido en parte de nuestra vida cotidiana. Videojuegos, maquetas virtuales, películas de animación...

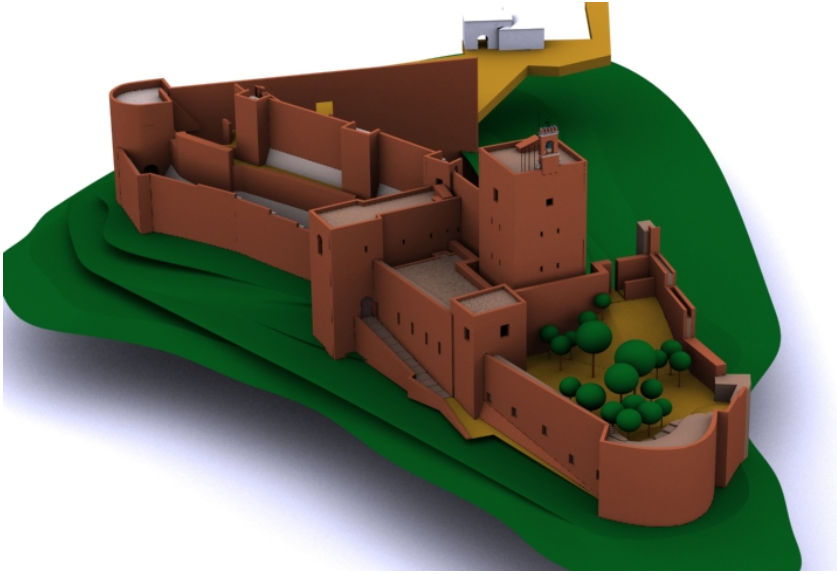


Figura 1.1 – Modelo virtual de la Alhambra de Granada. Los modelos que representan la realidad pueden hacerlo con más o menos detalle. (Imagen cortesía de Lorena Palacios, Miriam Palacios y Sergio Villegas, Facultad de Bellas Artes, Universidad de Granada).

con mayor o menor grado de realismo, las escenas generadas por ordenador están presentes en muchos contextos familiares para cualquiera. Sin embargo, aunque hablamos de *gráficos* de forma genérica, existe un conjunto heterogéneo de técnicas de manipulación y representación de los datos.

El concepto fundamental cuando hablamos de gráficos por ordenador es el de **modelo**. Un *modelo virtual* o sencillamente *modelo* es la descripción abstracta o *copia* de un objeto o conjunto de objetos, reales o imaginarios, con la que trabaja un ordenador. Por ejemplo, podemos tener el modelo de una vasija si alguien se tomara la molestia de *construirla* en algún formato que un ordenador sea capaz de procesar. O podemos tener, y de hecho es muy habitual, modelos de edificios que aún no se han construido, cuando un arquitecto hace los diseños del mismo con la ayuda de un ordenador. Una fotografía digital también es un modelo: es una representación de una escena, que provee de cierta información sobre la posición y el color de los objetos representados.

De esta explicación anterior surge el concepto de **dimensión del modelo**. En términos generales, la *dimensión* del modelo es la misma del espacio

Capítulo 1. Introducción

sobre el que está definido. Lo más común es que los modelos sean *bidimensionales* (esto es, **2D**, o planos) o *tridimensionales* (**3D**). Un modelo 2D es por ejemplo la fotografía antes aludida. Es una estructura plana, y por tanto tiene definidas tan sólo dos direcciones: arriba-abajo e izquierda-derecha. Por contra, si disponemos del modelo de un edificio completo, es forzosamente un modelo 3D: tiene altura, anchura y profundidad, pudiendo recorrerse en cualquiera de las tres direcciones. Si bien nuestro mundo es tridimensional y cualquier representación fidedigna de un objeto real debería también serlo, es posible que por simplicidad o cualquier otro motivo se realicen representaciones 2D de la realidad (como por ejemplo sucede en arquitectura cuando, en lugar de diseñar un edificio completo, se dibujan únicamente los planos).

El proceso de **generación** del modelo es un aspecto fundamental, pues el modelo debe contener absolutamente toda la información que va a ser visualizada o procesada. En algunos casos el proceso de creación es el fin en sí mismo (por ejemplo, cuando alguien diseña algo con ayuda de un ordenador), y es por ello que existen innumerables herramientas especializadas construidas con este fin. Estas aplicaciones se conocen, de forma genérica, como herramientas CAD (del inglés *Computer Aided Design*, diseño asistido por computadora). Un conocido ejemplo es el *AutoCAD*[®], una aplicación de diseño industrial que se ha convertido en el estándar de facto entre los arquitectos. Permite, entre otras muchas tareas, la creación de figuras planas con rapidez y precisión.

Pero el proceso de creación del modelo no tiene que ser necesariamente manual. Existen dispositivos que permiten al ordenador *reconocer* la forma o propiedades de los objetos reales, de forma que, ya sea mediante un procedimiento totalmente automático o parcialmente automático, se puede generar un modelo de los mismos. Es el caso de las máquinas de *tomografía axial computerizada* (TAC). Estos dispositivos, usados habitualmente en diagnóstico médico, son capaces de obtener las propiedades internas del cuerpo analizado, y generar automáticamente un modelo 3D del mismo.

Conviene tener presente cuando se trabaja con gráficos que el ordenador dispone únicamente de la información contenida en el modelo. Dicho de otro modo: sólo se puede mostrar de forma fidedigna la información que



Figura 1.2 – Modelo 3D de un pie escaneado mediante tomografía axial computarizada. En esta imagen ha sido procesada para darle color y hacer translúcido el músculo, de forma que se aprecie el hueso. (Modelo cortesía de Philips Research, Hamburgo).

se proporciona; cualquier especulación más allá del nivel de detalle ofrecido será pura invención y no tiene por qué corresponderse con la realidad. (Esto, que parece una afirmación trivial, no parece ser tan evidente en algunas producciones cinematográficas de ciencia ficción, que aseguran poder ampliar una fotografía borrosa de forma indefinida hasta dar con la cara del asesino). Es por esto que deberemos buscar, generar, adquirir o construir un modelo que se adapte a nuestras necesidades, dotándolo de toda la información necesaria de cara a su visualización o procesamiento posterior. Sean 2D o 3D, los modelos pueden contener y representar información de lo más dispar: desde la información de color de una escena capturada por una lente que ofrece una fotografía, hasta la detallada información espacial que puede ofrecer un modelo tridimensional de una habitación.

En lo que sigue se expondrán una serie de características más concretas sobre los modelos y las formas más comunes de representarlos mediante un computador.

1.2.1 Representaciones

Cuando un modelo abstracto se traduce en un conjunto concreto de datos en un ordenador se habla de la **representación** del modelo [M87]. Dependiendo de la naturaleza del modelo una representación puede resultar más idónea que otra para un determinado propósito. Las diferencias entre las distintas representaciones están lejos de ser sutiles y su estudio es fundamental para poder elegir la que mejor se adapte al problema enunciado. A continuación, y sin ánimo de realizar un estudio exhaustivo, se muestra una clasificación general de las mismas que resultará de especial interés para el objetivo que persigue este trabajo. En primer lugar se estudiarán las representaciones más comunes para los modelos 2D, analizándose posteriormente sus generalizaciones en las tres dimensiones.

1.2.1.1 Modelos bidimensionales

Cuando se ha de construir un modelo 2D se puede definir la información del mismo de dos formas básicas, lo cual lleva a dos tipos de representación:

De frontera Son aquellas en las que se almacena información que se refiere exclusivamente a los contornos o bordes de los objetos representados. Típicamente la información que se guarda es puramente geométrica, ya sea en forma de líneas, puntos, curvas, funciones paramétricas o ecuaciones. La representación se compone pues de una serie de primitivas matemáticas que indican el límite entre el interior y el exterior. Por ejemplo, para guardar una circunferencia se almacenaría la posición de su centro y su radio (tal y como se puede apreciar en la [Figura 1.3a](#)), y para especificar un triángulo sólo habría que determinar la posición de sus tres vértices. En este grupo entran los gráficos vectoriales con los que trabajan los programas como *Inkscape* o *Adobe® Illustrator®*.

Completas Son las representaciones que ofrecen una visión conjunta del objeto, ofreciendo información no sólo del contorno de lo representado si no del plano en su totalidad. Éstas pueden ser además:

Continuas Para componer una representación continua se especi-

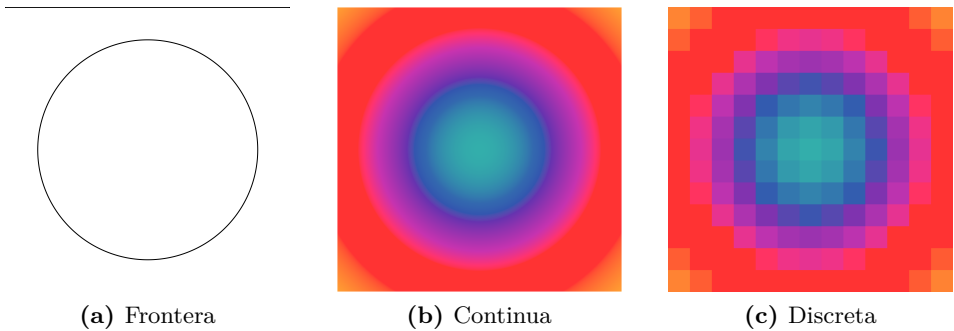


Figura 1.3 – Representaciones 2D. A la izquierda, representación vectorial de un círculo, que se define con su centro y radio. En el centro, una representación continua que modela la distancia al origen de coordenadas de todos los puntos del plano ($f(x, y) = x^2 + y^2$). A la derecha, gráfico ráster, compuesto por un número finito de píxeles ordenados en retícula.

can por norma general un conjunto de funciones matemáticas en el plano, las cuales, al ser evaluadas en un punto concreto, generan el valor o valores asociado al modelo en ese punto (ver [Figura 1.3b](#)). Una representación plana continua con un único valor para cada punto del espacio equivale a lo que en física se conoce como un *campo escalar*, y como un *campo vectorial* si el valor asociado a cada punto es múltiple. Este tipo de representaciones garantizan en cualquier caso un valor para todos los puntos del plano o un subconjunto acotado del mismo (al contrario que los modelos vectoriales, en los que todo el espacio, salvo los puntos por los que pasa la superficie, está vacío).

Discretas Son por definición un conjunto de *muestras*, ordenadas o no, que especifican con mayor o menor precisión las propiedades de lo representado en el plano. Dentro de esta clasificación entran los gráficos *ráster*, en los que las muestras están ordenadas en una retícula cuadrada en el plano con un valor (sencillo o compuesto) asociado a cada una de ellas. El ejemplo más paradigmático de gráfico raster es una imagen digital: ésta se compone de un conjunto de puntos llamados *píxeles*, que contienen información sobre el color de la imagen en ese punto. Cada píxel contiene tres valores, los cuales indican la proporción de

colores primarios que debe mezclarse para generar el color asociado. Los píxeles, al ordenarse en una cuadrícula, componen la imagen completa, lo cual se ilustra en la [Figura 1.3c](#). Si son lo suficientemente pequeños y densos la imagen dará una sensación de continuidad, a pesar de estar compuesta en última instancia por puntos discretos sin conexión entre sí.

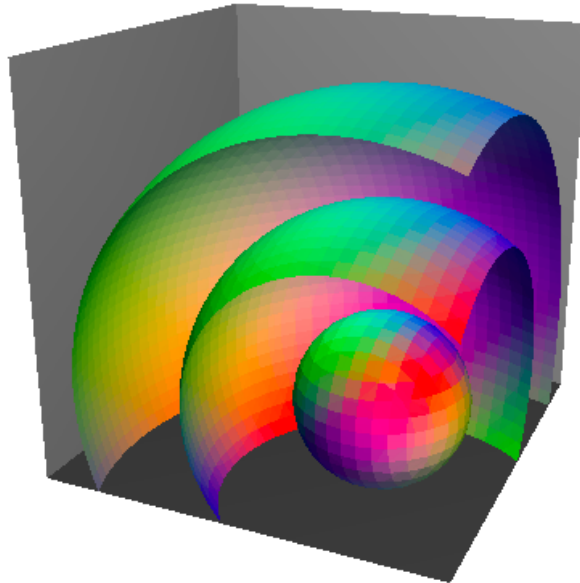
1.2.1.2 Modelos tridimensionales

De forma análoga a sus equivalentes 2D, las representaciones de los modelos 3D pueden fabricarse con distintos tipos de primitivas y elementos, resultando en la siguiente casuística:

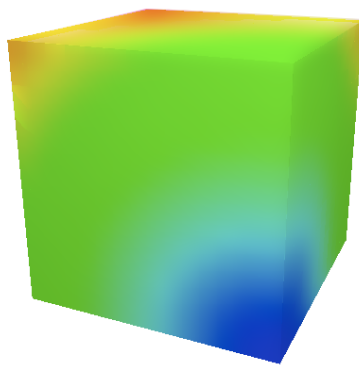
De superficie Son aquellas representaciones que, partiendo de la misma idea que se desarrolló al describir las representaciones de frontera, almacenan únicamente la información de contorno de un objeto 3D ([Figura 1.4a](#)). En el caso tridimensional las primitivas son, además de líneas y puntos, planos (si bien la tecnología actual reduce este concepto al triángulo, componiéndose todas las primitivas de un determinado número de estos). También se pueden especificar los contornos con superficies paramétricas o cualquier otra clase de función matemática que determinen unívocamente de forma implícita una superficie. Para representar un objeto es necesario pues *modelar* su contorno con planos y curvas hasta que la forma obtenida sea lo suficientemente parecida al objetivo.

Volumétricas Serán representaciones volumétricas las que ofrezcan información detallada sobre todos los puntos de una zona concreta del espacio, independientemente de los contornos y las fronteras. Pueden ser además:

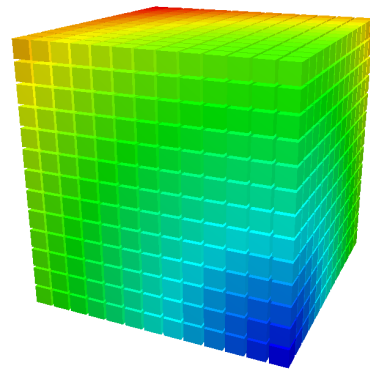
Continuas La generalización 3D de sus homólogas planas se realiza usando funciones matemáticas en el espacio tridimensional (en tres variables). Igualmente estas representaciones son equivalentes a un campo escalar si son monovaluadas, y a un campo vectorial si cada punto del espacio posee múltiples propiedades. Un ejemplo se muestra en la [Figura 1.4b](#).



(a) De superficie



(b) Continua



(c) Discreta

Figura 1.4 – Los tres tipos de representaciones 3D. En la parte superior, una representación de superficie. Se puede apreciar que únicamente aparecen *contornos*, que representan los límites del objeto representado (en esta imagen, esferas concéntricas). Abajo a la izquierda, una representación volumétrica continua en el que todos los puntos del espacio tienen el valor dado por la función que define el modelo (en este caso $f(x, y, z) = \sqrt{x^2 + y^2 + z^2}$). Abajo a la derecha se muestra una representación volumétrica discreta, en el que se aprecian las muestras ordenadas en el espacio tridimensional (representadas por cubos separados por un pequeño espacio intermuestral).

Discretas Las representaciones volumétricas discretas se componen, al igual que sus homólogas bidimensionales, de una serie de muestras en el espacio, esta vez, tridimensional. Es también usual referirse a ellas como *modelos de elementos finitos* (o por las siglas FEM, del inglés *Finite Elements Model*). Cuando las muestras se ordenan de forma homogénea, conforman una estructura tridimensional que puede ser vista como una sucesión de cortes planos, apilados uno tras otro. Así se compone un cubo de muestras que, unidas, conforman la imagen tridimensional (ver [Figura 1.4c](#)). Como evolución natural del píxel aparece el *vóxel*, que es, en términos generales, el nombre que se le da a las muestras cuando se trabaja con este tipo de estructuras.

1.2.1.3 Análisis comparativo

Es importante, dado lo dispar de los distintos tipos de representación que pueden construirse para un mismo modelo tridimensional, elegir correctamente la tipología más adecuada para cada problema. Para ello resulta conveniente analizar los puntos fuertes y débiles de cada uno de los tipos. La elección de uno u otro tipo puede determinar incluso la propia solución del mismo. En lo que sigue se discutirán las ventajas e inconvenientes de las tres clases descritas, las cuales se deducen, de forma más o menos evidente, de sus propias definiciones.

Las representaciones de superficie cuentan con la ventaja inmediata de ser livianas y, adicionalmente, de contar con un hardware específico para su procesamiento. Por una parte es evidente que es más económico en términos de almacenamiento guardar tan sólo la superficie de un modelo (por contraposición a almacenar toda la información interna tal y como hacen las representaciones volumétricas). Por otra el manejo de los gráficos de superficie es extremadamente sencillo pues el hardware gráfico actual está orientado a este tipo de estructuras. Las tarjetas gráficas ofrecen interfaces relativamente sencillas para trabajar con ellas, y cada vez integran procesadores más potentes y optimizados para visualizar y procesar listas de triángulos, puntos y líneas. Su mayor desventaja es precisamente lo que desde otro punto de vista es su mayor virtud: no es posible representar con

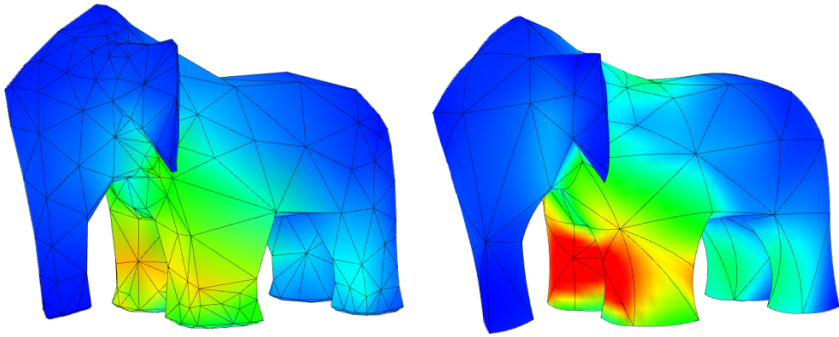


Figura 1.5 – Dos modelos de elementos finitos (usados por Mezger et al. [MTPS08] para el cálculo de tensiones en sólidos plásticos), en el que las muestras están situadas en puntos arbitrarios del espacio.

ellas las propiedades interiores de los objetos representados. Su naturaleza inspirada en el contorno o *frontera* de los objetos implica necesariamente una discontinuidad en las zonas adyacentes a los planos que la definen: se definen los límites o superficies del modelo; el resto del espacio está necesariamente vacío y no se dispone de información intermedia.

Las representaciones volumétricas continuas cuentan con un handicap impuesto por su propia definición: es necesario, para construir una representación de estas características, encontrar un conjunto de funciones en el espacio que se asemeje a la entidad que se quiere representar. Esta tarea es especialmente compleja cuando es necesario manipular las propiedades internas de los objetos modelados, pues es necesario controlar la evolución de todos los puntos del espacio con funciones por partes o superpuestas. Su procesamiento, además, implica costosas evaluaciones que consumen tiempo de computación. Su mayor ventaja es que todas las representaciones son implícitas y muy ligeras en términos de espacio: para definir un modelo tan sólo es necesario especificar la lista de funciones que lo componen y en qué partes del espacio están definidas cada una.

Las representaciones discretas se caracterizan, principalmente, por requerir ingentes cantidades de almacenamiento para representar hasta los modelos más simples. En su versión tridimensional, es necesario guardar un *cubo* de muestras, por lo que los requerimientos de espacio crecen de forma potencial cada vez que se incrementa el alto, ancho o fondo del mo-

delo. Un modelo con muestras ordenadas en una retícula cúbica de lado 100 (el equivalente a una imagen bidimensional de 100×100 píxeles) implica el almacenamiento de 1.000.000 de muestras. De forma inmediata se comprueba que con un ligero incremento del lado se dispara el número de millones de muestras que es necesario almacenar. Existen técnicas de compresión e indexación que moderan el tamaño de los modelos, si bien tiene sentido aplicarlas únicamente cuando existen muchas zonas vacías u homogéneas en los mismos. También es posible, en algunos casos, evitar el muestreo regular de forma que sólo se especifican muestras en zonas de interés (lo cual, si bien requiere guardar la posición de cada muestra, puede conllevar una reducción considerable del tamaño del modelo). En la [Figura 1.5](#) se puede ver un modelo en el cual las muestras se organizan en una estructura de tetraedros, siguiendo una distribución arbitraria. En cualquier caso es evidente la mayor complejidad de los modelos volumétricos, con las dificultades obvias que de cara a su manipulación esto puede suponer. Con todo lo expuesto es posible, no obstante, que las ventajas puedan, en algunos casos, compensar estas trabas. Las representaciones volumétricas discretas ofrecen una información detallada de las propiedades interiores de los objetos que representan. No se establece ninguna división artificial entre *exterior*, *frontera* o *interior*, al contrario de lo que sucede en las representaciones de superficie. Las muestras, organizadas en el espacio, ofrecen una visión continua del modelo, lo cual es crucial si la aplicación que las tratará necesita para funcionar no sólo conocer los contornos, sino también cómo es el interior de lo representado.

1.2.2 Modelos volumétricos discretos

La anterior exposición justifica pues el uso de una representación volumétrica para la aplicación concreta que se propone en este trabajo. No en vano la intención es simular no sólo los posibles efectos de la interacción del usuario (esto es, el cambio en las propiedades geométricas del modelo o *deformación*), sino también las tensiones internas del objeto simulado, con el fin último de producir una sensación táctil coherente con el efecto real. Los estudios realizados por Moore et al. [[MM07](#)] y Delingette [[Del98](#)] apoyan esta elección: en ambos se reconoce que las representaciones de superficie pre-

sentan grandes carencias a la hora de representar cuerpos deformables, pues las propiedades internas son cruciales para una simulación realista. Las representaciones discretas se adaptan además de forma natural a los sistemas de partículas y a los modelos formales de mecánica que se expondrán más adelante, por lo que resultan más adecuados para el problema aquí enunciado. En lo que sigue se describirán varias propiedades y aplicaciones de las representaciones volumétricas discretas que serán de utilidad para el desarrollo de este trabajo, siendo algunas de ellas especialmente convenientes, tal y como se verá, de cara a la simulación de un proceso físico.

Cabe aclarar que, aunque en un análisis formal como el que se ha realizado existe una diferencia clara entre el modelo como entidad matemática abstracta y su representación, una vez se ha elegido una representación concreta la frontera entre los dos conceptos pierde cierta relevancia. Es habitual por tanto hablar de forma indiferente de *modelo*, *representación* o *gráfico* volumétrico, pudiéndose entender en el contexto de este trabajo que cualquiera de las expresiones anteriores se refiere a la representación discreta que se ha erigido como óptima para los fines que se persiguen.

1.2.2.1 Organización de las muestras

Como se ha podido entrever en las anteriores referencias a los modelos volumétricos existe una gran variedad de métodos de distribución de las muestras. Se puede establecer una clasificación general si se considera la organización espacial de las muestras que los componen [LS08, VA02]. La distribución en el espacio puede ser por tanto:

No estructurada Las muestras se sitúan en puntos arbitrarios del espacio sin ningún orden preestablecido, como en el ejemplo de la [Figura 1.6c](#). Otro ejemplo se puede apreciar en la [Figura 1.5](#) en la que, si bien las muestras se sitúan en los vértices de una estructura de tetraedros apilados, su posición no presenta ningún tipo de regularidad.

Estructurada Las muestras siguen algún tipo de estructura ordenada según un sistema de coordenadas concreto, por ejemplo, polar o cartesiano. La estructura puede ser:

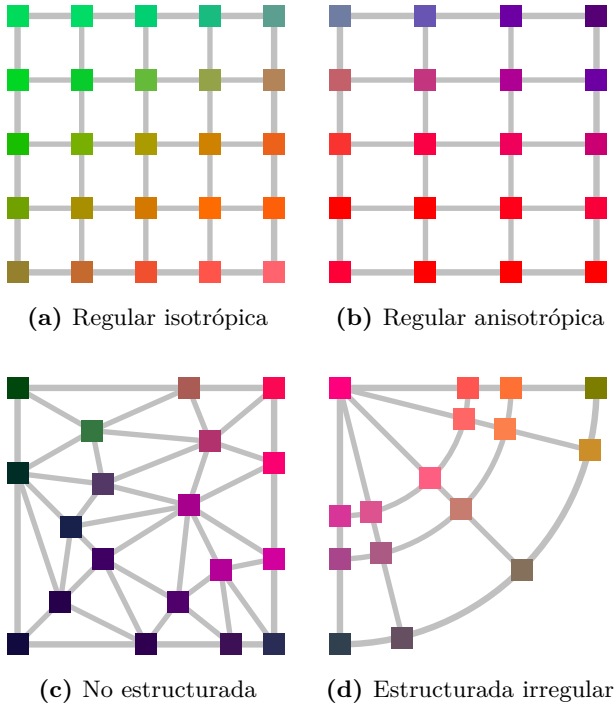


Figura 1.6 – Diferentes distribuciones de muestras para un modelo bidimensional. Las muestras se representan con cuadrados coloreados. Arriba se muestran dos distribuciones estructuradas y regulares en un sistema cartesiano. La estructura 1.6a es además isotrópica, puesto que en los dos ejes la distancia intermuestral es la misma, mientras que en 1.6b se observa que los espaciados vertical y horizontal son distintos. La distribución de 1.6c es claramente aleatoria y no sigue ningún orden, mientras que en 1.6d se aprecia una organización en un sistema polar, si bien las distancias entre muestras no son constantes en ninguno de los dos ejes.

Irregular Las muestras, si bien se organizan siguiendo las direcciones del sistema de coordenadas, no siguen un patrón concreto a lo largo de los ejes de distribución, por lo que la distancia entre las mismas puede variar de forma arbitraria (ver Figura 1.6d).

Regular Existe un patrón claro y definido que marca la distancia entre las muestras en el sistema de coordenadas, con lo que además se puede representar su posición de forma implícita (típicamente con una expresión matemática). Si la estructura está definida sobre un espacio cartesiano se dice que es *rectilínea*, pudiendo además ser *isotrópica* si el espacio intermuestral es constante en

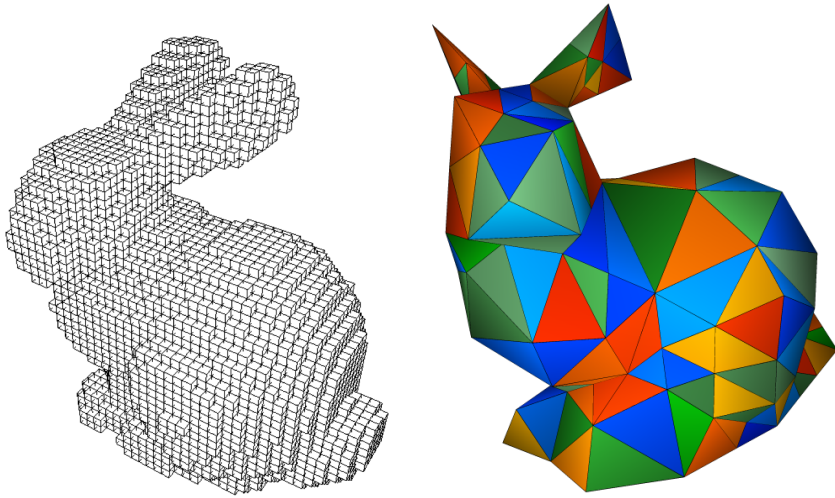


Figura 1.7 – A la izquierda, modelo de muestras organizadas en cubos del conocido *Stanford Bunny* (Shen y Huang 2008). A la derecha, el mismo modelo con muestras organizadas en tetraedros (Mezger et al. 2008).

todas las direcciones o *anisotrópica* si no lo es (ver Figuras 1.6a y 1.6b, respectivamente).

La elección de uno u otro tipo de estructura dependerá del problema concreto que se vaya a resolver, de la naturaleza del objeto real y de las posibilidades técnicas. Las distribuciones regulares dan una visión uniforme del modelo, mientras que las irregulares o las no estructuradas permiten situar más muestras en zonas de alto interés y realizar un muestreo menos denso en el resto.

Es posible *agrupar* las muestras cercanas en poliedros disjuntos si se consideran éstas como vértices [LS08]. De este modo se puede entender la estructura como una unión continua de estos elementos, en la que dos poliedros adyacentes comparten una y sólo una cara. En la Figura 1.7 se muestra el mismo modelo con dos muestreos y dos organizaciones distintas. En una se ha realizado un muestreo regular e isotrópico que resulta en una estructura de cubos apilados, mientras que en la otra se ha realizado un muestreo no estructurado en el que las muestras se organizan en tetraedros conectados por las caras. Si se obvian las caras de los poliedros y únicamente se tienen en cuenta los vértices se observa que las muestras conforman una

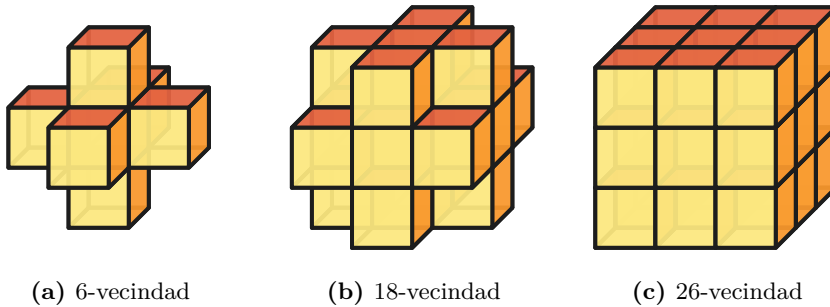


Figura 1.8 – Diferentes grupos de vecinos en un modelo volumétrico tridimensional en rejilla regular rectilínea, en la que cada muestra se representa como un cubo. En la 1.8a aparecen las 8 muestras vecinas que comparten una cara con la muestra central. En 1.8b se añaden los 12 vecinos que comparten una arista, conformando la 18-vecindad. Los 8 vecinos que comparten sólo un vértice completan la 26-vecindad que se ilustra en 1.8c.

red conectada o **rejilla**, siendo este último un término común para referirse a la estructura.

Debe notarse que las estructuras no tienen por qué representarse de forma explícita, siendo estas en la mayoría de los casos una mera abstracción que simplifica la localización y manipulación de las muestras.

1.2.2.2 Vecindades y distancias

Cuando las muestras se encuentran ordenadas en el espacio se pueden establecer lo que se denominan **relaciones de vecindad** entre las muestras. Esta relación depende estrictamente de la disposición de las muestras en el espacio, y se definirá según el tipo de *cercanía* o *adyacencia* que una muestra presente respecto del conjunto de muestras circundantes.

Tómese por ejemplo un modelo bidimensional como el que se encuentra en la [Figura 1.6a](#), en el que las muestras se encuentran ordenadas y equidistantes en el plano. Cada muestra del modelo tiene cuatro muestras que son inmediatamente adyacentes (norte, sur, este y oeste) y cuatro más en las diagonales (noreste, noroeste, sureste y suroeste), por lo que se puede definir una vecindad básica de cuatro vecinos y una extendida de ocho. En un modelo tridimensional de muestras organizadas en rejilla regular como el de la [Figura 1.4c](#) los vecinos se pueden clasificar en tres grupos si se

consideran las muestras como cubos apilados en el espacio: seis vecinos con los que se comparte una cara (izquierda, derecha, arriba, abajo, adelante y atrás), doce con los que se comparte una arista, y ocho más con los que se comparte un vértice, para sumar un total de veintiséis vecinos. Se pueden de hecho definir sobre este concepto varios tipos de vecindad, tal y como se muestra en la [Figura 1.8](#).

Es frecuente que sobre esta definición de vecindad se construyan conceptos adicionales tales como el de *distancia en muestras* de un punto a otro. Por ejemplo, es común referirse a la *distancia Manhattan* entre dos muestras en modelos de rejilla regular rectilínea como los que se han visto en los ejemplos anteriores. La distancia Manhattan se define como el número mínimo de muestras que es necesario recorrer para llegar desde una muestra a otra, no permitiéndose avanzar pasando de un vecino a otro en diagonal.

Otra operación que se pueden realizar con una vecindad bien definida es el cálculo del gradiente. En términos físicos, el gradiente en un punto de un campo escalar es la dirección en la que los valores crecen más rápidamente. Trasladado a un modelo volumétrico discreto y calculado sobre una muestra, el gradiente indicará la dirección en la que el crecimiento del valor de propiedad es mayor, y dará además una medida de dicho crecimiento. En las rejillas rectilíneas el cálculo se puede realizar de forma sencilla calculando la diferencia entre los vecinos opuestos de la 8-vecindad.

De la misma forma que sucede con el gradiente, la simplicidad de la vecindad en las rejillas rectilíneas facilita otras muchas operaciones de *filtrado*, tales como suavizado o detección de bordes, y que se reducen en la mayor parte de los casos a realizar operaciones aritméticas en la 26-vecindad y cambiar el valor de la muestra central.

1.2.2.3 Continuidad del modelo

A pesar del carácter discreto de las representaciones discretas (por definición basados en un conjunto de muestras) se puede asumir, bajo ciertas premisas, que la estructura es *continua*.

Todo muestreo, por definición, representa el valor de propiedad en un punto concreto del espacio, o el promedio de una zona delimitada y conocida. Se considera por tanto que las muestras son *puntuales* y por tanto

Capítulo 1. Introducción

adimensionales. El espacio intermuestral está consecuentemente vacío y no es posible, en principio, conocer los valores reales de otro punto que no sean exactamente los que contienen muestras (tal y como ya se ha discutido anteriormente). Sin embargo, es posible realizar una *reconstrucción* del modelo para poder trabajar en un espacio continuo de valores. Siendo siempre conscientes de que esto es una recreación arbitraria y que no tiene necesariamente que coincidir con la realidad, se puede *rellenar* el espacio entre muestras de dos formas básicas:

Valor más cercano Si se asigna a todos los puntos del espacio el valor de la muestra más cercana se obtendrá un modelo con zonas homogéneas y saltos bruscos en las fronteras (puntos equidistantes a dos o más muestras). Esta partición del espacio se conoce como formalmente como *teselación de Voronoi*, y se muestra gráficamente para un modelo 2D con muestras organizadas en rejilla en la [Figura 1.9c](#). El resultado es una distribución de valores continua pero no derivable, pues aparecen cambios repentinos en los valores de las fronteras de las distintas zonas homogéneas.

Interpolación Se puede realizar una reconstrucción con transiciones suaves (esto es, *derivables*) si se aplica una fórmula de *interpolación* o *promedio* entre las muestras cercanas: para cada punto del espacio, se calcula su valor como la suma ponderada de los valores de las muestras del entorno. Esta operación puede ser más o menos sofisticada, según el número de muestras involucradas en la operación y el factor de ponderación usado. El más sencillo de los métodos de interpolación es el lineal, donde el valor en cada punto del espacio es el promedio entre las muestras inmediatamente más cercanas. La configuración espacial de las muestras es la que determina el número de las mismas usadas para la interpolación: en un modelo 2D con muestras organizadas en una malla de triángulos serán tres, mientras que en un modelo 3D de rejilla regular rectilínea como el que se ve en la [Figura 1.4c](#) serán ocho (igual al número de vértices de los cubos que forman las muestras). El factor de ponderación será en cualquier caso la distancia normalizada a las muestras vecinas. Un ejemplo se puede ver en la [Figura 1.9d](#),

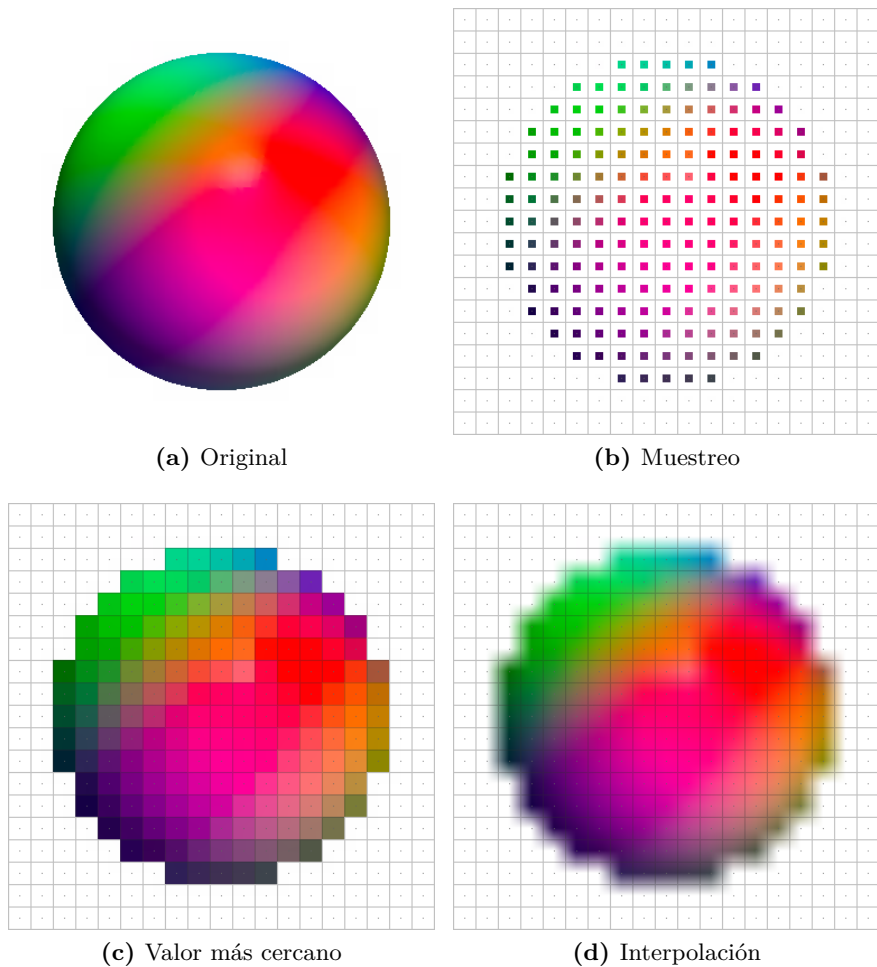


Figura 1.9 – Reconstrucción de un modelo 2D compuesto por muestras dispuestas en rejilla regular isotrópica. Arriba a la izquierda se presenta la fuente, y a su derecha se puede ver el muestreo realizado (homogéneo de 19×19 muestras). Cada celdilla representa a una muestra, y la marca central el punto del original donde se tomó el color. Abajo a la izquierda aparece la representación del modelo si se considera que todos los puntos toman el valor de la muestra más cercana (lo cual se conoce como *teselación de Voronoi*). En el modelo de abajo a la derecha se realiza, para calcular el valor de los puntos del espacio situados entre muestras, una *interpolación bilineal*, donde el valor de cada punto es la suma ponderada del valor de las cuatro muestras más cercanas.

donde se aplica este procedimiento a un modelo bidimensional con muestras en rejilla regular rectilínea isotrópica. Este tipo concreto de reconstrucción se conoce como *interpolación bilineal*.

Se puede conseguir a la vista de lo expuesto no sólo la continuidad del modelo sino también su derivabilidad. Dependiendo del tipo concreto de aplicación se podrá elegir un tipo de reconstrucción u otra, estando en cualquier caso implícitos los valores intermedios conocidos los valores de las muestras más cercanas.

1.2.2.4 Extracción de modelos de superficie

Como se acaba de exponer, el conjunto de muestras discretas se puede entender como una distribución continua y derivable de valores si se usa una técnica de interpolación. En este contexto es posible encontrar los puntos del modelo que tienen asociado exactamente el mismo valor: lo que se denomina *isocurva* en un modelo plano e *isosuperficie* en un modelo tridimensional.

Las isocurvas son un elemento frecuente fuera del mundo de los gráficos por ordenador. Las *curvas de nivel*, en los mapas, representan los puntos de los mismos que están situadas a la misma altura. Las *isobaras*, usadas en meteorología, unen los puntos de la atmósfera que en un instante concreto de tiempo están a la misma presión (ver [Figura 1.10a](#)). En general, se pueden trazar isocurvas en cualquier campo escalar bidimensional y continuo, tal y como se muestra en la [Figura 1.10b](#). Consecuentemente, y teniendo en cuenta que un conjunto discreto de muestras monovalor sobre el que se realiza una interpolación puede ser visto como un campo escalar continuo, es relativamente sencillo localizar y extraer isocurvas de cualquier conjunto plano de muestras con valores no compuestos.

El procedimiento para encontrar isocurvas en modelos de muestras se basa en el análisis secuencial de los polígonos virtuales que conforman las muestras de la rejilla, comprobando en cada uno si en la interpolación que se hace en su interior se encuentran puntos que coincidan con el valor de interés. Para ilustrar esto supóngase una rejilla regular rectilínea isotrópica en dos dimensiones. Se analizan los cuadrados que forman cada 4 muestras adyacentes, y se comprueba si se da el caso de que haya simultáneamente

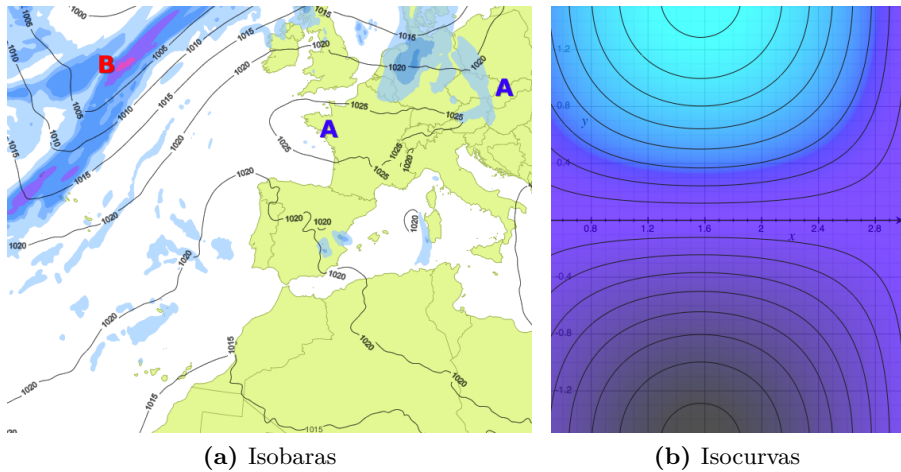


Figura 1.10 – Izquierda, isobaras. Las líneas negras sobre el mapa representan las líneas de igual presión atmosférica, obtenidas uniendo los puntos del campo escalar de presión que presentan el mismo valor (fuente: eltiempo.es). Derecha, isocurvas de un detalle del campo escalar $f(x, y) = \sin(x)\sin(y)$, donde los distintos valores se representan con distintos tonos de azul. Las líneas unen puntos para los que la función devuelve el mismo valor (y que por tanto tienen el mismo color en la imagen).

muestras por encima y por debajo del valor de interés o *umbral*. En caso afirmativo, *necesariamente* tiene que aparecer el valor umbral en el interior del cuadrado (por el teorema de Bolzano), por lo que se estudia el comportamiento de la función de interpolación bilineal para encontrar el segmento de isocurva que allí debe aparecer. En la [Figura 1.11](#) se muestra un ejemplo de este análisis.

De forma paralela a lo que sucede con los modelos planos, una *isosuperficie* muestra la curva tridimensional que une los puntos con idéntico valor de un modelo interpolado. El análisis que se realiza en los modelos tridimensionales, si bien similar al bidimensional, es ligeramente más complejo y fue un problema abierto durante algo más de dos décadas. El algoritmo original de extracción de isosuperficies, propuesto originariamente por Lorensen y Cline [[LC87](#)], se conoce como *Marching Cubes* (del inglés *cubos en proceso*) y debe su nombre precisamente al análisis progresivo al que se someten los cubos que las muestras conforman en una rejilla isotrópica. La aproximación original de la función de interpolación trilineal que calcula los valores

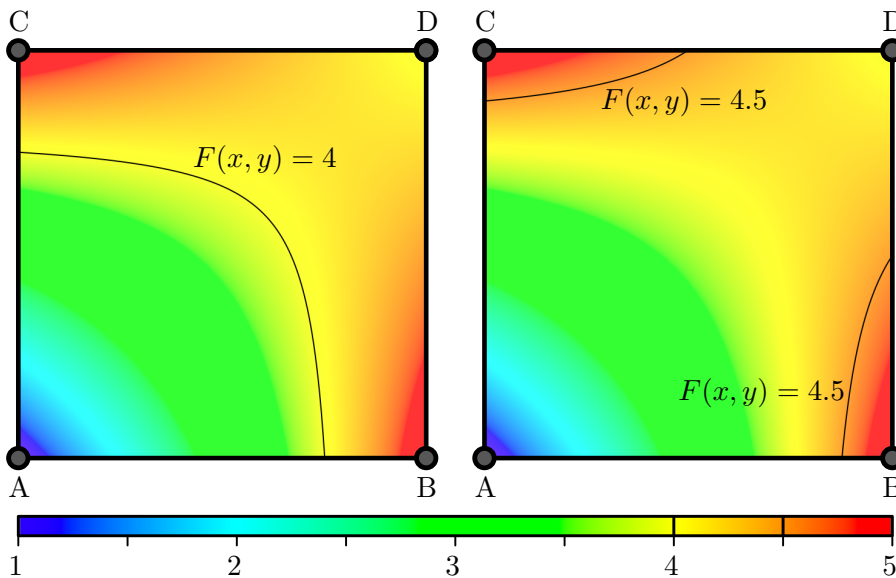


Figura 1.11 – Interpolación bilineal en el cuadrado formado por las cuatro muestras A, B, C y D, cuyos valores de propiedad son $A = 1$, $B = 5$, $C = 5$ y $D = 4$. Los colores se asocian a los diferentes valores de propiedad según la escala que se muestra abajo. A la izquierda se dibuja en negro la isocurva que pasa por los puntos con valor 4 (esto es $F(x, y) = 4$ si $F(x, y)$ es la función de interpolación). A la derecha la isocurva que pasa por los puntos con valor 4.5 (curva $F(x, y) = 4.5$). Se observa además que, dado el comportamiento hiperbólico de la función de interpolación bilineal, la isocurva puede aparecer en múltiples zonas del cuadrado.

de propiedad intermuestrales presentaba algunos problemas que se fueron resolviendo en posteriores revisiones del algoritmo [Che95, NH91, LB03], si bien en todos los casos el principio es el mismo: aproximar la sección de superficie interior a cada cubo con mallas sencillas de triángulos. En la [Figura 1.12](#) se muestran dos ejemplos de esta aproximación según la variante de Lopes y Brodlie [LB03]. Analizados todos los cubos se componen todas las secciones encontradas en un único modelo de superficie, siendo necesario realizar el procedimiento completo para cada valor umbral del cual se quiera mostrar la isosuperficie.

Gracias a los relativamente recientes avances en el hardware de procesamiento gráfico el procedimiento de extracción de isosuperficies se puede realizar sin necesidad de realizar ningún análisis previo ni construir una estructura de superficie intermedia [HSS⁺05]. El principio que se sigue es

el mismo, si bien éste se realiza de forma automática durante el proceso de visualización. Esto posible gracias a la potencia de cálculo de los procesadores gráficos específicos que casi cualquier computador integra: permiten realizar un gran número de operaciones en paralelo y reducir por tanto el tiempo de análisis de los cubos.

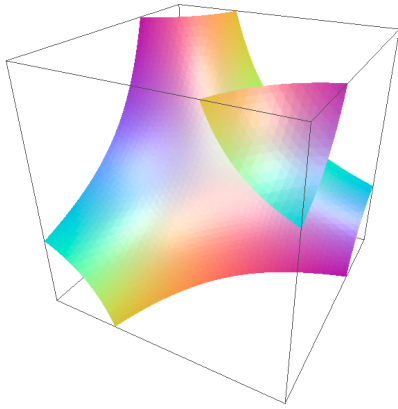
Cabe resaltar que para un único modelo de muestras interpolado existen infinitos modelos de superficie asociados: tantos como posibles valores umbrales. Pueden representarse por tanto todas las *fronteras* implícitas del objeto o medio representado, lo cual resulta de especial interés para determinadas aplicaciones. Cuando se trabaja con imágenes médicas puede ser interesante extraer por ejemplo la frontera exterior de un órgano o de una estructura ósea, obteniéndose así una representación aislada de la zona de interés. La extracción de isosuperficies es un valor añadido a las representaciones volumétricas que permite aprovechar la simplicidad de los modelos de superficie en caso de ser necesario.

1.2.2.5 Visualización de modelos volumétricos

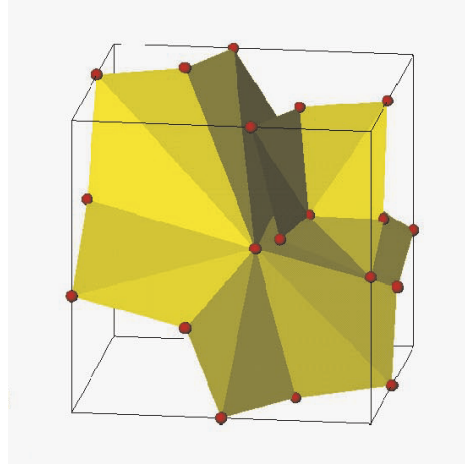
La visualización de un modelo volumétrico presenta una dicotomía inherente que es necesario sortear sea cual sea el método elegido: los modelos volumétricos representan información tridimensional continua en el espacio, mientras que los computadores necesitan componer una imagen plana para mostrarla en pantalla. Ni siquiera es posible realizar una *proyección* en un plano tal y como se hace con los modelos de superficie, pues existe información interna que en principio se antoja oculta por los datos más exteriores.

Conforme la tecnología ha avanzado se han propuesto diferentes métodos de visualización, los cuales se pueden clasificar en tres grandes grupos [BW01]:

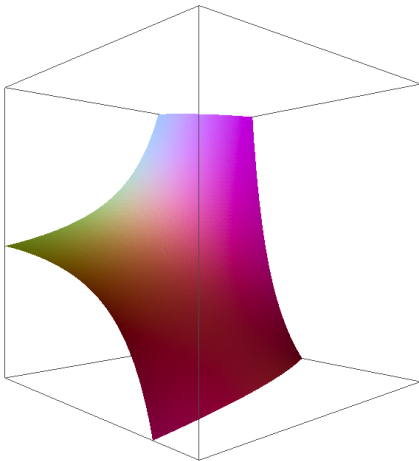
Representación de cortes Del inglés *slicing*, es quizá la técnica más antigua de visualización de volúmenes. Consiste en presentar la intersección del volumen con un plano determinado (esto es, una *rebanada*) tal y como se aprecia en el esquema mostrado en la [Figura 1.13a](#). Si bien esto puede presentar algunos problemas con los modelos de reji-



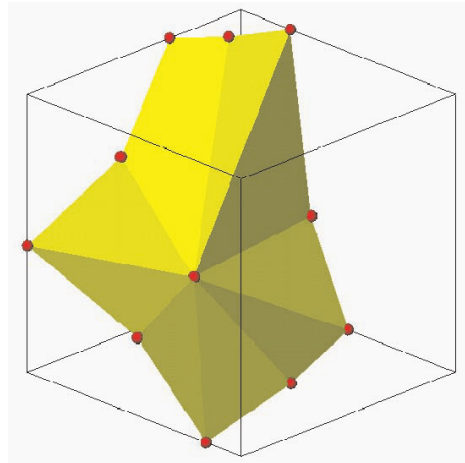
(a) Caso $P_{3/0}^E$



(b) Triangulación $P_{3/0}^E$



(c) Caso $P_{5/0}^E$



(d) Triangulación $P_{5/0}^E$

Figura 1.12 – Dos ejemplos de una isosuperficie detectada en el interior de un cubo y de su aproximación obtenida con la versión del *Marching Cubes* de Lopes y Brodlie. A la izquierda, dos casos distintos para los que la isosuperficie se ha obtenido de forma analítica: según la nomenclatura usada por los autores el caso $P_{3/0}^E$ con dos secciones enfrentadas de isosuperficie, y el $P_{5/0}^E$ con un único fragmento. A la derecha las correspondientes aproximaciones propuestas.

lla no estructurada, resulta verdaderamente eficiente cuando se trata de presentar cortes paralelos a los ejes en las rejillas rectilíneas.

Extracción de superficies Como se ha explicado anteriormente, es posible localizar y extraer un modelo de superficie de uno volumétrico buscando los puntos del último que coincidan con un valor de propiedad concreto. Otra opción, aplicable sobre todo a modelos de rejilla no estructurada, es localizar la frontera exterior del modelo y mostrar explícitamente la parte visible de la estructura subyacente (método de hecho utilizado para visualizar los modelos del elefante en la [Figura 1.5](#) y del conejo en la [Figura 1.7](#)). En cualquier caso el principio es el mismo: reducir el modelo a una superficie compuesta de polígonos planos que el hardware gráfico puede fácilmente procesar y mostrar al usuario (ver [Figura 1.13c](#)). Con esto se resuelve el problema de la visualización y además se consigue que el usuario perciba el sentido tridimensional del modelo; no obstante se pierde gran cantidad de información puesto que sólo se pueden mostrar contornos y se pierde la sensación de continuidad.

Visualización directa Del inglés *direct volume rendering*, esta estrategia consiste en componer una vista en profundidad del volumen asignando un color y un valor de *transparencia* a cada valor de propiedad. Con esto se permite ver *a través* de determinadas zonas del modelo, pudiéndose por tanto apreciar las partes de mayor interés incluso aunque sean internas. Sea por ejemplo la imagen médica que se mostró en la [Figura 1.2](#). El modelo volumétrico que ahí se muestra contiene la información tanto del hueso como del tejido blando circundante. Sin embargo la visualización se ha calibrado para que la piel y el músculo sean prácticamente transparentes, dejando por tanto a la vista los huesos del pie.

Si bien se han desarrollado numerosas técnicas que consiguen este efecto, una de las más populares se fundamenta en la extracción sucesiva de cortes en profundidad [[CCF94](#)] ([Figura 1.13b](#)). Una vez extraídas las *rebanadas*, se asignan los colores y transparencias pertinentes a los valores encontrados en dichos planos, para finalmente apilarlos en

profundidad, alineados perpendicularmente a la dirección en la que mira el observador. Cuanto mayor sea el número de cortes realizados mayor será el detalle en la vista resultante. Un número reducido de cortes puede llevar de hecho a que se aprecie el efecto de apilado, perdiéndose así parcialmente el realismo de la visión.

La capacidad computacional de procesadores gráficos actuales ha permitido no obstante aplicar un método de visualización algo más sofisticado [KW03]. Basándose en el principio de las técnicas de *ray tracing*, se lanzan una serie de *rayos* desde el punto de vista del observador que *atraviesan* el modelo (ver Figura 1.13d). Cada rayo va sumando información de color conforme atraviesa zonas translúcidas del modelo hasta que se obtiene un color completamente opaco. Con el número suficiente de rayos puede componerse una imagen todo lo detallada del modelo que se desee.

La siempre creciente agilidad con la que los computadores modernos manejan las grandes estructuras ha hecho que los métodos de presentación de cortes individuales y de extracción de isosuperficies pasen paulatinamente a un segundo plano y queden relegados a aplicaciones concretas y específicas, dando paso así a los métodos de visualización directa, los cuales claramente proporcionan una visión más completa y fidedigna de los modelos volumétricos.

1.2.3 Interacción con entornos 3D

De igual forma que es necesario *mostrar* el modelo al usuario, es también indispensable que el sistema disponga de un método de interacción para que el pueda *manipularlo*.

Al comentar los problemas de la visualización de los modelos volumétricos se ha puesto de manifiesto una cuestión que no sólo tiene que ver con la respuesta visual que proporciona el computador, sino también con la formas que dispone el usuario de manipular o *interaccionar* con un modelo tridimensional: la conexión entre los dispositivos 2D de escritorio, como la pantalla o el ratón, y los datos 3D con los que se desea trabajar.

Las tareas de interacción son muchas y dependen de la aplicación con-

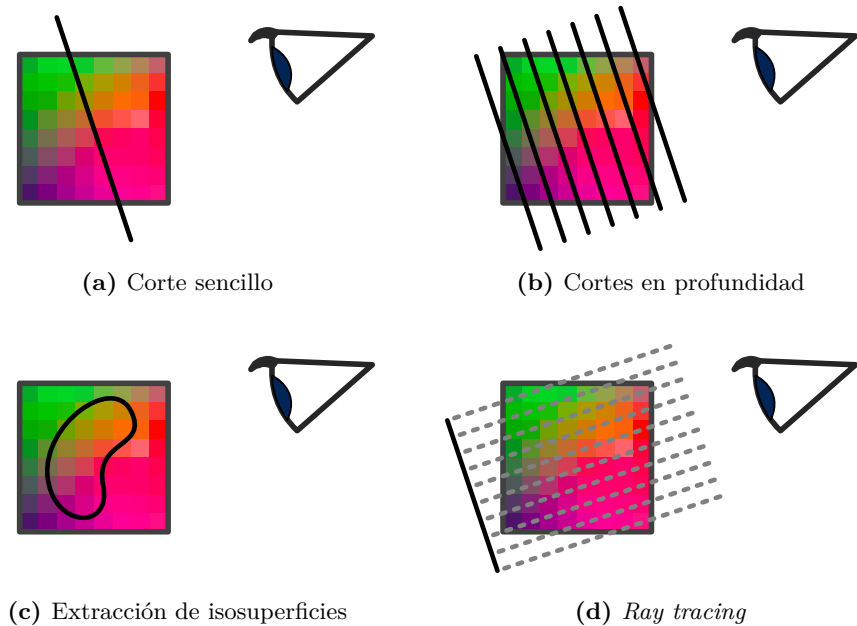


Figura 1.13 – Representación esquemática de distintos métodos de visualización de volúmenes, en la que se ilustra la escena desde un lateral, estando el punto de vista del observador representado por el ojo. En (a) se ilustra cómo se hace un único corte al volumen para mostrar una imagen plana al usuario. En (b) aparece la idea de los *cortes apilados en profundidad*, cuya composición da una visión completa del modelo. En (c) se esquematiza cómo puede también mostrarse una isosuperficie extraída del modelo. Finalmente en (d) se resume brevemente la visualización mediante *rayos* que avanzan por el modelo analizando los valores encontrados, para componer finalmente una imagen plana que contiene dicha información.

creta, si bien en general se pueden destacar tres que son comunes a casi todos los entornos: la *manipulación de objetos*, el *movimiento del punto de vista* y el *control de las opciones* de la aplicación [Han97]. Para el problema concreto que aquí se ha enunciado se pueden utilizar los métodos clásicos de interacción 2D, que permiten controlar la aplicación y el punto de vista de forma sencilla [CMS88]. Con este enfoque es el ratón, con ayuda de controles y operadores que aparecen en pantalla, el que cambia de tamaño, mueve, traslada y rota los objetos de la escena. Debe notarse que, al ser el ratón un dispositivo que trabaja sobre un plano, cualquier transformación puede controlarse hasta en *dos* dimensiones a la vez, mientras que la tercera quedará bloqueada. La conexión de los ejes norte-sur y este-oeste del ratón

con la operación concreta de traslación o rotación que se efectúa sobre el modelo vendrá determinada por los controles accesorios que se disponen en la pantalla.

Una especial atención merece no obstante la manipulación del modelo, dada la pretensión de enriquecer el sistema con una respuesta táctil. Este requerimiento pasa necesariamente por la inclusión de un *dispositivo háptico*. Si bien se profundizará más adelante en la naturaleza y funcionamiento de estos dispositivos cabe destacar en este punto que, además de aportar una vía para añadir una sensación táctil, disponen de un *puntero* que se puede mover libremente en las tres direcciones del espacio gracias a la movilidad que proporciona el brazo articulado del que están dotados. Esta capacidad facilita enormemente la representación una herramienta virtual que permita operar con el modelo, pues se tiene una asociación directa entre las posiciones del espacio virtual del modelo y las posiciones del espacio real por las que el usuario puede mover el puntero del dispositivo, no siendo por tanto incluir ninguna restricción en la interacción, tal y como sucede en el caso del ratón de escritorio.

1.3 Aportaciones

Para concluir este capítulo introductorio se describirán brevemente las aportaciones de este trabajo, indicando en cada caso las partes de este documento en la que dichos logros se justifican.

- Se ha diseñado una técnica que permite la *deformación de modelos volumétricos*. Esto se expondrá en los Capítulos 6 y 7, donde se detallarán los algoritmos y técnicas diseñadas para conseguir esta máxima.
- Para conseguir lo anterior, se han estudiado las diversas soluciones que se han propuesto anteriormente, lo que lleva al *estado de la cuestión* que se desarrolla en el [Capítulo 2](#).
- Se ha estudiado el modelo formal de deformación de sólidos que propone la *mecánica de medios continuos*. Los resultados de este estudio se resumen en el [Capítulo 4](#). Se ha formalizado una estructura (descrita en el [Capítulo 3](#)) sobre la que se adaptará esta teoría y que

culminará con la creación de las técnicas descritas en el primer punto.

- Se ha diseñado una experiencia de interacción enriquecida con *interacción háptica*, y que ha perseguido en todo caso una sensación *suave, realista, natural y coherente* con el proceso físico real. Esto se desarrollará en el [Capítulo 5](#).
- Se ha optimizado todo el proceso para que las colisiones, el cálculo de la respuesta háptica y la deformación se ejecuten a *frecuencias de actualización suficientemente altas* para sostener la *interactividad* del sistema.
- Se ha diseñado el método de deformación de forma que se permitan *cambios topológicos* en la estructura, como se verá en el [Capítulo 6](#).
- Se ha puesto el mayor empeño en maximizar la *escalabilidad* de todos los procesos, de forma que los incrementos de tamaño o resolución en los modelos afecten lo menos posible al rendimiento. Esto se justificará a lo largo de los [Capítulos 5, 6 y 7](#).

Capítulo 2

Trabajos previos

La simulación de los cuerpos sólidos deformables es un campo de estudio que ha sido históricamente muy llamativo para los investigadores del ámbito de la informática gráfica. Un modelo de deformación realista tiene aplicaciones inmediatas en una gran diversidad de áreas: realidad virtual, simulación de operaciones médicas, sistemas de escultura virtual, videojuegos... Desde 1984, cuando Sederberg y Parry presentaron el método Free-Form Deformation [SP86], se han publicado numerosos trabajos con la intención de crear métodos eficientes y realistas que generen la ilusión de elasticidad o de plasticidad.

Sin embargo, e igual que sucede en cualquier otra simulación física, el realismo y la eficiencia están claramente enfrentados [MM07]. Cuando se procura fidelidad al marco teórico que describe el proceso real es inevitable hacer un uso intensivo de los recursos de computación, sobre todo cuando aparecen sistemas de ecuaciones diferenciales en el camino hacia la solución del problema. Por otra parte, cuando se plantean soluciones heurísticas para salvar esta dificultad, se incurre inevitablemente en la *aproximación*, lo cual obviamente produce resultados más rápidos pero menos robustos.

La bibliografía existente arroja mucha luz sobre cómo abordar esta disyuntiva que se plantea entre realismo y velocidad. En algunos casos se plantean soluciones heurísticas creativas que desembocan en una solución convincente. Se aprecian diferentes niveles de fidelidad con la realidad, lo cual puede ser útil en algunas aplicaciones que no requieran una total exac-

titud pero sí se beneficien de su simplicidad (lo cual sucede, por ejemplo, en los videojuegos). En otros trabajos se encuentran brillantes adaptaciones del marco teórico a modelos simplificados que, no obstante, aportan simulaciones consistentes desde el punto de vista físico.

En este capítulo se abordará una revisión de los trabajos más notables relacionados con este área. Se repartirán, para un mejor análisis de cara al propósito de este trabajo, en tres grandes grupos. En primer lugar se hará una revisión de los métodos de exploración háptica que se han desarrollado para modelos estáticos. El problema de la navegación o exploración táctil de volúmenes está lejos de ser trivial, por lo que esta discusión puede resultar de especial interés de cara a inspirar el método de interacción que se pretende desarrollar. En segundo lugar se describirán aquellos que plantean una simulación estrictamente visual, para finalmente revisar los trabajos que plantean un método de deformación con respuesta tanto visual como táctil. En cada grupo se expondrán las soluciones por orden creciente de complejidad, empezando por los modelos heurísticos y cerrando con las simulaciones físicas más exactas.

2.1 Métodos de interacción háptica

Los esfuerzos dedicados a generar métodos de interacción eficientes y realistas han generado soluciones de exploración para prácticamente todos los tipos de modelos.

Los primeros resultados notables hicieron uso de una superficie intermedia para simular el sentido del tacto de los objetos gráficos virtuales. Es el llamado *método del campo vectorial* (del inglés *vector field method*) o de la *superficie intermedia*, introducido por Adachi et al. [AKO95] y mejorado por Mark et al. [MRF⁺96], que en ambos casos introducen en el modelo un plano virtual tangente a la superficie que se pretende explorar (ver [Figura 2.1](#)). El plano intermedio se crea en el momento en el que la herramienta háptica contacta la superficie, junto con un muelle y amortiguador virtuales conectados en paralelo a dicho plano. El plano puede oponer distintas intensidades de resistencia, sencillamente cambiando los valores de las constantes de resistencia del muelle y el amortiguador.

to check whether operators can find difference in level or not(Fig.6). In our experimental system, the limitation of D was estimated as 0.8 mm under the condition of K = 10000 (N/m) and B = 1000 (N/(m/sec)).

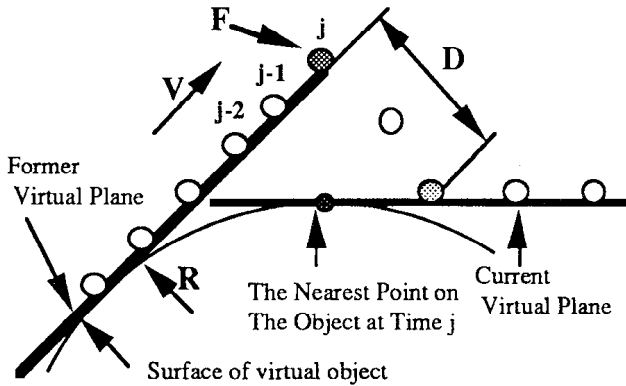


Figura 2.1 – **Figure 5. Limitation of the virtual plane** (Procesador para conseguir la navegación con plano intermedio [ZS95]). Se aprecia la limitación del sistema cuando la frecuencia de actualización del plano es baja: no concuerda la representación intermedia con la real y se producen discontinuidades.

Casi de forma simultánea, Ziles y Salisbury presentan el método de *proxy* [ZS95], método que, con las mejoras posteriores introducidas por Ruspini et al. [RKK97], resultó ser más versátil que su contemporáneo. El mecanismo, relativamente más sencillo que el propuesto por el método de campo vectorial, introducía un único punto virtual, llamado *proxy*, que está por definición contenido en la superficie del modelo. Mientras que el puntero háptico es libre de traspasar la superficie, el *proxy* marca la posición en la que estaría una herramienta real de no poder hacerlo (como se puede apreciar en la **Figura 2.2**). Para simular la fuerza de repulsión que se genera al traspasar la superficie del modelo, se genera una fuerza que atrae el puntero háptico hacia la posición del *proxy* siguiendo la Ley de Hook:

$$\vec{F} = k(\vec{p} - \vec{v})$$

donde \vec{v} es la posición real del puntero y \vec{p} es la posición del *proxy*, siendo por tanto la fuerza generada linealmente proporcional a la distancia entre las dos según la constante k . Al no ser necesariamente la fuerza perpendicular

Proceedings of the Virtual Reality Annual International Symposium (VRAS '95), 0-8186-7084-3/95 \$10.00 © 1995 IEEE

el *proxy* ofrece una gran versatilidad a la hora de simular efectos tales como rugosidad o pequeños obstáculos en la superficie. Además este método resulta ser más natural cuando aparecen superficies en ángulos muy cerrados

Distribution of GIE is... High resolution of... of each joints of... entation, moreover v... computed from the inform... spatial resolution is... space. Direct-drive m... SinMaywa Industries... reluctance cogging and... sensors and actuators... shows the result of pr... relation between force... output of SPICE was... exactly according to th... High computation... controller of SPICE. Inc.) with floating p... MHz clock speed in... virtual environment... virtual object, another... SPICE by using a vi... virtual plane are exch... However, these CPU... compute the control... three vector processor... were introduced in th... sustained rate of 206... point) at 50 MHz clo... Fig.12 shows ge... system IRIS 420/VC... virtual environment... front of a display of l... by the right hand.

Table 1: Specification

Joint	Brushless
Max Torque [N/m]	227.0
2	189.0
3	75.8
4	19.3
5(2.1)	3.71
6	1.66

goal. When the robot is unable to decrease its distance to the goal, it stops at the local minimum configuration.

Capítulo 2. Trabajos previos

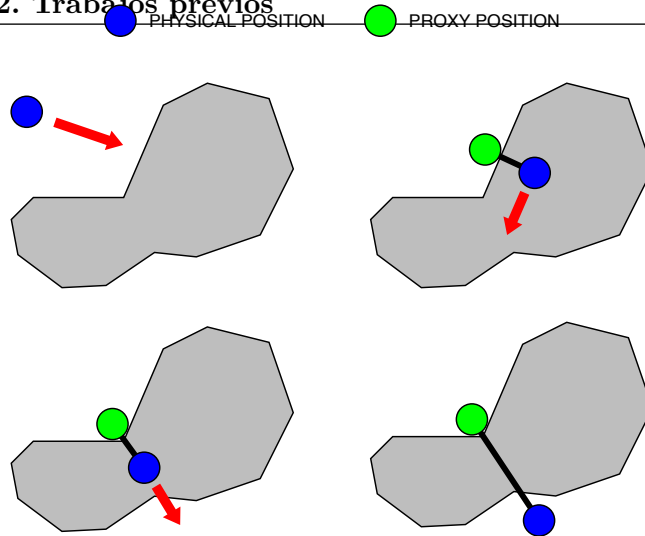


Figura 2.2 – Exploración de superficies con el método del proxy (Ruspini et al. 1997). El puntero del háptico se representa con el punto azul, y el proxy con el punto verde. Arriba a la izquierda está el caso de no interacción. Arriba a la derecha, navegación por proxy y ciclo de planificación y actualización del algoritmo. El sólido se crea el proxy, se implementa la superficie, se genera una fuerza de repulsión hacia el exterior. Abajo se representa la problemática existente en zonas estrechas: a la izquierda se examina un valle, y a la derecha la presión saca el puntero del háptico por el otro lado del sólido, si bien el proxy está atrapado en la superficie.

The remainder of this paper is organized as follows: In Section 2, we discuss previous work in haptic rendering. The basic algorithm employed to update the virtual proxy's position is presented in Section 3. In Section 4, we discuss the implementation of force shading—the haptic equivalent of Phong shading [17] in graphics. Los modelos volumétricos en regiones independientes, puros de interacción cuando la información es continua tiene diversas implicaciones no contempladas en métodos de sistemas de los sólidos por Section 6, and the low-level haptic controller is presented in Section 7. Sections 8 and 9 are devoted to the presentation of results and the discussion of future work.

2 BACKGROUND AND RELATED WORK

In penalty methods, forces are proportional to the amount of penetration into a virtual volume are applied to the haptic device. For simple geometries, like spheres and planes, the direction and magnitude (penetration) are easy to determine. The simplicity of this approach has facilitated the study of many interesting situations such as those involving dynamic objects and surface effects. This is not always the case, however. This technique by subdividing the internal volume and associating each sub-volume with a surface toward which repulsion forces are exerted [17]. This approach has also been used successfully to allow haptic interactions with volumetric data [1,12].

These approaches, however, have a number of drawbacks. When multiple primitives touch or are allowed to intersect it is often difficult to determine which exterior surface should be associated with a given internal volume. In the worst case, a

nearest exterior surface, as seen when a finger probe penetrates a surface closer to another surface of the object. This situation is illustrated in Figure 2(c), small or thin internal volume to generate the force to prevent the probe from passing through. This problem is particularly troublesome since most graphic models are composed of infinitely thin polygons, I

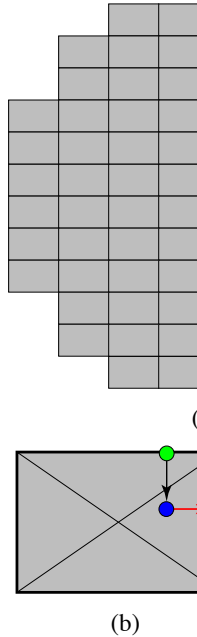


Figure 2: Limitations of “Penetration-based” methods. (a) Lack of locality: new nearest surface, (b) Force causes probe to be attracted through thin objects.

Constraint-based methods were used in applications by Zilles and others. Limitations of penalty-based methods employ a god-object which is constrained by the objects in the environment. This has been used to model interaction with a simple object and complex polygonal objects. This is an extension of this idea. This paper introduces surface constraints—force stiffness, and texture can be used to control the position of the virtual proxy. Since the proxy has a finite size, it does not always get stuck in most polygonal models without first having to reconstruct the geometry. This is required in the original god-object method.

3 UPDATING PROXY

For simplicity, we will represent the proxy as a massless sphere that moves in a 3D environment. Because of small size, the proxy are intended to share a common volume of the environment. The

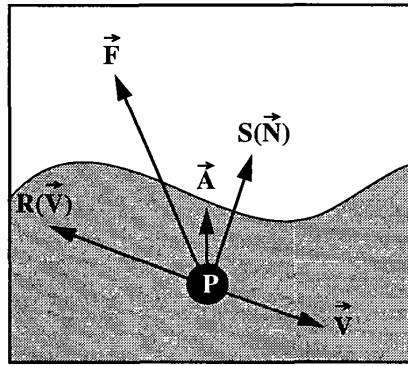


Figura 2.3 – Esquema de las componentes involucradas en el cálculo de la respuesta háptica (Avila y Sobierajski 1996). Se aprecia la oposición al movimiento $R\vec{V}$, la componente normal $S\vec{N}$ y la fuerza ambiental \vec{A} .

componente es una resistencia al movimiento, similar a la fricción que se observa en un fluido, y que simula un rozamiento dinámico. La tercera componente se opone al acercamiento a las zonas más densas, pues la normal \vec{N} apunta siempre a las zonas con menor valor de propiedad. Si bien se presentó junto al método de exploración un conjunto de operaciones que modificaba el volumen (a saber, *disminuir* y *aumentar* densidad) no se pretendió en ningún momento atacar la *forma* del objeto representado, por lo que no supuso un avance en la simulación de la deformación de volúmenes.

Tal y como más tarde apuntarían Lundin et al. [LYG02] en su estudio, el método de Avila y Sobierajski presentaba además la carencia de no tener ninguna contrapartida física: las componentes que se aplican para el cómputo de la fuerza se eligen de forma discrecional, no representando las propiedades de ningún material ni proceso físico real. En esta línea aparecen los trabajos de Chen et al. [CHS00] y de Lundin et al. [LYG02], los cuales trataron, en ambos casos, de buscar las formas geométricas implícitas en el volumen, para luego añadir interacción háptica a dichos objetos como si fueran modelos de superficie. Chen et al. propusieron, en la línea del método de campo vectorial, el uso de un plano intermedio para explorar las formas incrustadas en el volumen. El plano se adaptaría siguiendo la trayectoria del puntero háptico y la isosuperficie, calculada de forma implícita, siendo la fuerza de respuesta generada según el modelo muelle-amortiguador del método original para modelos de superficie. Lundin et al. propusieron una

adaptación directa del método del proxy a los modelos volumétricos, en la cual se constreñía el mismo a la isosuperficie calculada también de forma implícita, siguiendo sus movimientos los mismos principios que el algoritmo original. En la [Figura 2.4](#) se ilustra la similitud de este método con la contrapartida para modelos de superficie, con la salvedad de que el proxy se calcula según las propiedades del entorno en un dominio voxelizado. En

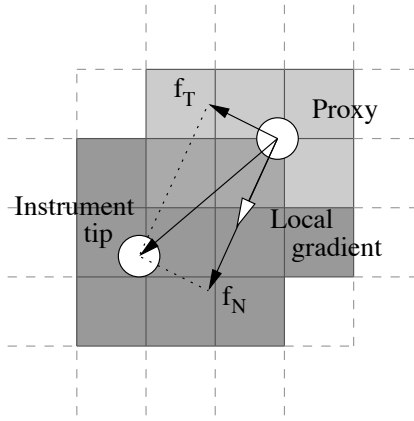


Figura 2.4 – Extra force components from the proxy displacement

Figura 2.4 – Extra force components from the proxy displacement (Lundin et al. 2002). Cuando el punto óptico traspasa el valor umbral se crea el proxy, y se permite su movimiento por la superficie implícita definida por el valor umbral.

ambos métodos el tratamiento del modelo es, una vez determinado el valor de la isosuperficie que se pretende explorar, exactamente el mismo que el de un modelo de superficie, descartándose las variaciones de propiedad en el entorno de la misma para el cómputo de la fuerza.

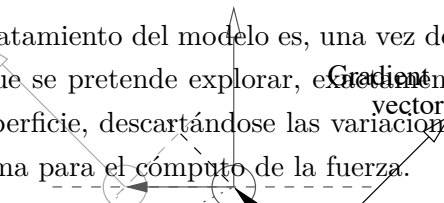


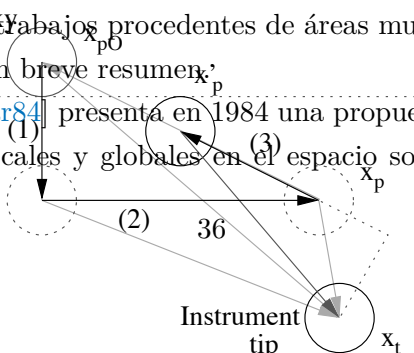
Figure 2. How the proxy follows a virtual surface

Figura 2. How the proxy follows a virtual surface (Lundin et al. 2002). Cuando el punto óptico traspasa el valor umbral se crea el proxy, y se permite su movimiento por la superficie implícita definida por el valor umbral.

2.2 Modelos deformables con respuesta visual

Algunos de los puntos de la aparición de los dispositivos hápticos, la elasticidad, plasticidad e incluso la fractura y el ruido en el objeto de estudio, pues es obvio el interés que tiene poder representar no sólo escenas estáticas sino

objetos que se mueven, colisionan y deforman. Es por ello que existe una amplia colección de trabajos procedentes de áreas muy dispares, de la cual se presentará aquí un breve resumen.



Como Barr [Bar84] presenta en 1984 una propuesta de primitivas que permiten cambios locales y globales en el espacio sobre el que se definen

friction, stiffness, viscosity by the scalar value C of the functions are used in the function are used in volume transfer functions in rendering is shown in table 1. Input and the user specified on, stiffness, etc. The mass, viscosity and surface $V(\vec{x}_s)$, $k = K_{tf}(V(\vec{x}_s))$,

los modelos, sienta las bases del que sería el primer modelo de deformación como tal: el *Free-Form Deformation* de Sederberg y Parry [SP86]. Este permitirá, definiendo un conjunto de puntos de control, la deformación manual de un objeto. El cambio de forma final viene dictado por una función de deformación cuyos parámetros son precisamente los desplazamientos de los puntos de control. Si bien el método no considera ningún principio físico

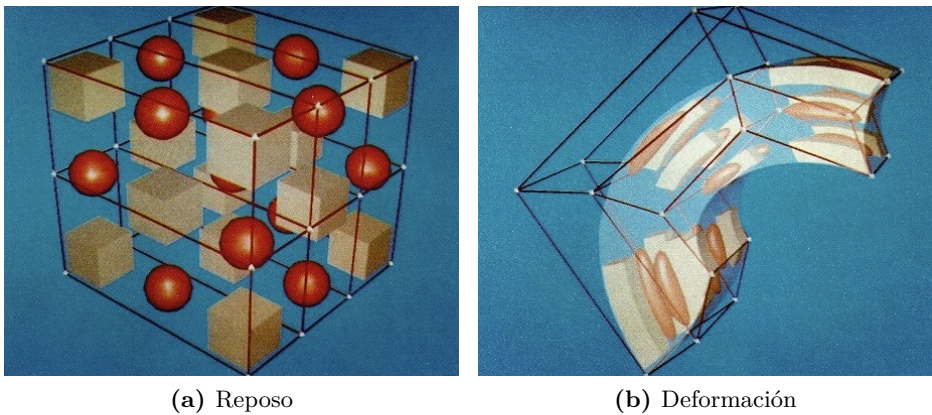


Figura 2.5 – Free-Form Deformation de Sederberg y Parry (Sederberg et al. 1986). A la izquierda, modelo en reposo. Los puntos de control se marcan en blanco. A la derecha, deformación causada por el desplazamiento de la estructura de control.

para controlar el cambio de forma de los objetos, proporciona una sensación moderadamente realista de elasticidad en los modelos que se someten a deformación. Los cambios de forma se calculan de forma interactiva ante las variaciones de los puntos de control. Por lo laxo de la definición de Sederberg y Parry es posible también usar una función de deformación distinta de la que se propuso originalmente, pudiéndose así conseguir diferentes efectos y variaciones.

Los modelos muelle-masa (del inglés *spring-mass*) han sido utilizados ampliamente en la simulación de tejidos humanos [Del98]. El principio básico de estas estructuras es sencillo: un conjunto de puntos (masas) distribuidas en el espacio, conectadas entre sí por enlaces estáticos que admiten cierto desplazamiento y/o resistencia (muelles). La posición de cada uno de los puntos viene expresada por la energía transmitida desde sus vecinos y los factores de compresión de los enlaces, normalmente lineales, siendo

Figure 2.5: Initial mesh of a human bust of sampling resolution reduced to 2.5k.



Initial mesh of a human bust of sampling resolution reduced to 2.5k.

Adaptive sampling of 2.5k.

to the task of matching the general appearance [8]. However,

fat dark	165, 140, 100	10.0
fat bright	190, 160, 110	20.0

2.5 Spring Mesh Topology

Capítulo 2: Trabajos previos

Every vertex \mathbf{x}_s^i of the facial surface mesh belongs to a vertex \mathbf{x}_b^i on the skull. This relation enables us to construct a mesh of springs between skin and skull. The attachment point on the skull \mathbf{x}_b^i is computed by tracing the surface normal \mathbf{n}_i of the skin through the tissue in the direction to the skull. \mathbf{n}_i is approximated by averaging the normals of all adjacent triangles. The skull intersection point \mathbf{x}_b^i is found by thresholding the CT data set registered to the facial surface. We distinguish between two types of springs.

Main springs connect the skin vertices \mathbf{x}_s^i with the skull vertices \mathbf{x}_b^i on the bone. They are normal to the skin surface.

Strut springs connect a skin vertex \mathbf{x}_s^i with the neighbors $\mathbf{x}_b^{neighbor(i,j)}$ of the related skull vertex \mathbf{x}_b^i .

Figure 7 illustrates this relationship. We do not use biphas springs and the masses that anchor the masses to the bone. In this case se

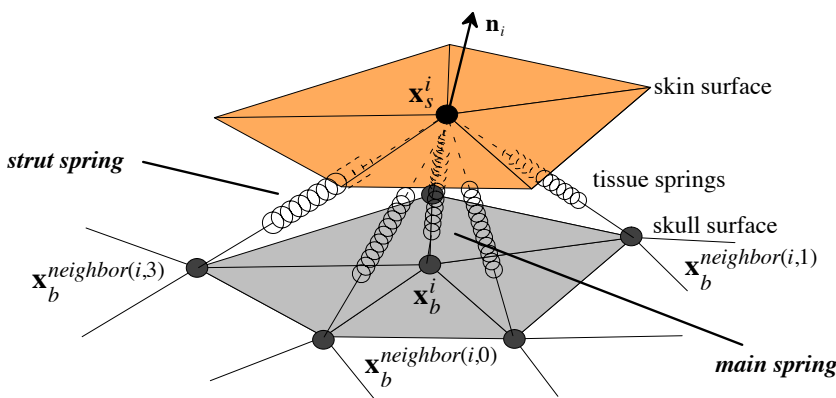


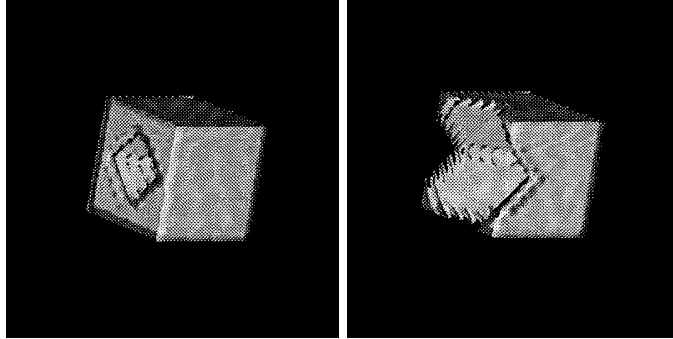
Figure 7: Spring mesh for soft tissue modeling.

Figura 2.6 – Modelo muelle-masa de Koch et al (1996). Se aprecian los enlaces a las masas vecinas (*strut springs*) y al cráneo (*main springs*).

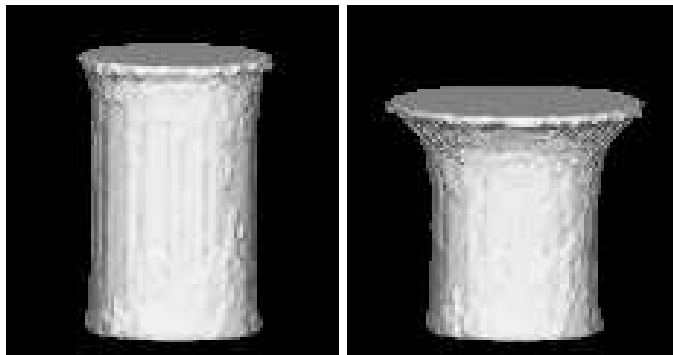
consigue estimando las constantes de cada enlace según la intensidad de los vóxeles asociados en un TAC, bajo la premisa de que la resistencia de los muelles es proporcional a la densidad del tejido.

En el mismo año, Arata et al. [ATTY99] y Sakamoto et al. [STK99] presentan un enfoque heurístico que permite simular materiales plásticos similares a la arcilla usando un modelo volumétrico en rejilla regular tridimensional. En ambos casos el concepto es similar, y considera el principio básico de que cada vóxel del modelo es un contenedor de material, y que este puede transferirse a celdas vacías vecinas en determinadas circunstancias. En el modelo de Sakamoto et al. cada vóxel posee una función de transferencia fija, de forma que, al detectarse un cambio, lo propaga a sus vecinos, que a su vez volverán a propagarlo en sucesivas iteraciones generando la

deformación global. Por su parte Arata et al. proponen unas reglas de propagación variables que se aplican en bloques de vecindad. Cuando en una vecindad se da la condición de que hay al menos un vóxel con más material del permitido, se inicia un proceso de relajación iterativo que transfiere material entre celdas hasta que todas están por debajo del máximo prefijado. Este proceso de relajación además mantiene constante la cantidad de



(a) Sakamoto et al., 1999



(b) Arata et al., 1999

Figura 2.7 – Resultados de la deformación de Sakamoto et al. (arriba) y Arata et al. (abajo). En el cubo de Sakamoto et al. la deformación se inicia en un único punto, y se propaga por el sólido abriendo un agujero cada vez mayor conforme pasan las iteraciones. Nótese que no se conserva constante el volumen del material. El cilindro de Arata et al. se comprime desde su parte superior, mostrándose dos estados distintos.

material y por tanto el volumen del modelo, lo cual no sucede en las transformaciones que se definen en el trabajo de Sakamoto et al. (ver Figuras 2.7a y 2.7b).

En la línea de estos trabajos, Dewaele y Cani [DC04a, DC04b] presenta-

ron un interesante método que permite la deformación plástica global de un modelo volumétrico de arcilla. La estructura se define igualmente como una rejilla regular tridimensional en el que cada muestra contiene un valor que representa el nivel de ocupación. Si bien el principio es el mismo, el sistema de transferencia es más sofisticado que sus predecesores. Las compresiones que se realizan en una zona del modelo tienen unas consecuencias globales, de forma que no se *propagan* los cambios, sino que se *aplican* sobre el sólido en su conjunto: cada aplicación de la herramienta genera un campo de desplazamientos de un cierto alcance, lo cual se traducirá en un *desplazamiento* del material que contiene cada uno de los vóxeles afectados por el campo. Un ejemplo se puede apreciar en la [Figura 2.8](#). Esta sutil diferencia en la

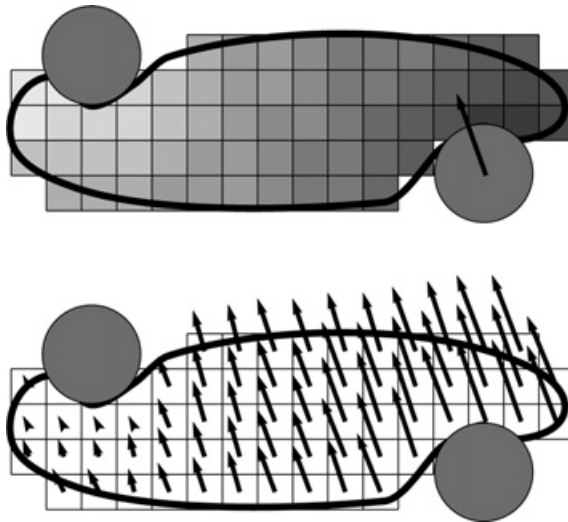


Figura 2.8 – Cálculo del alcance de una deformación (Dewaele y Cani, 2004).

aplicación de las modificaciones es la que permite que se puedan realizar deformaciones globales y no solo locales, como sucedía en sus predecesores. Cabe la posibilidad además de definir varias herramientas que actúan de forma simultánea, lo cual permite generar campos de desplazamiento que simulan efectos tales como el cambio de curvatura en varios puntos a la vez o torsiones en el material.

Hui y Leung [[HL02](#)] desarrollaron un modelo híbrido que usa una representación de superficie para calcular deformaciones en un cuerpo deformable, si bien se apoyaba en un modelo volumétrico de vóxeles para permitir

modificaciones locales de *añadir* y *quitar* material. Las deformaciones se calculan aplicando una condición de equilibrio sobre la superficie del sólido — resolviéndose por tanto la ecuación mediante el método de elementos de frontera (del inglés *Boundary Elements Method*, *BEM*). Los despla-

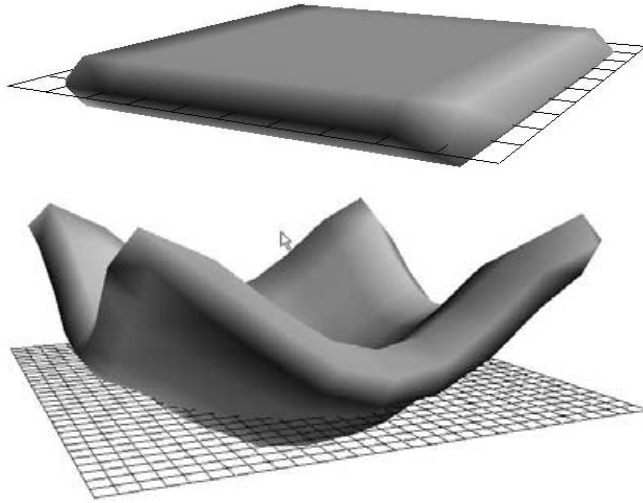


Figura 2.9 – Deformación de una plancha de material (Hui y Leung, 2002).

mientos y fuerzas se evalúan directamente sobre el modelo de superficie, actualizándose el modelo secundario de vóxeles una vez se ha producido la deformación. En la [Figura 2.9](#) se puede apreciar el efecto conseguido al aplicar tensión en los cuatro extremos de una plancha de material deformable. Por su parte, las aplicaciones de poner y quitar material son locales y no afectan en modo alguno a la forma global del sólido.

La fractura y deformación fueron metódicamente estudiadas por O'Brien et al. [OBH02]. Partiendo de un trabajo que inicialmente estaba enfocado a la simulación de la rotura de materiales frágiles, extendieron los resultados para abarcar materiales que admitieran cierta deformación elástica y plástica antes de presentar fracturas. Para ello hicieron uso de un modelo formal de la rotura basado en mecánica de medios continuos, resultando en un proceso realista y creíble para un amplio rango de materiales. Los parámetros variables de elasticidad, plasticidad y momento de rotura permiten de hecho simular desde materiales que presentan cierta componente elástica antes de admitir una deformación permanente hasta materiales pu-

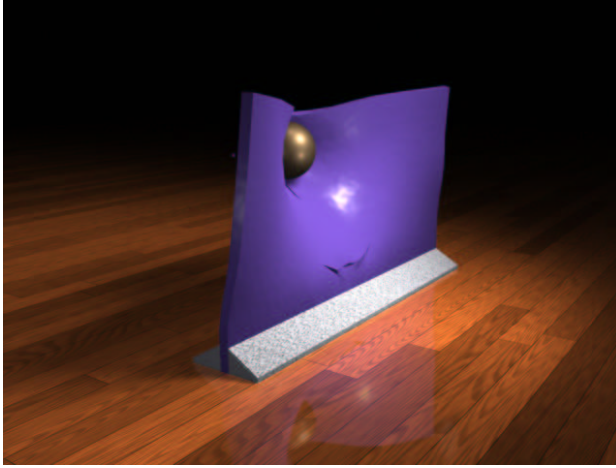


Figura 2.10 – Simulación de la rotura de una pared de material con propiedades elásticas y plásticas al lanzar un proyectil contra ella (Obrien et al. 2002).

ramente plásticos. El método que se plantea está en cualquier caso enfocado a la creación de animaciones realistas, no pretendiéndose por tanto ofrecer resultados de forma interactiva. No se plantea de hecho ningún método de interacción con el material.

Una solución exacta al problema de la deformación interactiva de materiales fue la que presentaron Metzger et al. [MTPS08]. Su propuesta se basa en la adición de un modelo de elementos finitos (FEM) a los modelos de superficie que se pretenden deformar. El modelo se considera un material plástico ideal, y la deformación se calcula de forma exacta siguiendo principios de mecánica de medios continuos y un método de integración limitado en el tiempo. Adicionalmente, y en lugar de trabajar con una estructura de tetraedros lineales como es usual, se utiliza un esqueleto de tetraedros cuadráticos o curvilíneos. Esto produce mejores resultados incluso con una reducción significativa del número de tetraedros presentes en la estructura. Las diversas optimizaciones diseñadas, tales como el cálculo retardado o *lazy* de algunos parámetros o el método de integración usado garantizan un tiempo limitado para cada etapa de cálculo. Esto permite que se puedan realizar simulaciones con una cota de actualización inferior a 25Hz en modelos compuestos de más de 1500 tetraedros lineales o de más de 300 tetraedros cuadráticos. En la [Figura 2.11](#) se puede apreciar los resultados

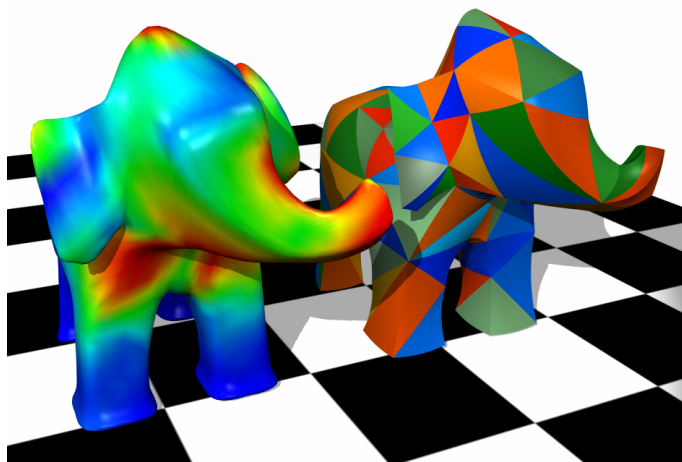


Figura 2.11 – Simulación de la deformación del modelo de un elefante, compuesto por 234 tetraedros, realizada de forma interactiva con una velocidad de actualización de 40Hz (Mezger et al. 2008). La tensión plástica acumulada se marca con el gradiente de color en la izquierda. A la derecha se muestra el modelo de tetraedros deformado.

obtenidos con el modelo del elefante visto en el [Capítulo 1 \(Figura 1.5\)](#), en su versión de tetraedros cuadráticos.

2.3 Modelos deformables con respuesta visual y háptica

Si bien algunas de las soluciones propuestas para simular el tacto de los sólidos deformables se basan en soluciones existentes vistas en la sección anterior, también se han desarrollado en algunos casos modelos originales de deformación, diseñados explícitamente para producir no sólo una respuesta visual creíble sino también una sensación táctil suave y coherente con el mundo real.

Los modelos muelle-masa son de aplicación directa cuando se quiere producir una fuerza de respuesta del sistema, puesto que su estructura está inspirada precisamente en un modelo físico de tensiones internas. Si bien son recreaciones arbitrarias que normalmente requieren estudios específicos para dotarlos de sentido físico [NC10], se han usado de hecho de forma recurrente para simular el tacto del tejido humano [Del98] y otros

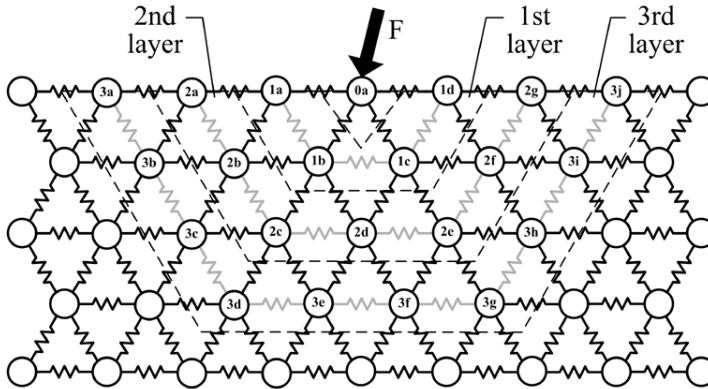


Figura 2.12 – Simplificación bidimensional de la estructura de muelle-masa con propagación de la tensión en capas de Choi et al. (2003).

materiales elásticos. A pesar de ser simplificaciones arbitrarias de la realidad, se han conseguido resultados interesantes para algunas aplicaciones concretas. Choi *et al.* [CSH03] propusieron un modelo de masas organizadas en tetraedros, con una propagación de la deformación en capas, de forma que la tensión y por tanto la deformación se atenúan con la distancia. Para ajustar las constantes de compresión de los muelles de la estructura se proponen tests empíricos sobre el material real.

De forma similar McDonnell et al. [MQW01] usan una red de masas y muelles incrustadas en una abstracción conocida como *sólidos de subdivi-*

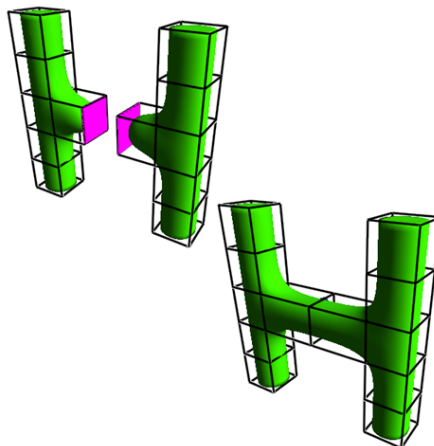


Figura 2.13 – Estructura de celdas divisibles de McDonnell et al. (2001).

sión. Las tensiones se calculan con el mecanismo habitual de los sistemas muelle-masa, si bien la variación permite *subdividir* las celdas originales en otras más pequeñas, modificándose la estructura de forma dinámica hasta el punto de permitir borrados, y consecuentemente, cambios topológicos.

El *Long Elements Method* (método de elementos alargados), desarrollado originariamente por Costa y Balaniuk [CB01] y posteriormente mejorado por Balaniuk y Salisbury [BS02], propone una estructura similar a los métodos muelle-masa, con la diferencia de que los muelles son elementos longitudinales que conectan caras opuestas del modelo, organizados en planos ortogonales. Los muelles se comprimen al contacto con el puntero

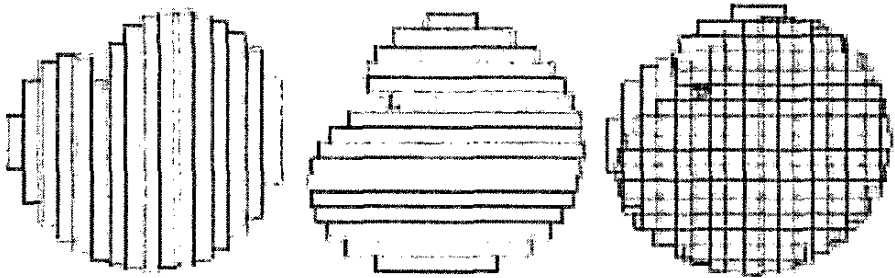


Figura 2.14 – Sección de un cilindro elástico modelado con el *Long Elements Method*. (Costa y Balaniuk, 2001).

háptico, creando la ilusión de elasticidad. Aunque el método original es particularmente sencillo, la revisión que realizan Balaniuk y Salisbury propone métodos para la estimación de las constantes de elasticidad, añadiéndose de esta forma cierta fidelidad al comportamiento real de los materiales que se pretenden simular.

Frissen-Gibson propuso un algoritmo heurístico algo más sofisticado para la simulación rápida y eficiente de materiales elásticos [FG99]. El *Chain-Mail Algorithm* (algoritmo de la cota de malla) usa una red regular de masas organizadas en planos ortogonales, en las que cada masa está conectada con sus ocho vecinos inmediatos por ocho enlaces elásticos. Cada masa admite un cierto desplazamiento máximo, superado el cual, arrastra en cadena a sus vecinos más cercanos. El resultado es un *volumen enlazado* que acepta cierto grado de deformación. En la [Figura 2.15](#) se puede observar el resultado de aplicar un desplazamiento a un nodo. Como se aprecia, el

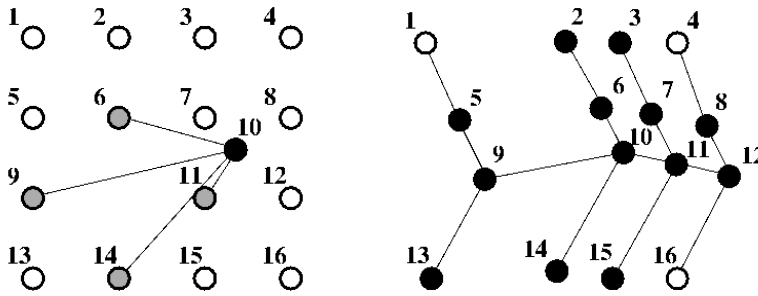


Figura 2.15 – Propagación de los desplazamientos según el *Chain-Mail Algorithm*. (Friskin-Gibson, 1999).

movimiento se transmite coherentemente a la vecindad, generando la sensación de deformación local. Por su parte, el método de cálculo de la fuerza sigue el mismo principio que en los métodos muelle-masa generales.

Existen también trabajos que han usado modelos de elementos finitos (FEMs) para simular el tacto de sólidos deformables. Digno de mención es el trabajo presentado por Cotin y Delingette [CD98] y posteriormente extendido por Delingette y Ayache [DA04], en el cual se usaron principios de mecánica de medios continuos para generar deformaciones realistas y robustas en tejidos orgánicos. El modelo de elementos finitos, en forma de malla de tetraedros, se obtiene de imágenes obtenidas mediante un escáner TAC

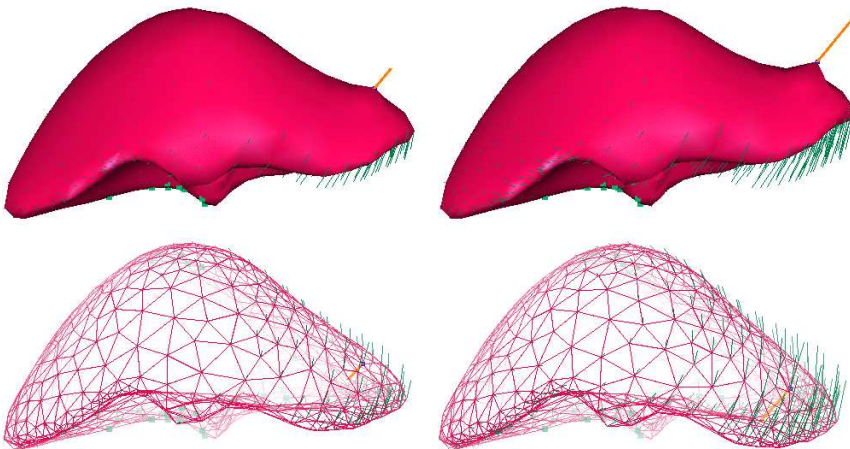


Figura 2.16 – Deformaciones de un modelo del hígado en el trabajo de Delingette y Ayache (Delingette et al., 2004). Arriba, levantamiento. Abajo, aplicación de presión hacia abajo.

en una etapa de preprocesamiento, lo cual garantiza la afinidad del modelo al tejido real. La respuesta háptica se garantiza con un set predefinido de operadores de interacción calculados en la etapa de preprocesamiento, puesto que la complejidad del modelo no hubiera permitido el cálculo de la fuerza en tiempo de ejecución. En la [Figura 2.16](#) se presentan dos muestras de la interacción con un modelo del hígado. Más adelante Peterlík et al. [[PM07](#)] presentaría algunas mejoras al método de Delingette, incluyendo optimizaciones en la aplicación de los principios de mecánica involucrados en la simulación.

Capítulo 3

Formalización del problema

La resolución de cualquier problema con la ayuda de un ordenador pasa necesariamente por la definición de la representación de los datos y de una interacción persona-computador adecuada al entorno. Es por ello que una vez analizados los requisitos del problema es necesario declarar formalmente estos aspectos. Esto es lo que permitirá traducirlos al lenguaje máquina en fases posteriores.

En el [Capítulo 1](#) se valoró la adecuación de los diferentes tipos de modelos al problema concreto que concierne a este trabajo, llegándose a la conclusión de que lo más idóneo, dado que se pretenden simular las tensiones internas de los materiales para conseguir un efecto realista, es usar un modelo volumétrico. Esta conclusión se ha visto reforzada tras el análisis de las soluciones existentes, donde han quedado patentes las carencias de los modelos de superficie y sus derivados. Sintetizando lo discutido en los [Capítulos 1](#) y [2](#) sobre los beneficios de los modelos volumétricos de cara a la simulación que se pretende realizar:

- Los modelos de superficie no aportan información sobre la estructura interna. Para paliar esta carencia es necesario introducir estructuras intermedias tales como redes *spring-mass*, que permitan de alguna forma controlar lo que sucede en el interior de los modelos.
- Las estructuras intermedias requieren de métodos heurísticos para aproximar el comportamiento del modelo a la casuística real, requi-

riéndose por tanto aproximaciones *ad hoc* en cada simulación.

- Si bien la simplicidad de las estructuras *spring-mass* permite realizar operaciones con gran eficiencia, no es posible controlar propiedades tales como la conservación del volumen ocupado por el modelo durante una deformación o permitir cambios topológicos sin complicar el procesamiento de forma dramática [Del98]. Mantener constante la cantidad de espacio que engloba el modelo requiere calcular regularmente el volumen de las celdas y realizar operaciones de *trasvase* cuando una se comprime, mientras que permitir un cambio topológico requeriría, por una parte, complicadas comprobaciones para detectar las colisiones e intersecciones entre las superficies involucradas, y por otra cambios en la estructura intermedia para reflejar la nueva configuración.
- Tanto la conservación del volumen ocupado como los cambios topológicos son fácilmente controlables cuando se trabaja con modelos volumétricos de rejilla regular [STK99] [ATTY99]. Las celdas actúan como contenedores unitarios, resultando relativamente sencillo controlar estos dos aspectos si se coordinan correctamente los cambios en las celdas afectadas y sus vecinas en una operación de compresión o expansión.

Con todo lo expuesto, y tras aparecer resultados prometedores en los trabajos que usaron modelos volumétricos de rejilla regular, se justifica la decisión de usar esta estructura para simular un sólido deformable, fin último de este trabajo. En lo que sigue se procederá a la definición formal del modelo que representará el material, así como de los demás elementos asociados al entorno del problema, tales como la herramienta que utilizará el usuario para interactuar, o la configuración del espacio virtual que más tarde se usará para reflejar cambios estructurales en el modelo. Estas definiciones sentarán las bases para el diseño de los procesos de visualización, deformación y de cálculo de las fuerzas implicadas en la simulación, tal y como se expondrá en los siguientes capítulos.

3.1 Material

Dado el origen discreto de los modelos volumétricos se formalizará en primer lugar la estructura primaria, compuesta de muestras aisladas que representan propiedades locales del material. Sobre ella se construirá la estructura volumétrica continua que representará el material deformable, para finalmente determinar la relación de los elementos abstractos con el modelo real.

3.1.1 Modelo de muestras

Se definirá el modelo volumétrico como un conjunto M de muestras ordenadas en una zona acotada de un espacio euclídeo tridimensional, siendo la distancia entre muestras δ_x , δ_y y δ_z constante para cada uno de los ejes. Las muestras se organizarán en una estructura de prismas de base rectangular, y se numerarán con tres índices enteros i , j y k , notándose cada muestra con m_{ijk} . El conjunto M se define por tanto:

$$M = \{ m_{ijk} \mid i \in [0, dim_x) \wedge j \in [0, dim_y) \wedge k \in [0, dim_z) \} \quad (3.1)$$

donde $dim_x, dim_y, dim_z \in \mathbb{N}$ y representan respectivamente el **número de muestras** en las dimensiones X , Y y Z (*ancho*, *alto* y *fondo* del modelo).

Cada muestra contendrá un único valor numérico positivo, al que se referirá en lo que sigue como **valor de propiedad**. Queda por tanto unívocamente definida cada muestra con una tupla de cuatro valores $\langle \vec{x}, d \rangle$ que indica su posición en el espacio tridimensional $\vec{x} \in \mathbb{R}^3$ y su valor de propiedad, $d \in \mathbb{R}^+ \cup \{0\}$. Por conveniencia se definen las funciones x y d :

$$x(m_{ijk}) : \mathbb{Z}^3 \rightarrow \mathbb{R}^3 \quad (3.2)$$

$$d(m_{ijk}) : \mathbb{Z}^3 \rightarrow \mathbb{R}^+ \cup \{0\} \quad (3.3)$$

que determinan, para cada muestra m_{ijk} , su vector de posición y su valor de propiedad. Teniendo en cuenta la estructura regular del conjunto M , fácilmente se comprueba que:

$$x(m_{ijk}) = i\delta_x\hat{x} + j\delta_y\hat{y} + k\delta_z\hat{z} \quad (3.4)$$

donde \hat{x} , \hat{y} y \hat{z} son los vectores unitarios y ortogonales de la base canónica que define el espacio tridimensional.

3.1.2 Reconstrucción del modelo

Tal y como se discutió en el [Capítulo 1](#) es posible extender el carácter discreto del conjunto M para trabajar en un espacio continuo de valores de propiedad. El objetivo es que la función d sea aplicable a \mathbb{R}^3 y no esté limitada a un conjunto discreto de puntos. Para esto el modelo se completará contemplando las siguientes directrices:

1. Se extenderá el muestreo a todo \mathbb{Z}^3 , de forma que existirán muestras fuera del conjunto M que seguirán la misma distribución regular. Para situarlas se usará la definición de la función x en [3.4](#), que define la posición de cualquier m_{ijk} tanto perteneciente como no perteneciente a M . Se determina que todo m_{ijk} que no esté en M viene dado por su posición en el espacio y el valor mínimo de propiedad: $\langle \vec{x}, 0 \rangle$. La función d se complementa por tanto con:

$$d(m_{ijk}) = 0 \quad \forall m_{ijk} \notin M \quad (3.5)$$

Los $m_{ijk} \notin M$ serán indistinguibles en cualquier caso de las muestras pertenecientes al conjunto M que también contengan el valor mínimo de propiedad.

2. Se utilizará el proceso de reconstrucción expuesto en el [Capítulo 1](#) según el cual todos los puntos del espacio toman el valor de propiedad de la muestra más cercana. En el caso de los puntos equidistantes a dos o más muestras, tomarán el valor de la muestra con mayor indexación en i , j o k .

El resultado de realizar la teselación de Voronoi sobre el modelo extendido según la primera directriz resulta en una partición de \mathbb{R}^3 en celdas o **vóxeles** con forma de prisma de base rectangular, cuyas dimensiones serán siempre la distancia entre muestras δ_x , δ_y y δ_z . Dado que toda muestra genera un único vóxel, se puede etiquetar sin ambigüedades a los vóxeles

con los índices i , j y k de la muestra que los define. Formalmente, y según lo explicado en la segunda directriz:

$$v_{ijk} = \left\{ \begin{array}{l} \vec{x} \in \mathbb{R}^3 \mid \|\vec{x} - x(m_{ijk})\| < \|\vec{x} - x(m_{lmn})\| \quad \forall i, j, k \neq l, m, n \\ \cup \left\{ \vec{x} \in \mathbb{R}^3 \mid \|\vec{x} - x(m_{ijk})\| = \|\vec{x} - x(m_{lmn})\| \quad \forall i, j, k > l, m, n \right\} \end{array} \right\} \quad (3.6)$$

de forma que todo punto \vec{x} del espacio queda unívocamente asignado a uno de los vóxeles v_{ijk} .

Se puede aplicar de forma trivial la definición de las funciones x y d definidas en 3.2 y 3.3 al dominio de los vóxeles, para obtener, respectivamente, el **centro** y **valor de propiedad** de cada uno. Añadiendo a estas dos definiciones la de la función sobreyectiva v :

$$v(\vec{x}) : \mathbb{R}^3 \rightarrow \mathbb{Z}^3 \quad (3.7)$$

que obtiene los índices del vóxel y muestra asociados a cada punto \vec{x} según lo definido en 3.6, se puede conocer para cada punto del espacio su valor de propiedad si se componen v y d :

$$(d \circ v)(\vec{x}) : \mathbb{R}^3 \rightarrow \mathbb{R}^+ \cup \{0\} \quad (3.8)$$

quedando así una asignación de valores de propiedad continua en todo el espacio tridimensional \mathbb{R}^3 .

Con la partición del espacio que se ha definido en la ecuación 3.6 se puede formalizar una relación de vecindad para cada vóxel. Al ser cada vóxel un prisma de base rectangular tendrá un total de veintiséis vecinos que se formalizarán con la función:

$$N(v_{ijk}) : \mathbb{Z}^3 \rightarrow \{v_{lmn}\} \quad (3.9)$$

la cual obtiene para un vóxel v_{ijk} determinado el conjunto de vóxeles v_{lmn} pertenecientes a su 26-vecindad. Estos se clasificarán en tres tipos, siendo ocho los vecinos con los que compartirá una cara, doce con los que compartirá una arista y ocho más con los que compartirá un vértice. Los vecinos de cara, arista y vértice se pueden diferenciar por los índices. Siendo la

diferencia máxima entre los índices de dos vecinos $i - l$, $j - m$ y $k - n$ de 1 en valor absoluto, se observa que los vecinos de cara compartirán dos índices, los de arista compartirán sólo uno y que los vecinos diagonales que comparten un vértice no tienen ningún índice igual.

3.1.3 Interpretación del modelo

El material virtual se representa, como se ha desarrollado en lo anterior, como una composición de elementos con valor de propiedad uniforme, teniendo cada uno de estos elementos un tamaño y límites definidos.

El valor de propiedad, que oscilará entre cero y un máximo arbitrario y conocido (a saber, el más alto de los valores de M) representará el *nivel de ocupación* de material presente en cada vóxel, pudiendo entenderse también como **densidad** del vóxel al ser el tamaño de todos los vóxeles constante. Un vóxel con densidad 0 representará pues, una zona *vacía*, mientras que una celda con el valor de propiedad máximo estará completamente llena de material.

La extensión del modelo de muestras que se diseñó en la sección anterior, en la que se extiende el modelo a todo el espacio \mathbb{R}^3 , recubriéndolo de vóxeles con valor de propiedad 0, se interpreta por tanto como espacio vacío que envuelve al material contenido en las muestras de M .

No obstante, el conjunto M puede contener también muestras con valor de propiedad nulo que generen por tanto vóxeles vacíos, no dándose por tanto una correspondencia unívoca entre dicho conjunto y el material. Por este motivo se introduce una nueva definición del sólido, siendo éste el conjunto de aquellos vóxeles del espacio que tienen un valor de propiedad distinto de cero. Se notará formalmente como el conjunto V , que se define:

$$V = \{ v_{ijk} \mid d(v_{ijk}) > 0 \} \quad (3.10)$$

3.2 Herramienta

La herramienta virtual que utilizará el usuario para trabajar con el modelo será un objeto tridimensional con un volumen distinto de 0, que se moverá libremente por el espacio \mathbb{R}^3 sobre el que se define el modelo vo-

lumétrico. La herramienta se puede determinar de forma exacta en cada momento por su *posición* en el espacio, su *orientación* y su *forma*.

La posición vendrá dada por un vector $\vec{c} \in \mathbb{R}^3$, que será el que determinará la localización de un punto de referencia en el objeto que representa a la herramienta de trabajo.

La orientación se determina con un segundo vector $\hat{o} \in \mathbb{R}^3$ de módulo unitario.

La forma de la herramienta se construirá evaluando una función en el espacio:

$$T(\vec{c}, \hat{o}) : \mathbb{R}^6 \rightarrow \{v_{ijk}\} \quad (3.11)$$

que determinará, para una posición y orientación concretas, un conjunto finito y conexo de vóxeles que se conocerán como el **volumen de la herramienta**.

Por simplicidad y sin pérdida de generalidad se asumirá que se trabaja con una herramienta esférica. Dado que la esfera presenta simetría respecto de cualquier plano que pase por su centro, se comprueba fácilmente que se cumplirá:

$$T(\vec{c}, \hat{o}) = T(\vec{c}, \hat{o}') \quad \forall \hat{o} \neq \hat{o}' \quad (3.12)$$

si se sitúa en \vec{c} el centro de la esfera. Para determinar correctamente T se usará el concepto de norma vectorial o distancia euclídea:

$$\|\vec{x}, \vec{c}\| = \sqrt{(x_x - c_x)^2 + (x_y - c_y)^2 + (x_z - c_z)^2} \quad (3.13)$$

entre $\vec{x} \equiv (x_x, x_y, x_z)$ un punto del espacio \mathbb{R}^3 y $\vec{c} \equiv (c_x, c_y, c_z)$ el centro al que se calcula la distancia. De esta forma se puede definir un subespacio esférico de \mathbb{R}^3 con la expresión:

$$\|\vec{x}, \vec{c}\| < r \quad (3.14)$$

donde r es el radio del subespacio y \vec{c} su centro. Se puede definir por tanto T para el caso de una herramienta esférica como:

$$T(\vec{c}, r) = \{v_{ijk} \mid \|x(v_{ijk}), \vec{c}\| < r\} \quad (3.15)$$

siendo este el conjunto de los vóxeles cuyo centro está incluido en la herramienta esférica de radio r y con centro en \vec{c} , y que no depende de \hat{o} .

3.3 Espacio

Dado que la cuestión que se pretende resolver es un problema de *deformación*, se asumirá pues un espacio *deformable*. Para esto se añadirá a la estructura volumétrica que se ha definido anteriormente una envoltura inspirada en el concepto de *Free-Form Deformation* (trabajo de Sederberg y Parry [SP86] del cual se hizo una breve descripción en el [Capítulo 2](#)). Este añadido permitirá realizar cambios geométricos temporales en el modelo, cualidad que será especialmente útil tal y como se explicará más tarde, cuando se discuta el método de deformación elástica.

La técnica *Free-Form Deformation* establece un espacio flexible controlado por una malla de puntos de control $P_{ijk} \in \mathbb{R}^3$ que conforman una partición en paralelepípedos de una zona del espacio. Estos puntos, al ser desplazados, deforman coherentemente todos los puntos X del espacio hasta una nueva posición X_{ffd} siguiendo el producto tensorial trivariado del polinomio de Bernstein:

$$X_{ffd}(s, t, u) = \sum_{i=0}^l \sum_{j=0}^m \sum_{k=0}^n B_i^l(s) B_j^m(t) B_k^n(u) P_{ijk} \quad (3.16)$$

donde s, t y u son las coordenadas de X en la región tal que $s, t, u \in [0, 1]$ para todos los puntos interiores al paralelepípedo; $l, m, n \in \mathbb{N}$ son el número de puntos de control en cada eje y las funciones B son los distintos polinomios de Bernstein definidos sobre el intervalo $[0, 1]$.

Extendiendo este principio, se definirán todos los puntos X del espacio de forma implícita, variando conforme al desplazamiento de un único punto de control no prefijado, y siguiendo una primitiva de deformación arbitraria e :

$$e(X, P, \Delta P) : \mathbb{R}^9 \rightarrow \mathbb{R}^3 \quad (3.17)$$

que desplazará todos los puntos del espacio conforme al punto de control $P \in \mathbb{R}^3$ y su vector de desplazamiento $\Delta P \in \mathbb{R}^3$. La función se definirá de

forma que $e(X) = X$ si $\Delta P = 0$, siendo esta la condición del estado de reposo o no deformación.

Trasladando esta idea al modelo, se completará pues la estructura con un conjunto finito de puntos X_{ijk} que recubrirá en todo caso al volumen V , y que se dispondrán de forma regular en el espacio:

$$X_{ijk} = X_0 + \frac{i}{l}\hat{x} + \frac{j}{m}\hat{y} + \frac{k}{n}\hat{z} \quad (3.18)$$

donde X_0 es el origen del recubrimiento, \hat{x} , \hat{y} y \hat{z} son la base canónica vectorial del espacio euclídeo \mathbb{R}^3 y $l, m, n \in \mathbb{N}$ son el número total de puntos en cada una de las direcciones de la base.

Nótese que, a diferencia de lo que sucede en el concepto FFD original, la estructura de paralelepípedos no la conforman los puntos de control, sino los X_{ijk} que serán desplazados. El punto de control P es único y, junto con la función e , es el causante de la deformación de la estructura cuando se dé la condición $\Delta P \neq 0$.

Capítulo 4

Teoría de la elasticidad y la plasticidad

La **deformación** se define como el cambio en la forma de un objeto. Es una propiedad intrínseca de los materiales, y se encuentra en medios tan diversos como la goma, los metales, el hormigón o la arcilla.

De forma más precisa, la deformación de un material está causada por el desplazamiento relativo de las partículas que lo conforman. Estos movimientos a nivel atómico o molecular producen cambios en la forma macroscópica del material, no únicamente en la superficie sino en todo el interior del cuerpo. Los cambios de forma pueden ser bien temporales o permanentes, según presente el medio, respectivamente, propiedades **elásticas** o **plásticas**. Los materiales que reúnen una o ambas de estas propiedades se conocen como **sólidos elastoplásticos**.

La deformación de los cuerpos sólidos es estudiada formalmente por una rama de la mecánica de medios continuos que se conoce como **mecánica de sólidos deformables**. Bajo esta disciplina se agrupan distintos modelos formales de la deformación, los cuales estudian minuciosamente los efectos que se producen en los materiales ante la aparición de agentes externos que producen un cambio de forma.

En mecánica de medios continuos el *medio*, sea gas, fluido o sólido, se entiende como un conjunto infinito de partículas que se estudia macroscópicamente, sin considerar las posibles discontinuidades a nivel de partícula. La

descripción matemática del medio se puede hacer por tanto con funciones continuas.

Resulta conveniente, considerando el objetivo final que se persigue de recrear de forma creíble el comportamiento de un sólido deformable real, hacer una revisión de los modelos formales que la mecánica propone para la deformación elastoplástica. Para ello se seguirá la exposición de los mismos que hacen Oliver y Agelet de Saracibar [OdsB06], resaltando y explicando en cada caso los conceptos que serán de aplicación directa en este trabajo. Este estudio teórico será un pilar fundamental tanto en el diseño de una respuesta en fuerza realista como en la recreación visual de los cambios de forma que el modelo admitirá durante la simulación. Una vez se presenten los principios matemáticos que después se usarán, se detallará en primer lugar el comportamiento de los sólidos puramente elásticos, según lo descrito en la **teoría de la elasticidad lineal**. Después se profundizará en la cuestión añadiendo la posibilidad de registrar deformaciones permanentes, tal y como se recoge en la **teoría incremental de la plasticidad**. Estos dos modelos, si bien no los más complejos de entre los que ofrece la mecánica de sólidos deformables, describen con suficiente precisión los procesos físicos reales, y serán una potente y útil herramienta en el proceso de diseño de la simulación propuesta.

4.1 Deformación

La deformación, de forma general, se estudia en términos de la configuración inicial o de referencia del sólido Ω_0 correspondiente al instante t_0 y la configuración deformada Ω_t del instante t . Se formaliza siguiendo el movimiento de una partícula determinada del medio continuo, considerando su posición relativa respecto de de las partículas de su entorno diferencial.

La **teoría de la deformación infinitesimal** [OdsB06] asume como hipótesis que los desplazamientos de las partículas son muy pequeños frente a las dimensiones típicas del medio continuo. Las deformaciones son infinitesimales y las configuraciones Ω_0 y Ω_t son por tanto prácticamente indistinguibles.

La relación entre las posiciones iniciales de las partículas y su evolución

en la configuración deformada se puede describir en términos del gradiente de los desplazamientos de las partículas del sólido. Sea \mathbf{x} un tensor de primer orden con las coordenadas de una partícula en la configuración Ω_t y $\mathbf{u}(\mathbf{x}, t)$ el campo tensorial de primer orden que determina los desplazamientos que han sufrido las partículas \mathbf{x} en un determinado instante de tiempo t . Bajo la hipótesis de la deformación infinitesimal, se define el tensor gradiente de los desplazamientos \mathbf{J} como:

$$\begin{aligned} \mathbf{J}(\mathbf{x}, t) &= \mathbf{u}(\mathbf{x}, t) \otimes \nabla \\ J_{ij} &= \frac{\partial u_i}{\partial x_j} \end{aligned} \tag{4.1}$$

donde \otimes es el producto tensorial abierto, ∇ el operador gradiente (notado aquí como un tensor de primer orden), u_i las componentes del vector de desplazamientos, y $(\hat{x}_1, \hat{x}_2, \hat{x}_3) \equiv (\hat{x}, \hat{y}, \hat{z})$ la base ortonormal del espacio, siendo por tanto \mathbf{J} un campo tensorial de segundo orden y J_{ij} sus nueve componentes ($i, j \in \{1, 2, 3\}$). El **tensor de deformación infinitesimal** $\boldsymbol{\varepsilon}$ se define pues como:

$$\begin{aligned} \boldsymbol{\varepsilon}(\mathbf{x}, t) &= \frac{1}{2} \left(\mathbf{J}(\mathbf{x}, t) + \mathbf{J}^T(\mathbf{x}, t) \right) \\ \varepsilon_{ij} &= \frac{1}{2} \left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right] \end{aligned} \tag{4.2}$$

siendo $\boldsymbol{\varepsilon}$ un tensor simétrico de segundo orden que está bien definido si se conoce el campo de desplazamientos $\mathbf{u}(\mathbf{x}, t)$.

Las componentes del tensor $\boldsymbol{\varepsilon}$ tienen una interpretación física directa, estando relacionados con los movimientos relativos del entorno diferencial de cada partícula \mathbf{x} . Supongamos los puntos P, Q, R y S del sólido, distribuidos de forma que los segmentos \overline{PQ} , \overline{PR} y \overline{PS} coinciden respectivamente con las direcciones de la base ortonormal $(\hat{x}, \hat{y}, \hat{z})$. La relación entre los puntos originales y sus homólogos P', Q', R' y S' en la configuración deformada Ω_t se recoge en las nueve componentes del tensor simétrico $\boldsymbol{\varepsilon}$:

$$\begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{bmatrix} \tag{4.3}$$

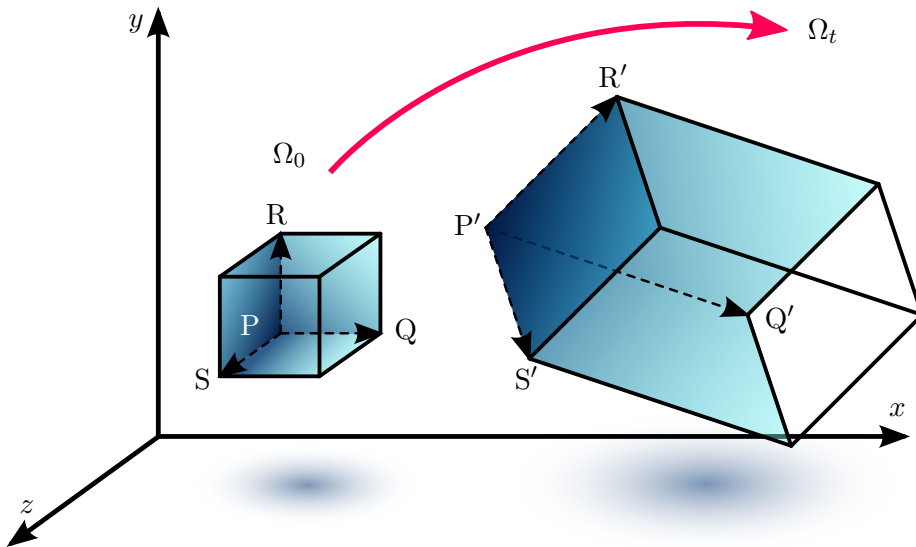


Figura 4.1 – Transformación de los puntos P, Q, R y S en sus homólogos en Ω_t P', Q', R' y S'. La distancia original entre los puntos ha cambiado, y el ángulo que formaban los segmentos, orientados en las direcciones \hat{x} , \hat{y} y \hat{z} en Ω_0 , es distinto de $\pi/2$ en la configuración Ω_t .

Las componentes ε_{xx} , ε_{yy} , y ε_{zz} de la diagonal principal de ε se denominan *deformaciones longitudinales*, y representan respectivamente los *alargamientos unitarios* de los segmentos \overline{PQ} , \overline{PR} y \overline{PS} orientados en las direcciones de la base ortonormal, definiéndose el alargamiento unitario como es el incremento de longitud del segmento diferencial deformado $\overline{P'Q'}$ por unidad de longitud del segmento diferencial original \overline{PQ} . Debe notarse que:

- Si $\varepsilon_{xx} = 0$ no existe alargamiento en la dirección de \hat{x} y por tanto el segmento \overline{PQ} no cambia.
- Si $\varepsilon_{yy} = 0$ no existe alargamiento en la dirección de \hat{y} y por tanto el segmento \overline{PR} no cambia.
- Si $\varepsilon_{zz} = 0$ no existe alargamiento en la dirección de \hat{z} y por tanto el segmento \overline{PS} no cambia.

El resto de componentes ε_{xy} , ε_{xz} y ε_{yz} y sus simétricas se denominan *deformaciones transversales*, y contienen la variación de los ángulos de los segmentos orientados en las direcciones \hat{x} , \hat{y} y \hat{z} . Paralelamente:

- Si $\varepsilon_{xy} = 0$ entonces la deformación no produce variación del ángulo entre los segmentos \overline{PQ} y \overline{PR} , orientados inicialmente en \hat{x} e \hat{y} .
- Si $\varepsilon_{xz} = 0$ entonces la deformación no produce variación del ángulo entre los segmentos \overline{PQ} y \overline{PS} , orientados inicialmente en \hat{x} y \hat{z} .
- Si $\varepsilon_{yz} = 0$ entonces la deformación no produce variación del ángulo entre los segmentos \overline{PR} y \overline{PS} , orientados inicialmente en \hat{y} y \hat{z} .

El tensor deformación $\varepsilon(\mathbf{x}, t)$ contiene, según lo anterior, tanto la información de los alargamientos unitarios en las direcciones perpendiculares a los ejes como el cambio de dirección de los mismos, dando una descripción completa de los cambios de configuración del entorno diferencial de cada partícula \mathbf{x} .

4.2 Tensión

El segundo concepto fundamental necesario para determinar la deformación de un medio continuo es el de **tensión** [OdSB06]. La tensión es la fuerza por unidad de superficie que se aplica sobre un cuerpo, y se mide como resultado de la composición de las fuerzas que actúan sobre el mismo.

Se considera que las fuerzas que pueden actuar sobre un medio continuo son de dos tipos: *fuerzas másicas*, que actúan en todo el volumen del medio (tales como la fuerzas gravitatorias o las de atracción magnética) y las *fuerzas de superficie*, las cuales actúan sobre el contorno del material, y que son producidas por el contacto de las partículas de la superficie del medio con otros agentes externos.

Las fuerzas másicas se describen mediante el campo vectorial $\mathbf{b}(\mathbf{x}, t)$ de *fuerzas másicas por unidad de masa* (notado aquí como un campo tensorial de orden uno). La fuerza resultante total \vec{F}_m de fuerzas en el instante t sobre el volumen V_t del medio se obtiene por tanto mediante la expresión:

$$\vec{F}_m = \int_{V_t} \rho \mathbf{b}(\mathbf{x}, t) dV \quad (4.4)$$

donde dV es el elemento diferencial de volumen, ρ es la densidad del medio, y $\rho \mathbf{b}(\mathbf{x}, t)$ es por tanto el vector de fuerzas másicas por unidad de volumen.

Capítulo 4. Teoría de la elasticidad y la plasticidad

Las fuerzas superficiales se expresan en términos de $\mathbf{t}(\mathbf{x}, t)$, el campo vectorial de *fuerzas superficiales por unidad de superficie*. La resultante total \vec{F}_s de las fuerzas que actúan en el instante t sobre la superficie ∂V_t del volumen V_t será:

$$\vec{F}_s = \int_{\partial V_t} \mathbf{t}(\mathbf{x}, t) dS \quad (4.5)$$

donde dS es el elemento diferencial de superficie.

Según la segunda ley de Newton, la resultante de todas las fuerzas \vec{f}_i que actúan sobre cada partícula i de un cuerpo es igual a la suma de todas las masas m_i por su aceleración a_i . Por tanto, la suma de las fuerzas másicas y superficiales de un medio continuo puede escribirse como:

$$\int_{V_t} \rho \mathbf{b}(\mathbf{x}, t) dV + \int_{\partial V_t} \mathbf{t}(\mathbf{x}, t) dS = \int_{V_t} \mathbf{a}(\mathbf{x}, t) \rho dV \quad (4.6)$$

donde $\mathbf{a}(\mathbf{x}, t)$ es el campo vectorial continuo de aceleraciones, ρ es la densidad del medio y por tanto ρdV es la unidad diferencial de masa.

Los esfuerzos internos pueden representarse paralelamente como un campo vectorial de tracción, si se considera, para un determinado punto P del interior del sólido, un plano de normal unitaria \mathbf{n} que corta al sólido en dos partes (ver [Figura 4.2](#)). Según los *postulados de Cauchy*, el vector tracción del punto P depende únicamente de la normal del plano elegido, por lo que el campo vectorial:

$$\mathbf{t}(\mathbf{P}, \mathbf{n}) \quad (4.7)$$

describe las tensiones internas (primer postulado), y además se cumple la igualdad:

$$\mathbf{t}(\mathbf{P}, -\mathbf{n}) = -\mathbf{t}(\mathbf{P}, \mathbf{n}) \quad (4.8)$$

lo cual se conoce como el principio de acción y reacción (segundo postulado), y determina que, para un punto P y dos planos de normales opuestas \mathbf{n} y $-\mathbf{n}$, el vector de tracción es también opuesto.

No obstante, la tensión en el interior del medio continuo se describe mediante el **tensor de tensiones**, el cual, como se mostrará a continuación, es independiente del plano de corte elegido, necesario para definir el campo vectorial del primer postulado de Cauchy. Supongamos un tetraedro

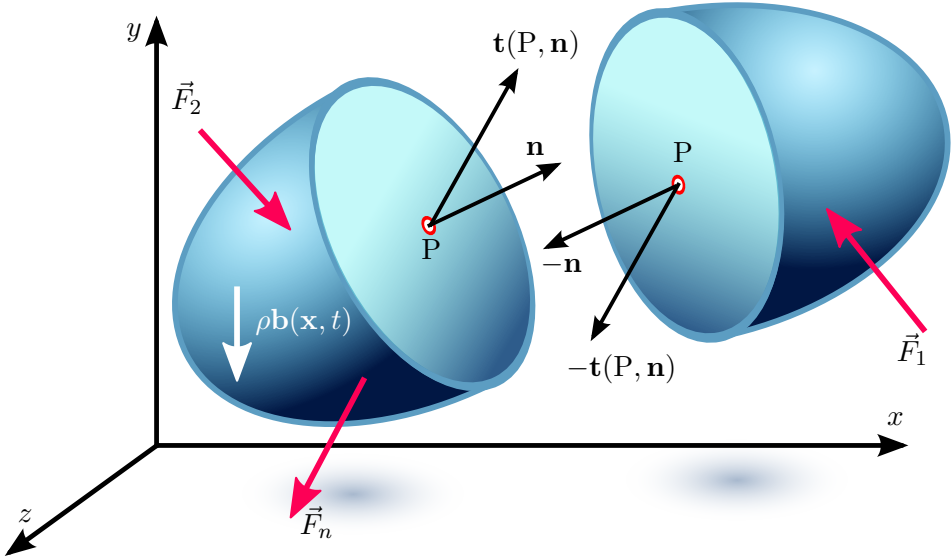


Figura 4.2 – Tensiones internas según los postulados de Cauchy de un medio continuo sometido a las fuerzas externas $\{\vec{F}_1, \dots, \vec{F}_n\}$ y al campo de fuerzas por unidad de masa $\rho\mathbf{b}(\mathbf{x}, t)$. En el punto interno P actúa un vector de tracción $\mathbf{t}(P, \mathbf{n})$, que depende del plano de normal \mathbf{n} que corta al sólido en dos partes. Además para el plano de normal opuesta $-\mathbf{n}$, el vector de tracción es $\mathbf{t}(P, -\mathbf{n}) = -\mathbf{t}(P, \mathbf{n})$.

elemental situado en el entorno de una partícula P del medio, que se sitúa por conveniencia y sin pérdida de generalidad en el origen de coordenadas. El tetraedro tiene un vértice en P y sus caras se definen mediante un plano de normal $\mathbf{n} \equiv (n_x, n_y, n_z)$ que corta a los planos ordenados definiendo una superficie S . Quedan igualmente definidas las restantes tres caras del tetraedro, con superficies S_x, S_y y S_z , contenidas respectivamente en los planos ordenados y con normales $-\hat{x}, -\hat{y}$ y $-\hat{z}$. Si se aplica la ecuación 4.6 al tetraedro definido, queda:

$$\int_V \rho\mathbf{b}(\mathbf{x}, t)dV + \int_S \mathbf{t}(\mathbf{x}, \mathbf{n})dS + \int_{S_x} \mathbf{t}(\mathbf{x}, -\hat{x})dS + \int_{S_y} \mathbf{t}(\mathbf{x}, -\hat{y})dS + \int_{S_z} \mathbf{t}(\mathbf{x}, -\hat{z})dS = \int_V \mathbf{a}(\mathbf{x}, t)\rho dV \quad (4.9)$$

donde se suman tanto las fuerzas másicas como los esfuerzos internos sobre las cuatro caras del tetraedro (definidos según el primer postulado de Cau-

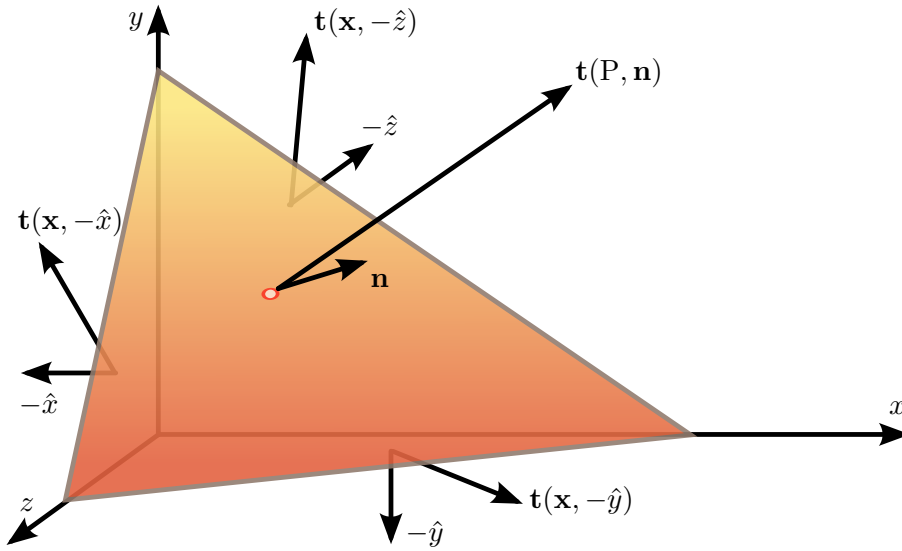


Figura 4.3 – Representación del tetraedro definido por el punto P y su normal \mathbf{n} . Se ilustra el vector de tracción $\mathbf{t}(P, \mathbf{n})$ sobre la cara S de normal \mathbf{n} , y los vectores perpendiculares a las caras perpendiculares a \hat{x} , \hat{y} , \hat{z} .

chy de 4.7) para obtener la fuerza total resultante. Si se reduce el tetraedro hasta dimensiones infinitesimales su altura tiende a cero, anulándose los términos que dependen del volumen. Se cumple por tanto que:

$$\mathbf{t}(P, \mathbf{n}) = \mathbf{t}(X, \hat{x})n_x + \mathbf{t}(Y, \hat{y})n_y + \mathbf{t}(Z, \hat{z})n_z \quad (4.10)$$

donde el vector de tracciones internas \mathbf{t} se expresa en términos de los vectores en las caras opuestas del tetraedro, siendo X, Y y Z puntos arbitrarios de las caras S_x , S_y y S_z , respectivamente. Si se descomponen los vectores de tracción de éstas en sus componentes cartesianas:

$$\mathbf{t}(X, \hat{x}) = \sigma_{xx}\hat{x} + \sigma_{xy}\hat{y} + \sigma_{xz}\hat{z} \quad (4.11)$$

$$\mathbf{t}(Y, \hat{y}) = \sigma_{yx}\hat{x} + \sigma_{yy}\hat{y} + \sigma_{yz}\hat{z} \quad (4.12)$$

$$\mathbf{t}(Z, \hat{z}) = \sigma_{zx}\hat{x} + \sigma_{zy}\hat{y} + \sigma_{zz}\hat{z} \quad (4.13)$$

donde necesariamente se cumple que $\sigma_{xy} = \sigma_{yx}$, $\sigma_{zx} = \sigma_{xz}$ y $\sigma_{zy} = \sigma_{yz}$ por el balance del momento angular. No obstante, y dado que los planos S_x , S_y y S_z se definen en cualquier caso en función de P, los vectores $\mathbf{t}(X, \hat{x})$,

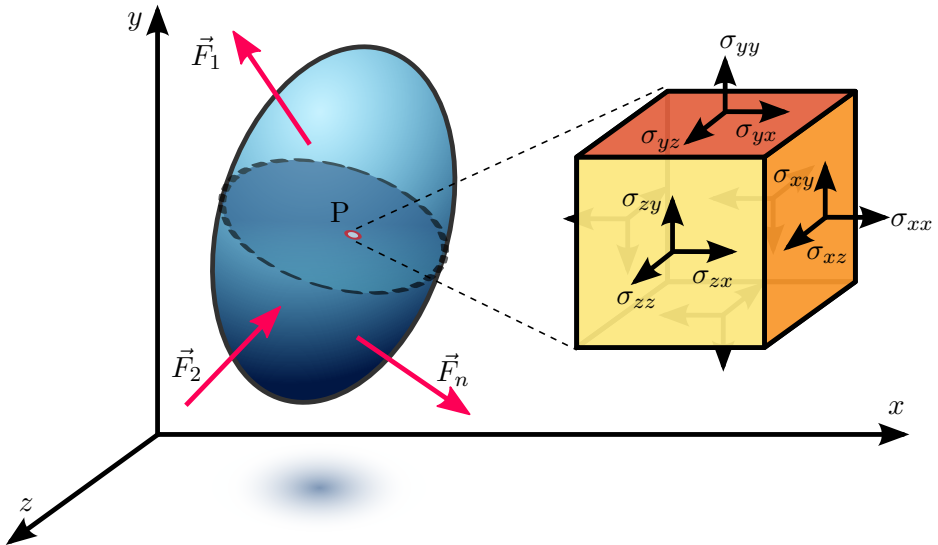


Figura 4.4 – Representación del tensor de tensiones de Cauchy en un paralelepípedo. La tensión del punto P se descompone en las tensiones en los tres planos perpendiculares a los vectores de la base $\hat{x}, \hat{y}, \hat{z}$. Cada σ_{ij} representa la tensión sobre el plano perpendicular a i en la dirección del eje j .

$\mathbf{t}(Y, \hat{y})$ y $\mathbf{t}(Z, \hat{z})$ dependen únicamente del punto P elegido. Las componentes cartesianas de estos tres vectores se reúnen en el **tensor de tensiones de Cauchy** $\sigma(\mathbf{P})$:

$$\begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \quad (4.14)$$

siendo éste un tensor de segundo orden simétrico que permite reformular la ecuación 4.10 como:

$$\mathbf{t}(\mathbf{P}, \mathbf{n}) = \sigma(\mathbf{P})\mathbf{n} \quad (4.15)$$

y por tanto tener una expresión de vector de tracciones del punto P que es independiente del plano de corte elegido.

Es usual representar gráficamente el tensor de tensiones como un paralelepípedo elemental alrededor del punto P sobre el que se define, lo cual proporciona una vista intuitiva de cómo actúan las componentes del tensor sobre los planos perpendiculares a \hat{x} , \hat{y} y \hat{z} , tal y como se aprecia en la [Figura 4.4](#).

4.3 Teoría de la elasticidad lineal

Bajo ciertas hipótesis puede considerarse una relación lineal entre la tensión aplicada a un material y la deformación que sufre, tal y como se describe en la **teoría de la elasticidad lineal** [OdSB06]. Estas restricciones son:

1. La suposición de que los desplazamientos y sus gradientes son pequeños, tal y como se describió anteriormente cuando se expuso la *teoría de pequeñas deformaciones*.
2. La existencia de un estado neutro en el instante t_0 en el que todas las deformaciones y tensiones son nulas, y en el que por tanto:

$$\begin{aligned}\boldsymbol{\sigma}(\mathbf{x}, t_0) &= \mathbf{0} \\ \boldsymbol{\varepsilon}(\mathbf{x}, t_0) &= \mathbf{0}\end{aligned}\tag{4.16}$$

3. La consideración del proceso de deformación como *isotérmico* (que tiene lugar a temperatura constante) y *adiabático* (que no produce calor en ningún momento). En general, los procesos de deformación *lentos* suelen considerarse adiabáticos.

4.3.1 Ecuación constitutiva

Considerando las premisas anteriores, se puede generalizar lo que sucede en el caso elástico unidimensional, en el que la deformación ε es proporcional a la tensión σ aplicada:

$$\sigma = E\varepsilon\tag{4.17}$$

relación que se conoce como *Ley de Hooke*, en la que E es el *módulo elástico* del material. En el caso general, esto se formaliza con una relación lineal entre tensor de tensiones $\boldsymbol{\sigma}$ y de deformaciones $\boldsymbol{\varepsilon}$:

$$\begin{aligned}\boldsymbol{\sigma}(\mathbf{x}, t) &= \mathbf{C} : \boldsymbol{\varepsilon}(\mathbf{x}, t) \\ \sigma_{ij} &= \mathbb{C}_{ijkl} \varepsilon_{kl}\end{aligned}\tag{4.18}$$

obteniéndose la *Ley de Hooke generalizada*, con $:$ el producto tensorial cerrado, los índices tensoriales $i, j, k, l \in \{1, 2, 3\}$, y donde \mathbf{C} es el llamado *tensor de constantes elásticas*.

El tensor \mathbf{C} es un tensor simétrico de cuarto orden que, para materiales isótropos, se define como:

$$\begin{aligned}\mathbf{C} &= \lambda \mathbf{1} \otimes \mathbf{1} + 2\mu \mathbf{I} \\ \mathbb{C}_{ijkl} &= \lambda \delta_{ij} \delta_{kl} + \mu (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk})\end{aligned}\tag{4.19}$$

donde \otimes es el producto tensorial abierto, \mathbf{I} es el tensor simétrico unitario de cuarto orden, cuyos elementos se definen como $I_{ijkl} = \frac{1}{2} [\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}]$, δ_{ij} la delta de Kronecker, y donde λ y μ son las constantes de Lamé, que se obtienen de forma experimental mediante ensayos uniaxiales para cada material. Las 3^4 componentes del tensor de cuarto orden \mathbf{C} se pueden representar en forma matricial distribuyendo los \mathbb{C}_{ijkl} en nueve grupos de nueve elementos si se fijan los índices k y l :

$$\begin{bmatrix} [\mathbb{C}_{ij11}] & [\mathbb{C}_{ij12}] & [\mathbb{C}_{ij13}] \\ [\mathbb{C}_{ij21}] & [\mathbb{C}_{ij22}] & [\mathbb{C}_{ij23}] \\ [\mathbb{C}_{ij31}] & [\mathbb{C}_{ij32}] & [\mathbb{C}_{ij33}] \end{bmatrix}\tag{4.20}$$

con lo que la expresión completa de \mathbf{C} queda:

$$\begin{array}{c|ccc|ccc|ccc} c & 0 & 0 & 0 & \mu & 0 & 0 & 0 & \mu \\ 0 & \lambda & 0 & \mu & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 & 0 & \mu & 0 & 0 \\ \hline 0 & \mu & 0 & \lambda & 0 & 0 & 0 & 0 & 0 \\ \mu & 0 & 0 & 0 & c & 0 & 0 & 0 & \mu \\ 0 & 0 & 0 & 0 & 0 & \lambda & 0 & \mu & 0 \\ \hline 0 & 0 & \mu & 0 & 0 & 0 & \lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu & 0 & \lambda & 0 \\ \mu & 0 & 0 & 0 & \mu & 0 & 0 & 0 & c \end{array}\tag{4.21}$$

donde $c = \lambda + 2\mu$. Por la simetría de $\boldsymbol{\sigma}$ y $\boldsymbol{\varepsilon}$ necesariamente se cumple que $\mathbb{C}_{ijkl} = \mathbb{C}_{jikl}$, $\mathbb{C}_{ijkl} = \mathbb{C}_{ijlk}$ y además $\mathbb{C}_{ijkl} = \mathbb{C}_{klij}$, reduciéndose el número de constantes elásticas de 81 a 21. La condición de isotropía reduce este número a 2.

La expresión de cada σ_{ij} se puede obtener realizando el producto tensorial cerrado de \mathbf{C} y $\boldsymbol{\varepsilon}$, de forma que $\sigma_{ij} = \sum_{kl} \mathbb{C}_{ijkl} \varepsilon_{kl}$, o bien sustituyendo

la ecuación 4.19 en 4.18:

$$\begin{aligned}\boldsymbol{\sigma} &= \lambda \text{Tr}(\boldsymbol{\varepsilon})\mathbf{1} + 2\mu \boldsymbol{\varepsilon} \\ \sigma_{ij} &= \lambda \delta_{ij} \varepsilon_{ll} + 2\mu \varepsilon_{ij}\end{aligned}\tag{4.22}$$

con lo que además se obtiene la *ecuación constitutiva para materiales elásticos isótropos*, con índices tensoriales $i, j \in \{1, 2, 3\}$ y donde $\text{Tr}(\boldsymbol{\varepsilon})$ es la traza de $\boldsymbol{\varepsilon}$. Con esto, las componentes de $\boldsymbol{\sigma}$ quedan:

$$\begin{aligned}\sigma_{xx} &= \lambda(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}) + 2\mu\varepsilon_{xx} \\ \sigma_{yy} &= \lambda(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}) + 2\mu\varepsilon_{yy} \\ \sigma_{zz} &= \lambda(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}) + 2\mu\varepsilon_{zz} \\ \sigma_{xy} &= 2\mu\varepsilon_{xy} \\ \sigma_{yz} &= 2\mu\varepsilon_{yz} \\ \sigma_{zx} &= 2\mu\varepsilon_{zx}\end{aligned}\tag{4.23}$$

con todo lo cual se puede obtener, conocidas las constantes de Lamé del material y el campo tensorial de deformaciones $\boldsymbol{\varepsilon}(\mathbf{x}, t)$, el tensor de tensiones $\boldsymbol{\sigma}(\mathbf{x}, t)$ en cada punto \mathbf{x} del sólido.

4.3.2 Planteamiento del problema elástico lineal

Se denomina *problema elástico lineal* al conjunto de ecuaciones que describen la evolución a lo largo del tiempo un sólido elástico lineal bajo la acción de un campo de fuerzas másicas $\mathbf{b}(\mathbf{x}, t)$ en todo el volumen y un campo de tracción $\mathbf{t}(\mathbf{x}, t)$ en su contorno. El objetivo es conocer la evolución en el tiempo de los campos de desplazamiento $\mathbf{u}(\mathbf{x}, t)$, de deformación $\boldsymbol{\varepsilon}(\mathbf{x}, t)$ y de tensión $\boldsymbol{\sigma}(\mathbf{x}, t)$. Las ecuaciones que describen el problema son:

1. La ecuación de Cauchy del balance de la cantidad de movimiento:

$$\begin{aligned}\nabla \cdot \boldsymbol{\sigma}(\mathbf{x}, t) + \rho_0 \mathbf{b}(\mathbf{x}, t) &= \rho_0 \frac{\partial^2 \mathbf{u}(\mathbf{x}, t)}{\partial t^2} \\ \frac{\partial \sigma_{ij}}{\partial x_i} + \rho_0 b_j &= \rho_0 \frac{\partial^2 u_j}{\partial t^2} \quad j \in \{1, 2, 3\}\end{aligned}\tag{4.24}$$

donde ρ_0 es la densidad del medio, la cual se considera constante en el

tiempo en los problemas de elasticidad lineal. La ecuación de Cauchy resulta en tres ecuaciones en derivadas parciales.

2. La ecuación constitutiva:

$$\begin{aligned}\boldsymbol{\sigma} &= \lambda \text{Tr}(\boldsymbol{\varepsilon})\mathbf{1} + 2\mu \boldsymbol{\varepsilon} \\ \sigma_{ij} &= \lambda \delta_{ij} \varepsilon_{ll} + 2\mu \varepsilon_{ij} \quad j \in \{1, 2, 3\}\end{aligned}\tag{4.25}$$

Por la simetría de los tensores de tensión y deformación en la que sólo seis de las nueve componentes son distintas, la ecuación constitutiva añade al sistema únicamente las seis ecuaciones desarrolladas en 4.23.

3. La ecuación geométrica, que relaciona las deformaciones infinitesimales y los desplazamientos:

$$\begin{aligned}\boldsymbol{\varepsilon}(\mathbf{x}, t) &= \nabla^S \mathbf{u}(\mathbf{x}, t) = \frac{1}{2}(\mathbf{u} \otimes \nabla + \nabla \otimes \mathbf{u}) \\ \varepsilon_{ij} &= \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \quad i, j \in \{1, 2, 3\}\end{aligned}\tag{4.26}$$

que supone seis ecuaciones más en el sistema, por la simetría del tensor deformación $\boldsymbol{\varepsilon}$.

Las ecuaciones 4.24, 4.25 y 4.26 constituyen un sistema de 15 ecuaciones diferenciales en derivadas parciales (EDPs) con 15 incógnitas en tiempo y espacio:

- $\mathbf{u}(\mathbf{x}, t)$ — 3 incógnitas
- $\boldsymbol{\varepsilon}(\mathbf{x}, t)$ — 6 incógnitas
- $\boldsymbol{\sigma}(\mathbf{x}, t)$ — 6 incógnitas

Y, dado que \mathbf{x} tiene tres componentes, el sistema debe ser resuelto por tanto en $\mathbb{R}^3 \times \mathbb{R}^+$, añadiendo las adecuadas condiciones de contorno.

4.4 Teoría incremental de la plasticidad

Existe un comportamiento de los sólidos deformables que no puede ser descrito mediante el modelo elástico. Cuando las tensiones (o las deformaciones) sobrepasan cierto valor umbral en ciertos materiales tiene lugar un

fenómeno de deformación conocido como **elastoplasticidad**, que presenta dos grandes diferencias respecto del comportamiento puramente elástico:

1. Se pierde la *linealidad* entre las deformaciones y las tensiones.
2. La aparición de la *deformación plástica* o *permanente*, que implica que no toda la deformación que se genera mientras se está aplicando la tensión se recupera al liberarse el cuerpo de las fuerzas externas.

Si bien el comportamiento elastoplástico puede ser caracterizado por modelos matemáticos de distinta complejidad, un modelo sencillo y extensamente usado es el propuesto por la **teoría incremental de la plasticidad** [OdSB06]. En esencia se pretende aproximar la respuesta del sólido deformable a una fuerza externa mediante estados lineales, en los que se diferencian la deformación elástica (que puede ser recuperada si la tensión disminuye) de la plástica (que es permanente y no se recupera).

4.4.1 En una dimensión

Para ilustrar el comportamiento elastoplástico se expondrá primero el caso unidimensional, que describe la relación entre tensión y deformación en una barra de material con una fuerza de tracción o compresión en sus extremos (ver [Figura 4.5](#)). En el caso uniaxial la tensión σ es un escalar que mide la fuerza por unidad de área, por lo que se tiene que $\sigma = F/A$, con la fuerza F también un escalar y el área A la sección de la barra de material. La deformación ε se aproxima, según la interpretación física de la misma vista en la [Sección 4.1](#) y la definición de *alargamiento unitario* que allí se proporcionó, por la relación entre la compresión o el alargamiento δ y la longitud de la barra L , por lo que se cumple que $\varepsilon = \delta/L$.

Según la teoría incremental de la plasticidad en una dimensión, la relación entre la tensión aplicada σ y la deformación ε que tiene lugar se puede modelar en trozos que relacionan las dos magnitudes escalares de forma *lineal*, tal y como se aprecia en la [Figura 4.6](#). En un ciclo de carga y descarga completo (marcados en secuencia temporal en la imagen con los puntos 0, 1, 2 y 3) se producen ciertos cambios en la relación entre σ y ε , que vienen determinados por el cambio en la tensión y el *límite elástico* σ_e .

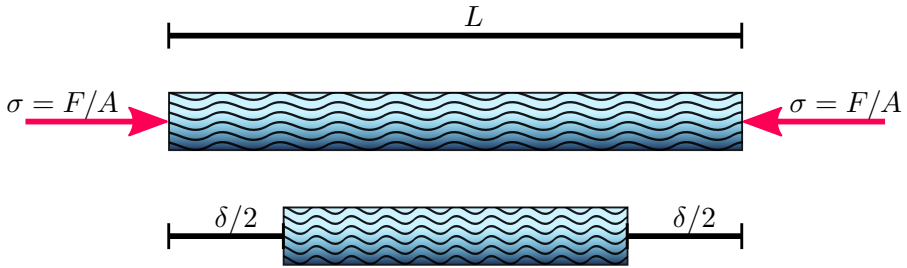


Figura 4.5 – Representación del comportamiento elastoplástico en una dimensión. Una barra de material es sometida a una tensión σ en su eje transversal, por lo que se produce una deformación $\varepsilon = \delta/L$ que modifica su longitud total. Al contrario de lo que pasa con los materiales elásticos, la deformación no guarda una relación lineal con la tensión σ .

En el punto $\boxed{0}$ la tensión y la deformación son nulas. Cuando se aplica una tensión el sólido entra en un estado elástico en el cual las fuerzas son conservativas y la deformación recuperable (tramo $\boxed{0}$ — $\boxed{1}$), y la relación entre ambas es lineal y está caracterizada por el módulo elástico E tal que $\sigma = E\varepsilon$. Si disminuye la tensión σ antes de alcanzarse el *límite elástico inicial* σ_e , la deformación disminuirá coherentemente sin producirse ningún efecto permanente. Una vez alcanzado el límite elástico σ_e (punto $\boxed{1}$) la relación entre σ y ε cambia, entrando el sólido en una rama de carga plástica (tramo $\boxed{1}$ — $\boxed{2}$), en la que se mantiene mientras no disminuya la tensión σ . De producirse una caída en la misma (punto $\boxed{2}$), el sólido retorna a la relación elástica inicial (tramo $\boxed{2}$ — $\boxed{3}$). Cuando la tensión vuelve a ser nula (punto $\boxed{3}$) se recupera la parte elástica de la deformación ε_e , si bien se ha registrado en el sólido una deformación plástica ε_p no recuperable. En un ciclo posterior de carga será necesario llegar a la tensión marcada en el punto $\boxed{2}$ para entrar en la rama elastoplástica, conociéndose este efecto como *endurecimiento*.

La relación entre σ y ε formaliza pues mediante la variable de endurecimiento, para lo cual se definirá la deformación en términos de su tramo

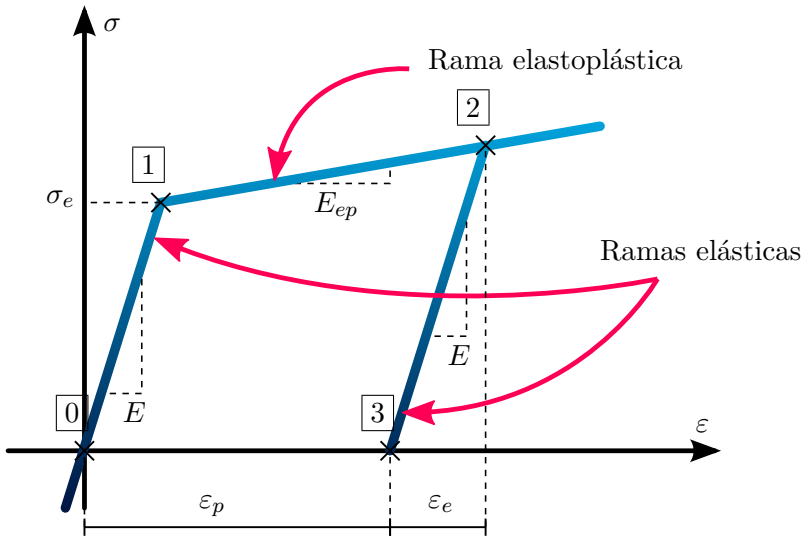


Figura 4.6 – Relación tensión-deformación para el caso unidimensional. Tras la carga en las ramas elástica y plástica (secuencia 0—1—2), se produce un descenso en la tensión en la que se recupera una parte ε_e de la deformación y deja una parte plástica ε_p que no se recupera (punto 3). Nótese que esta evolución no está relacionada de ningún modo con el tiempo o la velocidad de la deformación, y que sencillamente determina qué tensión es necesario aplicar para conseguir un determinado incremento en el desplazamiento.

elástico y plástico:

$$\begin{aligned} \varepsilon &= \varepsilon_e + \varepsilon_p \\ d\varepsilon &= d\varepsilon_e + d\varepsilon_p \end{aligned} \tag{4.27}$$

ecuación que se conoce como la *descomposición aditiva de la deformación*, la cual desglosa la deformación total en la parte elástica ε_e y la plástica ε_p . A la vista de la [Figura 4.6](#) se comprueba fácilmente que:

$$\begin{aligned} \varepsilon_e &= \frac{\sigma}{E} \\ d\varepsilon_e &= \frac{d\sigma}{E} \end{aligned} \tag{4.28}$$

donde E es el módulo elástico del material, y que además es la pendiente de todas las ramas elásticas, por lo que, para cualquier valor de σ , se puede conocer la parte de la deformación ε_e que se recupera. Se define entonces

la variable de endurecimiento $\alpha(\sigma, \varepsilon)$ como:

$$\begin{aligned}\alpha &= |\varepsilon_p| \\ d\alpha &= |d\varepsilon_p|\end{aligned}\tag{4.29}$$

por lo que será siempre positiva, independientemente de que la deformación sea una compresión o un estiramiento. La variable de endurecimiento permite definir la *tensión de fluencia* $\sigma_f(\alpha)$ como:

$$\begin{aligned}\sigma_f(\alpha) &= \sigma_e + H'\alpha \\ d\sigma_f(\alpha) &= H'd\alpha\end{aligned}\tag{4.30}$$

donde H' es el parámetro de endurecimiento. De esta definición se obtiene que $\sigma_f(0) = \sigma_e$, que siempre se cumple que $\sigma_f(\alpha) > 0$ y que $\sigma_f(\alpha)$ es siempre el valor máximo permitido para σ en la zona elástica, aumentando coherentemente con el endurecimiento.

Por conveniencia se define la *función de fluencia plástica*:

$$F(\sigma, \alpha) = |\sigma| - \sigma_f(\alpha)\tag{4.31}$$

que determina dos estados tensionales distintos:

- Si $F(\sigma, \alpha) < 0$ entonces $|\sigma| < \sigma_f(\alpha)$ y σ se encuentra en el *dominio elástico* \mathbb{E}_σ , el cual se define formalmente como:

$$\mathbb{E}_\sigma = \{ \sigma \in \mathbb{R} \mid F(\sigma, \alpha) < 0 \}\tag{4.32}$$

- Si $F(\sigma, \alpha) = 0$ entonces $|\sigma| = \sigma_f(\alpha)$ y σ se encuentra en la *superficie de fluencia* $\partial\mathbb{E}_\sigma$, que se formaliza:

$$\partial\mathbb{E}_\sigma = \{ \sigma \in \mathbb{R} \mid F(\sigma, \alpha) = 0 \}\tag{4.33}$$

y que representa la cota superior de la tensión.

Nótese que no puede suceder que $F(\sigma, \alpha) > 0$, ya que, por la definición 4.30, $\sigma_f(\alpha)$ es el valor tensional más alto alcanzado.

Con todo lo anterior se pueden definir tres situaciones distintas que

determinan completamente el comportamiento del sólido y que permiten construir su **ecuación constitutiva**:

Régimen elástico La tensión σ pertenece al dominio elástico $\sigma \in \mathbb{E}_\sigma$:

$$d\sigma = E d\varepsilon \quad (4.34)$$

Régimen elastoplástico en carga plástica La tensión σ está en la superficie de fluencia $\sigma \in \partial\mathbb{E}_\sigma$ y además el incremento en la tensión es positivo de modo que $dF(\sigma, \alpha) = 0$:

$$d\sigma = E_{ep} d\varepsilon \quad (4.35)$$

Régimen elastoplástico en descarga La tensión σ está en la superficie de fluencia $\sigma \in \partial\mathbb{E}_\sigma$ y además el incremento en la tensión es negativo de modo que $dF(\sigma, \alpha) < 0$:

$$d\sigma = E d\varepsilon \quad (4.36)$$

donde E es el módulo elástico del material y E_{ep} es el *módulo de deformación elastoplástico*, que se deriva del parámetro de endurecimiento H' definido en la ecuación 4.30.

4.4.2 En tres dimensiones

La generalización a tres dimensiones es inmediata siguiendo el mismo razonamiento del caso uniaxial. En primer lugar, se puede realizar la descomposición aditiva del tensor deformación ε :

$$\begin{aligned} \varepsilon &= \varepsilon_e + \varepsilon_p \\ d\varepsilon &= d\varepsilon_e + d\varepsilon_p \end{aligned} \quad (4.37)$$

estando la deformación elástica siempre definida por el tensor de propiedades elásticas \mathbb{C} descrito en la ecuación 4.19:

$$\begin{aligned} \varepsilon_e &= \mathbb{C}^{-1} : \sigma \\ d\varepsilon_e &= \mathbb{C}^{-1} : d\sigma \end{aligned} \quad (4.38)$$

La relación entre la variable de endurecimiento α y la deformación plástica viene determinada por la *regla de flujo*:

$$\begin{aligned} d\boldsymbol{\varepsilon}_p &= \lambda \frac{\partial G(\boldsymbol{\sigma}, \alpha)}{\partial \boldsymbol{\sigma}} \\ d\alpha &= \lambda \end{aligned} \tag{4.39}$$

con $\alpha \in [0, \infty)$, y donde λ recibe el nombre de *multiplicador plástico* y $G(\boldsymbol{\sigma}, \alpha)$ el de *función de potencial plástico*.

La *tensión de fluencia* se define de forma completamente similar a su homónima unidimensional:

$$\begin{aligned} \sigma_f(\alpha) &= \sigma_e + H' \alpha \\ d\sigma_f(\alpha) &= H' d\alpha \end{aligned} \tag{4.40}$$

siendo σ_e el mismo umbral elástico inicial obtenido en ensayo uniaxial. H' juega el mismo papel que en la contrapartida unidimensional, determinando el cambio del límite elástico a medida que crece α . Con todo esto la *función de fluencia plástica* queda:

$$F(\boldsymbol{\sigma}, \alpha) = \phi(\boldsymbol{\sigma}) - \sigma_f(\alpha) \tag{4.41}$$

donde $\phi(\boldsymbol{\sigma})$ recibe el nombre de *tensión uniaxial equivalente*. Se define por tanto el *dominio elástico* \mathbb{E}_σ como:

$$\mathbb{E}_\sigma = \{ \boldsymbol{\sigma} \in \mathbb{R} \mid F(\boldsymbol{\sigma}, \alpha) < 0 \} \tag{4.42}$$

y la *superficie de fluencia* $\partial\mathbb{E}_\sigma$ como:

$$\partial\mathbb{E}_\sigma = \{ \boldsymbol{\sigma} \in \mathbb{R} \mid F(\boldsymbol{\sigma}, \alpha) = 0 \} \tag{4.43}$$

quedando por tanto bien definido el límite entre el espacio de tensiones de la zona elástica y el de la zona elastoplástica.

La **ecuación constitutiva** distingue las mismas tres situaciones que el caso uniaxial, quedando la misma como sigue:

Régimen elástico La tensión $\boldsymbol{\sigma}$ pertenece al dominio elástico $\boldsymbol{\sigma} \in \mathbb{E}_\sigma$:

$$d\boldsymbol{\sigma} = \mathbf{C} : d\boldsymbol{\varepsilon} \quad (4.44)$$

Régimen elastoplástico en carga plástica La tensión $\boldsymbol{\sigma}$ está en la superficie de fluencia $\boldsymbol{\sigma} \in \partial\mathbb{E}_\sigma$ y además el incremento en la tensión es positivo de modo que $dF(\boldsymbol{\sigma}, \alpha) = 0$:

$$d\boldsymbol{\sigma} = \mathbf{C}_{ep} : d\boldsymbol{\varepsilon} \quad (4.45)$$

Régimen elastoplástico en descarga La tensión $\boldsymbol{\sigma}$ está en la superficie de fluencia $\boldsymbol{\sigma} \in \partial\mathbb{E}_\sigma$ y además el incremento en la tensión es negativo de modo que $dF(\boldsymbol{\sigma}, \alpha) < 0$:

$$d\boldsymbol{\sigma} = \mathbf{C} : d\boldsymbol{\varepsilon} \quad (4.46)$$

donde \mathbf{C} es el mismo tensor de propiedades elásticas usado en 4.38, y \mathbf{C}_{ep} es el *tensor constitutivo elastoplástico*, que registra tanto las propiedades elásticas del medio como el endurecimiento del material, y que se define como:

$$\begin{aligned} \mathbf{C}_{ep}(\boldsymbol{\sigma}, \alpha) &= \mathbf{C} - \frac{\mathbf{C} : \frac{\partial G}{\partial \boldsymbol{\sigma}} \otimes \frac{\partial F}{\partial \boldsymbol{\sigma}} : \mathbf{C}}{H' + \frac{\partial F}{\partial \boldsymbol{\sigma}} : \mathbf{C} : \frac{\partial G}{\partial \boldsymbol{\sigma}}} \\ \mathbb{C}_{ijkl}^{ep} &= \mathbb{C}_{ijkl} - \frac{\mathbb{C}_{ijpq} \frac{\partial G}{\partial \sigma_{pq}} \frac{\partial F}{\partial \sigma_{rs}} \mathbb{C}_{rskl}}{H' + \frac{\partial F}{\partial \sigma_{pq}} \mathbb{C}_{pqrs} \frac{\partial G}{\partial \sigma_{rs}}} \end{aligned} \quad (4.47)$$

con índices tensoriales $i, j, k, l, p, q, r, s \in \{1, 2, 3\}$. Como se aprecia en la ecuación, el tensor \mathbf{C}_{ep} recoge en su definición todos los elementos que rigen el límite elástico: el parámetro de endurecimiento H' , la función de potencial plástico G de la ecuación 4.39 y la función de fluencia F .

El problema plástico queda completo cuando se define la función de fluencia o *criterio de fallo*, entendido este como el fin del comportamiento

elástico. Esta se define, a tenor de lo descrito en la ecuación 4.41, mediante la función tensión uniaxial equivalente $\phi(\boldsymbol{\sigma}) : \mathbb{R}^9 \rightarrow \mathbb{R}$ que compone los elementos de $\boldsymbol{\sigma}$ en un escalar. Según el *criterio de Von Mises* esta se define como:

$$\begin{aligned}\phi(\boldsymbol{\sigma}) &\equiv \sqrt{\frac{3}{2} \boldsymbol{\sigma}' : \boldsymbol{\sigma}'} \\ \phi(\boldsymbol{\sigma}) &\equiv \sqrt{\frac{3}{2} \sigma'_{ij} \sigma'_{ij}}\end{aligned}\tag{4.48}$$

donde $\boldsymbol{\sigma}'$ es la *tensión desviadora*, que se calcula en función de $\boldsymbol{\sigma}$ y su traza $Tr(\boldsymbol{\sigma})$ tal que:

$$\begin{aligned}\boldsymbol{\sigma}' &\equiv \boldsymbol{\sigma} - \frac{1}{3} Tr(\boldsymbol{\sigma}) \mathbf{1} \\ \sigma'_{ij} &\equiv \sigma_{ij} - \delta_{ij} \frac{1}{3} \sigma_{ll}\end{aligned}\tag{4.49}$$

Capítulo 5

Fuerza

De las dos características sensoriales que procurará el entorno de simulación que se pretende construir la sensación táctil es sin duda la menos común. Es usual generar imágenes que permitan al usuario *ver* en pantalla el proceso de deformación provocado mediante la interacción de algún dispositivo de entrada, si bien son necesarias ciertas condiciones y hardware específico para que se pueda *tocar* el material que se está deformando.

Los periféricos especiales de salida que permiten generar una sensación táctil virtual se conocen como **dispositivos hápticos**. Se tratan por norma general de brazos robóticos articulados con un elemento de sujeción o **mango** que se sitúa en su extremo, tal y como se aprecia en el ejemplo de la [Figura 5.1](#). Entre el mango y el brazo robótico existe un punto llamado **puntero háptico** o **HIP** (del inglés *Haptic Interface Point*), el cual se interpreta en la computadora como un **cursor** virtual que se mueve en un espacio tridimensional. El puntero puede moverse libremente por la zona del espacio delimitada por la longitud y margen de flexión de las articulaciones del brazo robótico.

No obstante la utilidad fundamental de los dispositivos hápticos no es la captura digital de la posición del puntero, sino su capacidad de generar una sensación táctil artificial mediante un ordenador. El brazo robótico permite generar un *empuje* de intensidad variable gracias a los motores situados en sus articulaciones. Esta reacción puede interpretarse de varias formas según evolucione la intensidad del empuje, y en términos generales se calcula



Figura 5.1 – Dispositivo háptico (foto cortesía de SensAble Technologies). El brazo robótico articulado posee un mango que permite sujetar el dispositivo y mover el puntero háptico.

considerando la variación de la posición del puntero y/o la orientación del mango. Se denominará **fuerza de respuesta** o sencillamente *fuerza* a esta actuación del dispositivo háptico. La fuerza es perceptible cuando se mueve el cursor a una zona del espacio virtual que active la sensación táctil, o, dicho de otro modo, en la que haya algo que tocar o percibir.

Si bien en el [Capítulo 2](#) se describieron algunas técnicas complejas de generación de la sensación táctil al interactuar con medios elásticos, se pueden extraer y aislar los principios utilizados para el cálculo de la fuerza en estos trabajos y clasificarlos por su funcionamiento elemental:

Basados en la posición relativa a una entidad — Se puede calcular la fuerza considerando, por ejemplo, la distancia relativa de la posición virtual del puntero a la superficie del objeto más cercano. Este es el principio de las técnicas que generan una presión hacia el exterior del objeto cuando el cursor *invade* el espacio que estaría ocupado por el interior del cuerpo virtual. Sobre esta idea se construyen las técnicas basadas en *proxy* y los *métodos del campo vectorial*, pues ambas generan una respuesta en fuerza proporcional a la distancia a un punto determinado de la superficie, siguiendo la Ley de Hook.

Basados en la posición absoluta del cursor y su entorno — Es posible también simular campos de fuerza que, para cada punto del espacio, generan un vector de fuerza determinado que es previamente conocido o que se puede calcular con la información del medio simulado. Avila y Sobierajsky [AS96] usaron esta idea para su método de exploración háptica, en tanto que una de las componentes de la fuerza resultante depende únicamente de la densidad del entorno.

Basados en el movimiento del cursor — Existen fuerzas naturales que dependen exclusivamente del incremento en la posición o en la velocidad de los elementos simulados. Por ejemplo, la simulación de una masa en movimiento requiere conocer la aceleración del cuerpo, pues la fuerza necesaria para moverla o frenarla depende de la aceleración en cada instante. En estos casos el entorno no entra en juego en el cálculo, pues los valores de la fuerza dependen exclusivamente del historial de posiciones en el tiempo.

Obviamente y como de hecho han demostrado las técnicas expuestas en el [Capítulo 2](#) estos principios pueden combinarse y extenderse para generar estrategias distintas, que producirán una sensación táctil adecuada para cada caso y entorno. Es por tanto una de las tareas más importantes de este trabajo encontrar una composición adecuada de estos principios. Esta composición debe permitir trasladar el comportamiento teórico que se desarrolló en el [Capítulo 4](#) a la rutina de trabajo del dispositivo háptico, intentando siempre conseguir la mayor fidelidad a la descripción teórica y la mayor eficiencia en los cálculos que sea posible.

5.1 El flujo de trabajo háptico

Salvo contadas excepciones que se pueden encontrar entre los modelos de alta gama, los dispositivos hápticos ofrecen seis grados de libertad en los parámetros de entrada y tres en los de salida (lo que se etiqueta en inglés como "*6DoF input / 3DoF output*", donde *DoF* son las siglas de *Degree of Freedom*). Los grados de libertad de los parámetros de entrada se refieren a la posición del puntero, representada como un vector en el espacio tridimen-

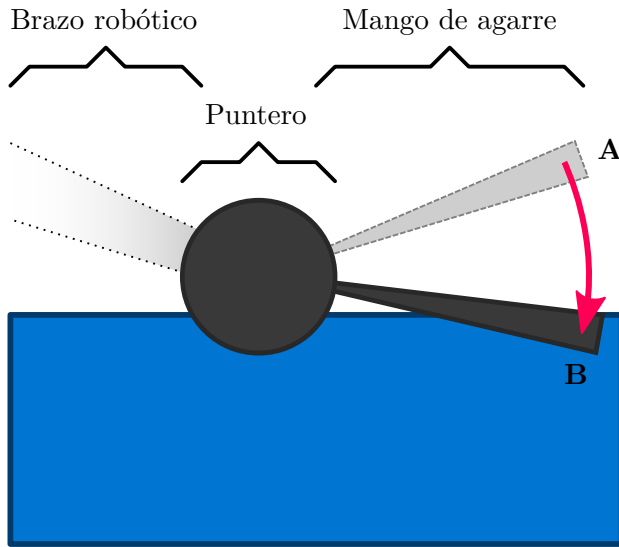


Figura 5.2 – Esquema de la situación anómala en la que el mango real invade el espacio ocupado por el cuerpo virtual cuando el usuario pivota desde la postura original **A** hasta la **B** mientras el puntero permanece en una posición inalterada. Se puede crear una fuerza de respuesta cuando la representación virtual del puntero se encuentra con el material, si bien no se puede enviar respuesta háptica alguna ante el giro del mango si sólo se dispone de una salida con tres grados de libertad.

sional, y a la dirección del mango, que se describe como un vector unitario. La salida únicamente permite generar un vector tridimensional, lo cual se traduce en un vector de fuerza con dirección y módulo. Es por tanto común en los dispositivos de media y baja gama que, aunque se permita leer la dirección del mango, no sea posible *bloquearla* ni ofrecer resistencia alguna a su movimiento. Esto limita la experiencia táctil que se puede conseguir con tres grados de libertad en la salida. Por ejemplo, en una simulación de un medio líquido no se puede reproducir la sensación que se percibe al mover el mango dentro del líquido si su extremo permanece en una posición constante. De igual forma no se puede simular la fricción al rotar una herramienta, como por ejemplo un destornillador al apretar o aflojar un tornillo. Adicionalmente, determinadas simulaciones pueden derivar en situaciones anómalas como la que se puede apreciar en la [Figura 5.2](#), en las que el mango debería *chocar* con el objeto simulado. No puede ofrecerse sin embargo resistencia a dicho movimiento al no disponerse de motores que

controlen la rotación de la barra de agarre.

Formalmente, el dispositivo háptico permite leer la posición muestreada del puntero como un vector $\vec{c} \in \mathbb{R}^3$, el cual representa el cursor virtual en cada instante discreto $t_k = k\Delta T$, donde $k \in \mathbb{N}$ y ΔT es el periodo de muestreo. Además, el mango de agarre se representa como otro vector $\hat{o} \in \mathbb{R}^3$, y que indica la orientación o dirección del mango con el que el usuario mueve el puntero. El cálculo de la fuerza de salida $\vec{f}_h \in \mathbb{R}^3$ que se enviará al usuario se realiza en una función de cálculo llamada *callback háptico*, que se notará como h , y que puede usar como parámetros de entrada las lecturas de posición y orientación que ofrece el dispositivo háptico, así como cualquier información relevante que se requiera del medio simulado. La función de cálculo se puede definir por tanto como:

$$h(\vec{c}, \hat{o}, \omega) : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^n \rightarrow \mathbb{R}^3 \quad (5.1)$$

donde $\omega \in \mathbb{R}^n$ es un conjunto de cardinal n que contiene la información relevante del entorno, tales como vectores de distancia a los objetos, gradientes o cualquier otra magnitud que sea vinculante para el cómputo de la fuerza \vec{f}_h .

Los manuales de uso de los dispositivos hápticos recomiendan actualizar el valor de \vec{f}_h 1000 veces por segundo [Sen08]. Como todo dispositivo de salida, el aparato háptico emite valores discretos, por lo que esta frecuencia de 1kHz en la ejecución de la función h es necesaria para que el sistema nervioso no detecte discontinuidades en una variación progresiva de la fuerza [LD07]. Es necesario también que no se generen valores muy dispares en un intervalo corto de tiempo. Una variación brusca de la dirección o del módulo de \vec{f}_h podría ser percibida como un cambio anormal por el usuario, por lo que el método de cálculo de la función h debe procurar cambios suaves en la estimación de la fuerza.

Es necesario hacer notar que cada valor de la fuerza de respuesta \vec{f}_h se produce *después* de haber recogido los valores de posición \vec{c} , rotación \hat{o} y de información del entorno ω . Este ciclo de *obtención de muestras de entrada* \rightarrow *realización de cálculos* \rightarrow *envío de los valores de salida* es intrínseco al modelo actual de computación, de naturaleza discreta y secuencial, y supone además que la fuerza de respuesta estará retrasada al menos un ciclo

respecto de la configuración $\{\vec{c}, \hat{o}, \omega\}$ que la determina. Si bien el retraso no debería ser perceptible con frecuencias altas de actualización, es importante ser consciente de esta limitación a la hora de diseñar cualquier modelo de simulación que procure la máxima fidelidad al proceso original.

5.2 Adaptación del marco teórico al entorno de simulación

En la exposición que se hizo en el [Capítulo 4](#) de la teoría de la elasticidad lineal y de la teoría incremental de la plasticidad se puso de manifiesto que existen dos incógnitas elementales a la hora de resolver de forma teórica la interacción con un medio elastoplástico: la *tensión* σ y la *deformación* ϵ . El problema se puede resolver en cualquiera de las dos direcciones, bien planteado en tensiones, cuando la deformación es conocida; o bien planteado en desplazamientos, cuando lo que se conoce son las fuerzas aplicadas [[OdSB06](#)].

Lo ideal en el entorno virtual de simulación que se propone sería conocer las tensiones superficiales que se aplican sobre el cuerpo, pues por una parte la forma del material una vez deformado no es conocida, y por otra el cálculo de la fuerza de respuesta sería trivial, según lo dispuesto por la tercera ley de Newton. Conocida la magnitud y dirección de la fuerza que se invierte en deformar el cuerpo, la fuerza de reacción, y consecuentemente la que debería percibir el usuario, es de igual magnitud y de sentido opuesto. No obstante, y tal y como se ha explicado anteriormente, el dispositivo háptico no ofrece ninguna estimación de la fuerza aplicada por el usuario, estando disponibles únicamente los puntos del espacio por los que pasa el cursor. En este contexto se podría aplicar la segunda ley de Newton [[SZY88](#)]:

$$\vec{F} = \frac{d\vec{p}}{dt} \quad (5.2)$$

donde \vec{p} es la cantidad de movimiento, que se define como:

$$\vec{p} = m\vec{v} \quad (5.3)$$

siendo m la masa del cuerpo que ejerce la fuerza y \vec{v} su velocidad. Sabiendo

que por definición la aceleración instantánea es la derivada en el tiempo de la velocidad:

$$\vec{a} = \frac{d\vec{v}}{dt} \quad (5.4)$$

se tiene que:

$$\vec{F} = m\vec{a} \quad (5.5)$$

y que por tanto la fuerza que ejerce un cuerpo en movimiento (tal y como sería la herramienta que sujeta el usuario) se puede calcular conocida su masa y su aceleración. Si bien esta solución ha sido aplicada satisfactoriamente en dispositivos uniaxiales [SMS⁺01], su generalización tridimensional plantea dos problemas, uno teórico y uno práctico, que hacen que estimar la fuerza así sea poco fiable:

1. La masa m de la herramienta no es ni conocida ni constante. No se conoce *a priori* la masa conjunta del brazo robótico y del brazo del usuario que sujeta el puntero. Esta sería en cualquier caso variable, pues dependería del apoyo del usuario y de la postura del brazo robótico.
2. Si bien teóricamente sería posible calcular la aceleración del puntero háptico, pues sus posiciones en cada instante t_k en el que se muestrea son conocidas, no es en absoluto trivial calcular ni la velocidad ni la aceleración. La naturaleza discreta del muestreo hace que las trayectorias no sean continuas ni derivables. Las velocidades y aceleraciones instantáneas, obtenidas mediante diferencia entre posiciones consecutivas, presentan grandes cambios de dirección, al no ser la posición muestreada exactamente parte de la trayectoria. Esto se ilustra en la [Figura 5.3](#), en la que se muestra una trayectoria real y las muestras capturadas por el dispositivo. Si bien es posible filtrar el ruido generado por la discretización y obtener una estimación uniforme de la aceleración, se introducirían de forma inevitable simplificaciones y/o retrasos, lo que sería en cualquier caso un handicap en una simulación realista. A tenor de esta cuestión se realizó el estudio presentado en el [Apéndice B](#), en el que se profundiza en este problema y se presentan algunas soluciones para reconstruir la trayectoria y sus parámetros cinéticos.

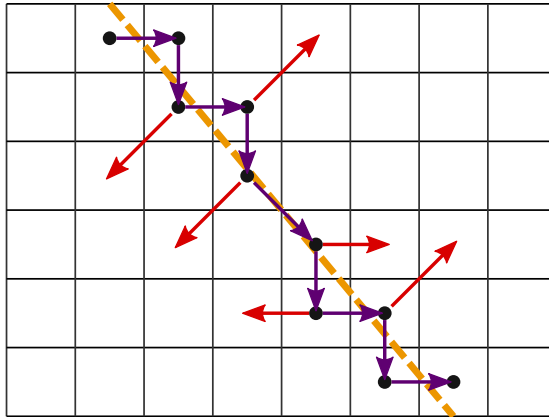


Figura 5.3 – Esquema simplificado en 2D de un espacio discreto de posicionamiento. La recta naranja punteada es la trayectoria rectilínea real trazada por el usuario. Los puntos muestreados se muestran en negro. Se marcan además los vectores de velocidad (en magenta, unen los puntos en los que se detecta) y de aceleración (vectores rojos). Tanto la velocidad como la aceleración deberían presentar la misma dirección que la trayectoria, si bien las aproximaciones derivadas de la discretización generan claras anomalías en el cálculo de ambas magnitudes.

Con estas cuestiones abiertas es evidente que no se puede conocer de forma exacta ni fiable la fuerza que aplica el usuario sobre la superficie del sólido, y consecuentemente tampoco el campo de tensiones $\sigma(\mathbf{x}, t)$.

Como se ha comentado anteriormente es también desconocido el campo de deformaciones $\epsilon(\mathbf{x}, t)$ que determina los desplazamientos $\mathbf{u}(\mathbf{x}, t)$ de las partículas del cuerpo, con lo cual el problema no se puede resolver mediante el enfoque clásico ni en una dirección ni en la otra.

En este entorno tan incierto se hacen por tanto necesarias ciertas simplificaciones que permitirán de un modo u otro operar para encontrar un camino para reconstruir los campos de deformación y de tensión, y por tanto para estimar la fuerza de respuesta.

5.3 Estimación de los campos tensoriales

Se plantea pues como necesaria la estimación de alguno de los campos tensoriales, ya sea el de las tensiones internas $\sigma(\mathbf{x}, t)$ o el de las deformaciones de las partículas $\epsilon(\mathbf{x}, t)$, de forma que se establezca un punto de partida lo

más fiable posible de cara a la ejecución de la simulación.

Para estimar el primero es condición necesaria conocer todas las fuerzas que actúan sobre el cuerpo, siendo la mayor incógnita la que aplica el usuario al manipular el dispositivo háptico. Considerando la discusión de la sección anterior, en el mejor de los casos se podría disponer de una estimación fiable de la aceleración, si bien quedaría abierta la cuestión de la masa con la que se aplica la fuerza, altamente variable e imposible de estimar sin aparatos de medida adicionales. Al no ser posible decidir si el usuario está sujetando el mango con dos dedos o si está usando todo el peso de su brazo, una estimación de la fuerza usando este principio no sería en absoluto útil.

El campo de deformaciones tampoco es conocido en su totalidad, pues para ello habría que predecir de antemano la posición final de todas las partículas del cuerpo tras una interacción con la herramienta simulada. No obstante, es posible estimar parte de la deformación macroscópica con un método heurístico, y considerar dicha configuración como un campo de deformaciones válido. La respuesta en fuerza podría pues inferirse del campo de tensiones asociado a dicha deformación *aproximada*.

5.3.1 Aproximación del campo de deformaciones

La heurística de estimación de $\varepsilon(\mathbf{x}, t)$ partirá de un principio físico fácilmente comprobable: el desplazamiento de partículas que tiene lugar en la zona de contacto de la herramienta simulada con la superficie del sólido virtual. Cuando la herramienta presiona el sólido la superficie se deforma coherentemente siguiendo la forma de la misma, arrastrando con ella las partículas circundantes. Esto se puede comprobar a la vista de los resultados experimentales obtenidos por Prevost [Pre84]. En su trabajo, en el cual se resuelve formalmente el problema mecánico de la deformación sobre un modelo de elementos finitos, se comprueba el efecto que tiene la aplicación de presión con una herramienta sobre un material deformable. La [Figura 5.4](#) ilustra este fenómeno.

Partiendo de esto y ante la falta de información adicional sobre el estado deformado del sólido, se supondrá que toda la fuerza aplicada se invierte en crear los desplazamientos generados por la superficie de contacto entre la herramienta y el sólido. Se obviarán las deformaciones en la superficie

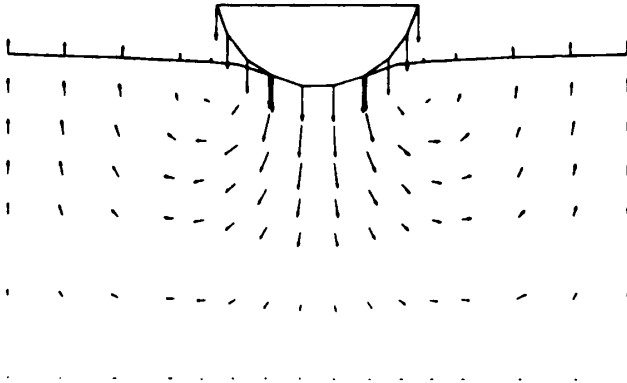


Figura 5.4 – Resultados la simulación bidimensional de Prevost al aplicar una herramienta circular sobre un medio plástico. Se muestra el campo de velocidad obtenido con tres bandas reflectantes en los bordes inferior, izquierdo y derecho.

periférica a la herramienta, sobre las cuales no se ha aplicado presión y que pueden producirse bajo determinadas condiciones de contorno. También se descartará para estos cálculos cualquier otra deformación en el interior del sólido. Se obtiene de este modo un campo de desplazamientos estimado y parcial $\mathbf{u}_{\tilde{S}}(\mathbf{x}, t)$ para la superficie como el que se ilustra en la [Figura 5.5](#), sobre el cual se pueden aplicar las ecuaciones constitutivas del problema elástico para calcular la tensión que las genera.

5.3.2 Inferencia del campo de tensiones

Para obtener un campo de tensiones válido asociado al campo de deformaciones propuesto, supóngase una herramienta cuadrada que se aplica paralela a una superficie horizontal y plana. Sea también el material elastoplástico un sistema discreto de partículas dispuestas de forma regular tal y como se muestra en la [Figura 5.6](#). La presión de la herramienta sobre esta superficie mueve uniformemente las $k \times k$ partículas con las que está en contacto, obteniéndose así un campo de desplazamientos como el que se muestra en la ilustración. Si el vector $\vec{u} \equiv (u_x, u_y, u_z)$ es la distancia recorrida por la herramienta desde que entra en contacto con la superficie hasta el instante t y S_t el área abarcada por la herramienta, es entonces trivial

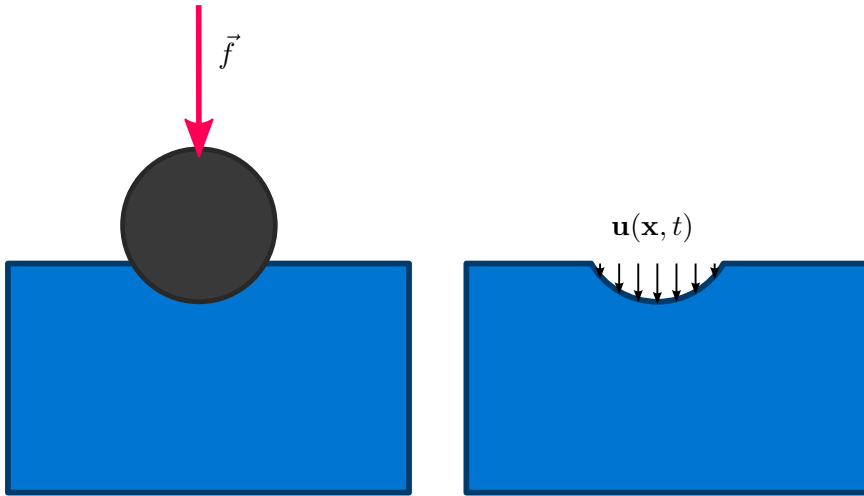


Figura 5.5 – Campo de desplazamientos estimado usando los movimientos de partículas forzadas por el contacto de la herramienta. A la izquierda, la herramienta presiona sobre la superficie con una fuerza \vec{f} . A la derecha, el campo de desplazamientos estimado $\vec{u}_S(\mathbf{x}, t)$ en la superficie, con el que se puede obtener el campo de deformaciones asociado según lo descrito en las ecuaciones 4.1 y 4.2.

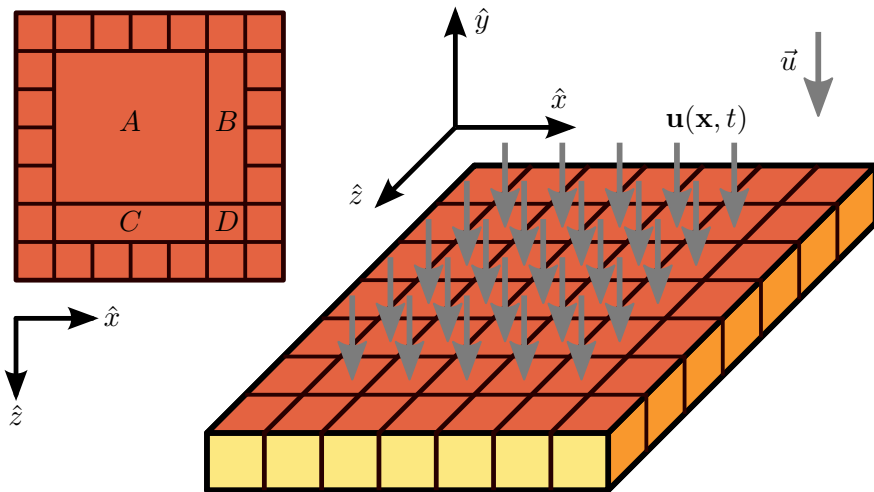


Figura 5.6 – Superficie horizontal sobre el que se aplica un campo de desplazamientos $\mathbf{u}(\mathbf{x}, t)$ constante y localizado. Cada cubo representa una partícula \mathbf{x} . Todas las partículas se desplazan uniformemente con la dirección y módulo de \vec{u} . En la vista bidimensional se dividen las partículas en cuatro zonas A, B, C y D.

que:

$$\mathbf{u}(\mathbf{x}, t) = \begin{cases} \vec{u} & \mathbf{x} \in S_t \\ 0 & \mathbf{x} \notin S_t \end{cases} \quad (5.6)$$

con lo que se puede obtener el campo de deformaciones $\boldsymbol{\varepsilon}(\mathbf{x}, t)$ asociado a dicha deformación. El tensor \mathbf{J} se calcula, siguiendo la definición dada en 4.1, como la variación del desplazamiento \mathbf{u} al moverse en las direcciones \hat{x} , \hat{y} y \hat{z} :

$$\mathbf{J}(\mathbf{x}, t) = \begin{pmatrix} \frac{\partial u_x}{\partial x} & \frac{\partial u_y}{\partial x} & \frac{\partial u_z}{\partial x} \\ \frac{\partial u_x}{\partial y} & \frac{\partial u_y}{\partial y} & \frac{\partial u_z}{\partial y} \\ \frac{\partial u_x}{\partial z} & \frac{\partial u_y}{\partial z} & \frac{\partial u_z}{\partial z} \end{pmatrix} \quad (5.7)$$

siendo la primera fila del tensor la variación al avanzar en \hat{x} , la segunda la variación al avanzar en \hat{y} , y la tercera la variación al avanzar en \hat{z} . En un sistema discreto se calcula por tanto con la diferencia en \mathbf{u} de cada partícula \mathbf{x} respecto de sus vecinas inmediatas en los tres ejes. Se puede además particionar S_t en cuatro zonas A , B , C y D tal y como se detalla en la Figura 5.6, en las cuales se comprueba fácilmente que el tensor \mathbf{J} es uniforme, por lo que su expresión quedaría:

$$\mathbf{J}(\mathbf{x}, t) = \begin{cases} \begin{pmatrix} 0 & 0 & 0 \\ -u_x & -u_y & -u_z \\ 0 & 0 & 0 \end{pmatrix} & \mathbf{x} \in A \\ \begin{pmatrix} -u_x & -u_y & -u_z \\ -u_x & -u_y & -u_z \\ 0 & 0 & 0 \end{pmatrix} & \mathbf{x} \in B \\ \begin{pmatrix} 0 & 0 & 0 \\ -u_x & -u_y & -u_z \\ -u_x & -u_y & -u_z \end{pmatrix} & \mathbf{x} \in C \\ \begin{pmatrix} -u_x & -u_y & -u_z \\ -u_x & -u_y & -u_z \\ -u_x & -u_y & -u_z \end{pmatrix} & \mathbf{x} \in D \end{cases} \quad (5.8)$$

Nótese que $\mathbf{J}(\mathbf{x}, t)$ puede no ser nulo para algunas partículas \mathbf{x} que no estén en el área S_t . No obstante puede obviarse su cálculo, pues como más adelante se justificará, únicamente es relevante el campo de tensiones de las partículas que sufren desplazamiento.

Es entonces inmediato el cálculo de $\boldsymbol{\varepsilon}(\mathbf{x}, t)$, pues según lo que se definió en la ecuación 4.2, se calcula realizando el promedio del tensor \mathbf{J} y su traspuesto \mathbf{J}^T para cada partícula \mathbf{x} , por lo que, definido igualmente por partes, su expresión queda:

$$\boldsymbol{\varepsilon}(\mathbf{x}, t) = \left\{ \begin{array}{l} \left(\begin{array}{ccc} 0 & -\frac{u_x}{2} & 0 \\ -\frac{u_x}{2} & -u_y & -\frac{u_z}{2} \\ 0 & -\frac{u_z}{2} & 0 \end{array} \right) & \mathbf{x} \in A \\ \left(\begin{array}{ccc} -u_x & \frac{-u_y - u_x}{2} & -\frac{u_z}{2} \\ \frac{-u_y - u_x}{2} & -u_y & -\frac{u_z}{2} \\ -\frac{u_z}{2} & -\frac{u_z}{2} & 0 \end{array} \right) & \mathbf{x} \in B \\ \left(\begin{array}{ccc} 0 & -\frac{u_x}{2} & -\frac{u_x}{2} \\ -\frac{u_x}{2} & -u_y & \frac{-u_z - u_y}{2} \\ -\frac{u_x}{2} & \frac{-u_z - u_y}{2} & -u_z \end{array} \right) & \mathbf{x} \in C \\ \left(\begin{array}{ccc} -u_x & \frac{-u_y - u_x}{2} & \frac{-u_z - u_x}{2} \\ \frac{-u_y - u_x}{2} & -u_y & \frac{-u_z - u_y}{2} \\ \frac{-u_z - u_x}{2} & \frac{-u_z - u_y}{2} & -u_z \end{array} \right) & \mathbf{x} \in D \end{array} \right. \quad (5.9)$$

pudiéndose obviar también el valor de $\boldsymbol{\varepsilon}(\mathbf{x}, t)$ para todas las partículas fuera del área S_t .

Conocido el campo de deformaciones se puede obtener el campo de tensiones asociado haciendo uso de la ecuación constitutiva correspondiente.

En virtud de la Ley de Hooke generalizada descrita en la ecuación 4.18 y conocido el tensor \mathbb{C} se puede calcular $\boldsymbol{\sigma} = \lambda \text{Tr}(\boldsymbol{\varepsilon})\mathbf{1} + 2\mu \boldsymbol{\varepsilon}$ (ver ecuación 4.22), por lo que para el tramo elástico de la deformación se obtiene:

$$\boldsymbol{\sigma}(\mathbf{x}, t) = \left\{ \begin{array}{l} \begin{pmatrix} -\lambda u_y & -\mu u_x & 0 \\ -\mu u_x & -2\mu u_y & -\mu u_z \\ 0 & -\mu u_z & -\lambda u_y \end{pmatrix} & \mathbf{x} \in A \\ \begin{pmatrix} \lambda(-u_y - u_x) & \mu(-u_y - u_x) & -\mu u_z \\ -2\mu u_x & \lambda(-u_y - u_x) & -\mu u_z \\ \mu(-u_y - u_x) & -2\mu u_y & -\mu u_z \\ -\mu u_z & -\mu u_z & \lambda(-u_y - u_x) \end{pmatrix} & \mathbf{x} \in B \\ \begin{pmatrix} \lambda(-u_z - u_y) & -\mu u_x & -\mu u_x \\ -\mu u_x & \lambda(-u_z - u_y) & \mu(-u_z - u_y) \\ -\mu u_x & -2\mu u_y & \mu(-u_z - u_y) \\ -\mu u_x & \mu(-u_z - u_y) & \lambda(-u_z - u_y) \\ & & -2\mu u_z \end{pmatrix} & \mathbf{x} \in C \\ \begin{pmatrix} \lambda(-u_z - u_y) & \mu(-u_y - u_x) & \mu(-u_z - u_x) \\ -\lambda u_x - 2\mu u_x & \lambda(-u_z - u_x) & \mu(-u_z - u_y) \\ \mu(-u_y - u_x) & -\lambda u_y - 2\mu u_y & \lambda(-u_y - u_x) \\ \mu(-u_z - u_x) & \mu(-u_z - u_y) & -\lambda u_z - 2\mu u_z \end{pmatrix} & \mathbf{x} \in D \end{array} \right. \quad (5.10)$$

Cada tensión $\boldsymbol{\sigma}$ del campo representa el tensor de tensiones de Cauchy (ver ecuación 4.14) para la partícula \mathbf{x} asociada. Tal y como se justificó el razonamiento que derivó en la expresión 4.15 se puede obtener el vector de tracción de cada partícula $\mathbf{t}(\mathbf{x}, \mathbf{n}, t)$ con sólo realizar el producto tensorial cerrado con la normal \mathbf{n} . Sabiendo además que la tensión es la fuerza por unidad de superficie, se puede integrar el campo vectorial de tracciones \mathbf{t} para obtener la fuerza total equivalente (ecuación 4.5), por lo que la fuerza \vec{f} que habría generado estos desplazamientos se puede calcular con la siguiente

expresión:

$$\vec{f} = \sum_{\mathbf{x} \in S_t} \boldsymbol{\sigma}(\mathbf{x}, t) \mathbf{n} \quad (5.11)$$

Al ser la normal \mathbf{n} constante para toda la superficie S_t , se puede integrar fácilmente en superficie para obtener una expresión genérica de la tensión sobre la superficie S_t . Sabiendo que la herramienta es cuadrada de lado k partículas, y que por tanto A contiene $(k - 1)^2$ partículas, B y C $k - 1$ partículas, y D una única partícula, se puede escribir:

$$\begin{aligned} \sum_{\mathbf{x} \in S_t} \boldsymbol{\sigma}(\mathbf{x}, t) &= (k - 1)^2 \boldsymbol{\sigma}(\mathbf{x}_A) \\ &+ (k - 1) \boldsymbol{\sigma}(\mathbf{x}_B) \\ &+ (k - 1) \boldsymbol{\sigma}(\mathbf{x}_C) \\ &+ \boldsymbol{\sigma}(\mathbf{x}_D) \end{aligned} \quad (5.12)$$

donde \mathbf{x}_A , \mathbf{x}_B , \mathbf{x}_C y \mathbf{x}_D representan una partícula de A , B , C y la única partícula de D , respectivamente. Desarrollando la expresión se obtiene:

$$\begin{aligned} \sum_{\mathbf{x} \in S_t} \boldsymbol{\sigma}(\mathbf{x}, t) &= \\ &\begin{pmatrix} -\lambda k^2 u_y - \lambda k u_z & -\mu k u_y - \mu k^2 u_x & -\mu k u_z - \mu k u_x \\ -(2\mu + \lambda) k u_x & & \\ -\mu k u_y - \mu k^2 u_x & -\lambda k u_z - \lambda k u_x & -\mu k^2 u_z - \mu k u_y \\ -\mu k u_z - \mu k u_x & -(2\mu + \lambda) k^2 u_y & \\ -\mu k u_z - \mu k u_x & -\mu k^2 u_z - \mu k u_y & -\lambda k^2 u_y - \lambda k u_x \\ & & -(2\mu + \lambda) k u_z \end{pmatrix} \end{aligned} \quad (5.13)$$

con lo que, realizando el producto tensorial cerrado con la normal $\mathbf{n} = (0, 1, 0)$ de la superficie horizontal se puede obtener la fuerza \vec{f} equivalente:

$$\vec{f} = \sum_{\mathbf{x} \in S_t} \boldsymbol{\sigma}(\mathbf{x}, t) = \begin{pmatrix} -\mu k u_y - \mu k^2 u_x \\ -\lambda k u_z - \lambda k u_x - (2\mu + \lambda) k^2 u_y \\ -\mu k^2 u_z - \mu k u_y \end{pmatrix} \quad (5.14)$$

Asumiendo además que para herramientas grandes sucede que $k^2 \gg k$, entonces se pueden desestimar los términos en los que aparece k , quedando:

$$\vec{f} \approx \begin{pmatrix} -\mu k^2 u_x \\ -(2\mu + \lambda) k^2 u_y \\ -\mu k^2 u_z \end{pmatrix} \quad (5.15)$$

y siendo por tanto la dirección de los desplazamientos es aproximadamente la misma que la de la tensión, cambiando ligeramente la proporción de la componente paralela a la normal. La interpretación física de esta desviación es que es necesario ejercer *menos* fuerza en la componente paralela a la normal para conseguir el mismo desplazamiento, puesto que el factor multiplicador es mayor; o lo que es lo mismo, que es más *difícil* mover el material en las direcciones perpendiculares a la normal. Si se prescinde de este efecto, y teniendo en cuenta que lo anterior es en cualquier caso una aproximación al caso real, se podría reducir el cálculo a la expresión:

$$\vec{f} \approx C k^2 \vec{u} \quad (5.16)$$

con C una constante de proporcionalidad elástica, k^2 el área de la herramienta y \vec{u} el desplazamiento de cada partícula.

Con las simplificaciones que se han expuesto, y generalizando el caso que se ha desarrollado para el resto de orientaciones, puede asumirse una relación sencilla entre la dirección de los desplazamientos y la fuerza que los genera.

5.4 Resultados

Partiendo del campo de desplazamientos $\mathbf{u}_S^{\sim}(\mathbf{x}, t)$ de la superficie que se ha estimado y utilizando la aproximación que se ha descrito se puede realizar un cálculo rápido y eficiente de la fuerza de respuesta \vec{f}_h requerida por el dispositivo háptico. Teniendo en cuenta el modelo volumétrico que se propone en el [Capítulo 3](#), así como la definición de la herramienta, los pasos a seguir son sencillos.

1. **Cálculo de colisiones.** Durante el movimiento de la herramienta simulada por el entorno, se rastrea la posición \vec{c} del cursor que captura la posición del puntero háptico y se comprueba si existe material en

alguno de los vóxeles del volumen implícito de la herramienta esférica $T(\vec{c}, r)$ de radio r y centrada en \vec{c} . La condición de colisión es, por tanto:

$$\exists v_{ijk} \in T(\vec{c}, r) / d(v_{ijk}) \neq 0 \quad (5.17)$$

Ante esta comprobación:

- (a) Si es negativa, se hace $\vec{c}_0 \leftarrow \vec{c}$ para guardar siempre la última posición en la que no se detecta colisión. Se vuelve al paso 1.
- (b) Si es positiva, se salta al paso 2.

2. **Cálculo de los desplazamientos.** En caso de colisión, se marcan los vóxeles de la superficie del material (esto es, que tienen al menos un vecino vacío) que entran en contacto con la herramienta. Se aproxima su desplazamiento por:

$$\vec{u}^{\sim} = \vec{c} - \vec{c}_0 \quad (5.18)$$

tomando por tanto el vector *desplazamiento estimado* \vec{u}^{\sim} el valor de la distancia desde la posición actual \vec{c} del cursor a la última posición sin colisión \vec{c}_0 .

3. **Cálculo de la fuerza en respuesta.** Con los vóxeles desplazados se calcula la fuerza total equivalente según se indica en la ecuación 5.15, sustituyendo el área de la herramienta plana por el área de contacto de la herramienta esférica S_t :

$$\vec{f}^{\sim} = C S_t \vec{u}^{\sim} \quad (5.19)$$

obteniéndose así una expresión para la *fuerza estimada* de la deformación \vec{f}^{\sim} en la que la constante de proporcionalidad C se puede obtener de forma experimental.

Debe hacerse notar que en estos cálculos no se ha considerado la visualización de la deformación. Este algoritmo realiza exclusivamente el cálculo de la respuesta háptica, por lo que habrá que considerar, más adelante, el aspecto visual. Cabe también resaltar que si bien el cálculo de la fuerza que se ha propuesto corresponde al desarrollo del tramo elástico, la estimación

es sin embargo aplicable a todo el ciclo de carga y descarga, pues a la vista de la [Figura 4.6](#) se observa que en absolutamente *todos* los puntos se cumple la relación elástica entre tensión y deformación. La resolución de la carga elastoplástica tiene no obstante otras consecuencias en el modelo, las cuales se discutirán y resolverán en el [Capítulo 6](#).

El mecanismo propuesto resulta, en esencia, similar a las técnicas basadas en *proxy*, en el sentido de que también contempla un objeto virtual anclado en la superficie del modelo (esto es, la posición \vec{c}_0 que se usa para el cálculo). No obstante, el algoritmo del *proxy* se diseñó como una heurística intuitiva orientada a la exploración de la superficie de los modelos, mientras que la inspiración en el modelo mecánico que dirige el diseño de la técnica propuesta permitirá, como en los capítulos siguientes se muestra, implementar un modelo de deformación natural, creíble y eficiente.

5.5 Experimentos

Para medir el rendimiento y la adaptabilidad del método se han realizado pruebas de interacción con varios tamaños de herramienta sobre modelos de distintos tamaños, midiendo los tiempos en cada caso para comprobar la velocidad de actualización de la estimación de la fuerza. Las variables en las mediciones fueron:

Interacción Se tomaron tiempos en el estado de reposo (sin colisión) y con interacción, para medir las diferencias según la carga del algoritmo.

Tamaño del modelo Las medidas se realizaron con modelos de 128, 256, y 512 vóxeles al cubo.

Tamaño de la herramienta Las frecuencias analizadas contemplaban tamaños de herramienta que iban desde los 4 vóxeles de radio hasta los 32, en pasos de tamaño 2.

La primera conclusión a la que se llegó, a la vista de los datos obtenidos en bruto, era que no había diferencias significativas en la variación del tamaño del modelo, ya que los grupos de datos no presentaban un patrón claramente creciente o decreciente. El estado de reposo sí supone por contra

una mejora en la frecuencia de la estimación respecto del estado de interacción, pues es necesario realizar menos operaciones cuando no se detecta colisión de la herramienta. Sin embargo, no supone una variación de magnitud suficiente como para realizar un estudio independiente, por lo que las mediciones en reposo se han incluido en una única valoración conjunta. El [Cuadro 5.1](#) muestra los datos resultantes del experimento. Se indica, para cada tamaño de la herramienta, la frecuencia media de actualización de la fuerza de respuesta generada, que se ha medido sobre un conjunto de intervalos de 1s en los que se detectaba colisión el 50% del tiempo. Para cada población se calcula además su desviación típica, así como el porcentaje que la desviación supone sobre el promedio.

Cuadro 5.1 – Tabla con las frecuencias medias de actualización según el tamaño de herramienta, con la desviación típica de los datos recogidos en cada caso, y el porcentaje de que la desviación típica supone sobre la media.

Radio	Frecuencia (Hz)	Desviación	Variación (%)
4	53564	3437	6
6	15315	479	3
8	6348	183	3
10	3263	59	2
12	1912	50	3
14	1209	19	2
16	798	11	1
18	559	9	2
20	405	4	1
22	307	4	1
24	231	2	1
26	171	3	2
28	128	2	1
30	99	1	1
32	84	3	4

Los datos del [Cuadro 5.1](#) se representan gráficamente en la [Figura 5.7](#), donde además se ha calculado la curva de ajuste potencial negativa que marca la tendencia de las mediciones. La caída potencial en el rendimiento coincide con lo esperado, teniendo en cuenta que la operación más costosa

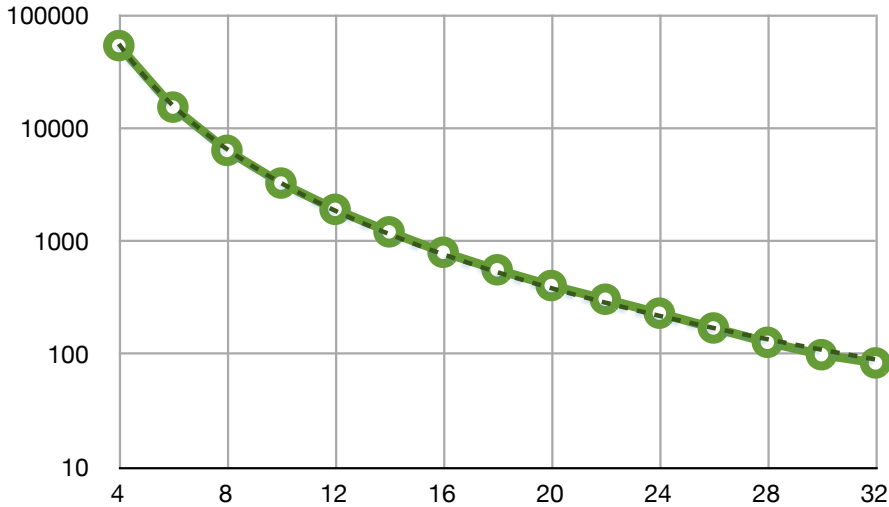


Figura 5.7 – Gráfica con los resultados de frecuencia según el tamaño de la herramienta, donde el radio en vóxeles se representa en el eje horizontal y la frecuencia de actualización en hercios en el eje vertical, con escala logarítmica. La regresión obtenida se muestra con una línea punteada, siendo esta una curva potencial de exponente $x^{-3.0888}$, mostrándose por tanto un decrecimiento potencial en el rendimiento conforme crece el radio y el volumen de la herramienta.

del algoritmo de estimación de la fuerza es la comprobación de la colisión de la herramienta con el volumen, que esta se detecta comprobando la cantidad de material en el interior de la herramienta esférica, y que el volumen crece de forma cúbica con el incremento del radio. Se aprecia a la vista de los datos obtenidos que la estimación de la fuerza cumple la velocidad de actualización mínima exigida por el dispositivo háptico para herramientas de tamaño medio. Una herramienta de radio 16 vóxeles permite una frecuencia de actualización del valor de \vec{f}_h cercana a $1kHz$, lo cual es un resultado aceptable teniendo en cuenta que permite trabajar en un área de 32 vóxeles de diámetro. Este tamaño supone la cuarta parte del lado en un volumen cúbico de lado 128 vóxeles y la octava parte en uno de lado 256 (ambos tamaños comunes en este tipo de representación volumétrica). Pasada la frontera del radio 18 el rendimiento baja de los $500Hz$ y, a pesar de que el método sigue siendo ágil, las frecuencias que se obtienen pueden dar lugar a una sensación táctil con saltos o discontinuidades.

5.6 Conclusiones

Al analizarse el marco real que sostiene a la simulación se concluyó que la teoría que relaciona la tensión y la deformación para materiales elásticos y plásticos ideales no es directamente aplicable, debido al vacío de información que no permite obtener ni derivar los parámetros que se usan para calcular la deformación formalmente.

Se han sugerido, a lo largo de este capítulo, una serie de simplificaciones y/o heurísticas que permiten reconstruir en parte el efecto original de deformar un sólido, siempre de forma aproximada, pero intentando en todo caso maximizar el realismo. No en vano, tras implementarse la estimación de la fuerza simplificada que se propone en la expresión 5.15 en el dispositivo háptico se consiguió una sensación táctil natural y robusta, exenta de vibraciones y con una evolución progresiva y suave. Si bien es cierto que son numerosas las simplificaciones realizadas, no es menos verdad que los distintos usuarios que han probado la aplicación que implementa el método de interacción propuesto aseguran percibir una sensación táctil parecida al trabajo real con una herramienta sobre un bloque de arcilla o plastilina, lo que, unido a que no se puede aspirar a la total fidelidad al modelo teórico, legitima en parte la elección del algoritmo de estimación de la fuerza.

Capítulo 6

Plasticidad

Tal y como se expuso detalladamente en el [Capítulo 4](#), dos son los tipos de efectos de deformación que pueden darse en un sólido elastoplástico: los temporales o elásticos y los permanentes o plásticos. La *teoría incremental de la plasticidad* los diferencia claramente, existiendo una transición clara y bien determinada entre los dos: el *límite elástico*.

En un problema de deformación con materiales reales se conocen las tensiones aplicadas y los parámetros que rigen el tránsito entre la zona elástica y plástica (representado formalmente por la *tensión de fluencia*). El cómputo de la deformación final se puede realizar formalmente con exactitud, planteando el sistema de ecuaciones diferenciales asociado. No en vano el ya mencionado trabajo de Prevost [[Pre84](#)] ofrece los resultados teóricos que se obtienen al resolver numéricamente el problema para varios tipos de materiales y herramientas. Definiendo correctamente la fuerza con la que presiona la herramienta, así como las constantes elásticas del material y las condiciones de contorno, se puede resolver de forma exacta el problema usando la formulación teórica del mismo.

No trabaja sin embargo la simulación aquí propuesta sobre este ideal marco de resolución del problema. La inclusión del dispositivo háptico conlleva una serie de restricciones que no permiten que se conozca directamente la fuerza que origina la deformación, siendo su valor en el mejor de los casos aproximado (ver discusión de la [Sección 5.2](#)).

Este punto de partida determina inevitablemente la recreación de la

deformación. Teniendo en cuenta el vacío de información inicial que no permite la resolución formal del problema, la configuración de las partículas del cuerpo una vez deformado será necesariamente estimada.

Es necesario por tanto diseñar un método heurístico que bordee esta y otras dificultades que aparecerán a la hora de simular la deformación sobre el modelo volumétrico propuesto. Con esta motivación se dividirán los esfuerzos, separando deformaciones permanentes y deformaciones temporales, de forma que, al aislar los efectos, se puedan abordar mejor los problemas. Este capítulo se ocupará del primero de los mencionados problemas: los cambios plásticos, los cuales alterarán la configuración espacial del volumen de forma permanente, moldeándose de forma definitiva los materiales simulados siguiendo los movimientos de la herramienta.

6.1 Aislamiento del efecto plástico

La tarea que se debe resolver consiste en registrar una posible deformación plástica en el modelo volumétrico voxelizado que se presentó en el [Capítulo 3](#), obviándose de cara a la visualización la parte recuperable del ciclo elastoplástico de carga-descarga. Entendida en este contexto la **deformación** es por tanto el proceso de cambio permanente de la estructura de datos que representa el material, de forma que tanto la superficie como el interior del sólido registran cambios.

En el marco de la teoría incremental de la plasticidad, estos cambios se producen una vez se ha generado una tensión suficiente como para abandonar la *zona de carga elástica* y entrar en la *zona de carga elastoplástica*, donde una parte de la energía se invierte en modificar permanentemente el sólido. Para traducir esta situación al dominio del volumen deben tenerse en consideración varias premisas:

- Existe una primera fase en la que se acumula una energía elástica que será recuperada cuando deje de aplicarse la presión de la herramienta.
- De no producirse una deformación permanente en el sentido antes expuesto, el sólido volverá al estado inicial sin importar cuánta energía elástica se haya acumulado.

- De producirse una deformación permanente, la distribución de valores de propiedad de los vóxeles en el entorno de la herramienta cambiará para reflejar los cambios producidos tanto en la superficie como en el interior del sólido.

Debe notarse que, si bien una acción que no supere el límite elástico y que por tanto no consiga entrar en la *zona de carga elastoplástica* no alterará en modo alguno los valores de propiedad del volumen, sí producirá en cambio una respuesta en fuerza acorde con lo dispuesto en el [Capítulo 5](#).

El algoritmo de deformación plástica debe considerar por tanto los cálculos ya realizados para calcular la fuerza que ya se han descrito, y usarlos como base para controlar la evolución de la deformación. En lo que sigue se formalizará el problema y sus parámetros, y se ofrecerá una solución para que la estructura volumétrica se vea alterada para reflejar la deformación permanente.

6.2 Viabilidad del modelo teórico

Tal y como se discutió cuando se realizó la estimación de la fuerza, la falta de información inherente a la configuración del entorno de simulación no permite resolver el problema de forma exacta. No obstante, y dado que se ha realizado una estimación de la fuerza con la que el usuario realiza la deformación, cabría la posibilidad de usarla como punto de partida para el cómputo de la solución formal.

Esta vía sin embargo plantea problemas adicionales. Para determinar el estado del sólido tras la deformación es necesario, como se ha comentado anteriormente, resolver las el sistema de ecuaciones en derivadas parciales en un modelo de elementos finitos, el cual bien podría coincidir con la estructura de datos volumétrica que representa el sólido en la simulación. Si bien el número de ecuaciones se puede reducir planteando la deformación como un problema cuasiestático en el cual el tiempo no es una variable, sigue existiendo un problema técnico en este enfoque, relacionado con el tiempo necesario para la resolución de un sistema de estas características.

Para cada cambio en las tensiones aplicadas (esto es, para cada deformación diferencial que se produzca en el sistema), es necesario resolver el

sistema de ecuaciones para todos los elementos del modelo. No obstante, podría sincronizarse el cálculo con la frecuencia de refresco de vídeo, la cual es sensiblemente menor que la del dispositivo háptico. Teniendo en cuenta que ésta debe actualizarse al menos a 25Hz para que se genere sensación de movimiento, sería necesario obtener una solución en un tiempo máximo de $1/25$ segundos, a saber, 40 milisegundos.

Es obvio que este intervalo de tiempo no es suficiente para resolver un sistema de ecuaciones en derivadas parciales de la magnitud vista en la [Sección 4.3.2](#), cuestión que puede extenderse durante horas [[GSL04](#)] si no se realizan algunas restricciones. En el trabajo de Mezger et al. [[MTPS08](#)], en el que se calculan las deformaciones de forma exacta usando los principios mecánicos expuestos, se usa de hecho un esquema de integración limitado en el tiempo con diversas optimizaciones que consiguen acotar una solución en los 40 milisegundos necesarios para conseguir interactividad. No obstante, para conseguir esta velocidad se acota el número máximo de nodos del modelo, limitándose en sus resultados el tamaño a 1500 tetraedros lineales.

Es por tanto evidente que, incluso con un método de integración limitada en el tiempo, es prácticamente inviable resolver este problema para un modelo que, en el mejor de los casos, tendrá algo más de dos millones de elementos (para un volumen cúbico de 128 vóxeles de lado se tienen $128^3 = 2^{21} \approx 2 \cdot 10^6$ vóxeles). Sería necesario en última instancia realizar los cálculos sobre una estructura asociada que agrupara los vóxeles en nodos de mayor tamaño para posteriormente realizar una interpolación, para trasladar los resultados de la deformación al modelo volumétrico.

En cualquier caso, cualquier resolución formal del problema tendría que asumir como condición inicial que la fuerza que se estima según el algoritmo explicado en la [Sección 5.4](#) es la que origina la deformación. Un cálculo exacto de la distribución de material realizado con dicho valor como parámetro no tiene por qué ser coherente con la premisa que se usa para estimar la fuerza: la forma de la superficie en la zona de contacto con la herramienta y su campo de desplazamiento estimado $\mathbf{u}_{\tilde{S}}(\mathbf{x}, t)$. Es posible que si la fuerza está sobreestimada la deformación teórica calculada fuera mayor, y que si se subestima la superficie no se deformara lo suficiente para seguir el movimiento de la herramienta. Esta posible descoordinación entre el movimiento

de la herramienta y la superficie tendría que ser solucionado igualmente, de modo que el usuario no percibiese una evolución disfuncional durante la interacción.

6.3 Aproximación heurística de la deformación plástica

Dadas las dificultades que plantea una resolución exacta del problema, cabe la posibilidad de diseñar una aproximación que obtenga una solución aceptable, intentando siempre conseguir la máxima naturalidad y/o fidelidad al proceso real.

La situación es poco más ventajosa que en el momento de estimar la fuerza de respuesta, puesto que:

- no se conoce ni se tiene estimación del campo de tensiones del material $\boldsymbol{\sigma}(\mathbf{x}, t)$;
- no se conoce ni se tiene estimación del campo de desplazamientos del material $\mathbf{u}(\mathbf{x}, t)$, salvo por el campo estimado de desplazamientos de la superficie $\mathbf{u}_{\tilde{G}}(\mathbf{x}, t)$ que se definió para poder aproximar la fuerza de respuesta, y que obviamente no es suficiente para conocer la deformación global.

El punto de partida es además un estado artificial del sólido en el cual la herramienta virtual *atraviesa* la superficie sin que se aprecie ninguna deformación, pues existe un tramo elástico recuperable que en este desarrollo no se considera. Este efecto lo enmascara parcialmente el hecho de que el usuario no percibe lo que sucede en el interior de la herramienta, precisamente porque ésta lo oculta al dibujarse superpuesta al modelo. No es por tanto este una situación especialmente dramática siempre y cuando la herramienta no llegue a ocultarse completamente sin que se modifique el material, situación que resultaría anormal y que debe evitarse (véase descripción de la situación en la [Figura 6.1](#)).

La deformación plástica debe en todo caso iniciarse cuando la tensión acumulada supera el límite elástico del material marcado por la tensión de

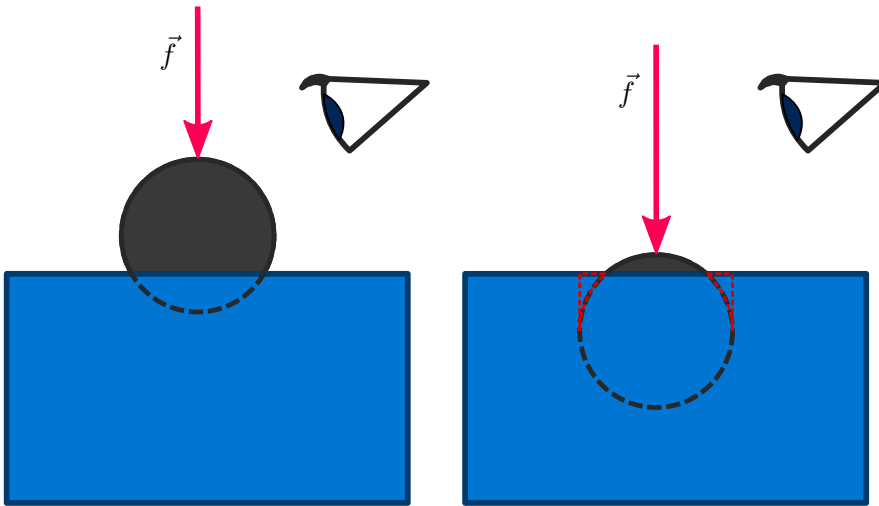


Figura 6.1 – Ilustración bidimensional de la oclusión de la herramienta en distintas etapas de la interacción. A la izquierda, la herramienta interseca con el material sin que el usuario perciba ninguna anomalía incluso aunque no se efectúe ningún cambio en el mismo. A la derecha, el usuario espera ver la transformación en el material causada por el empuje de la herramienta, por lo que es necesario, antes de llegar a este punto, realizar cambios en la distribución del material. En rojo punteado se muestra el espacio que debería estar vacío, fruto del *tunnel* escavado en el material.

fluencia $\sigma_f(\alpha)$, cuyo valor inicial es el umbral elástico inicial característico del material, σ_e , y que va incrementándose con el endurecimiento (ecuación 4.40). Una vez superado el límite elástico y ante un incremento en la presión de la herramienta se producirán una serie de desplazamientos del material que permanecerán después de recuperarse la parte elástica de la deformación.

A grandes rasgos, tres son las tareas que deben abordarse en el diseño de este algoritmo:

1. La primera consiste en cómo interpretar en dominio del volumen V los desplazamientos de partículas que tienen lugar en el proceso real, teniendo en cuenta que el modelo está compuesto de un conjunto discreto de vóxeles v_{ijk} con distintos valores de propiedad $d(v_{ijk})$, representando la densidad de material en cada una de las zonas aso-

ciadas a los vóxeles, y que los procesos de deformación se describen en términos de un campo de desplazamientos de las partículas del material.

2. La segunda es adaptar la ecuación constitutiva al modelo virtual, caracterizando para ello la función de fluencia $\sigma_f(\alpha)$ para poder decidir cuándo se abandona la zona elástica y se debe por tanto empezar a aplicar desplazamientos permanentes sobre las partículas del sólido simulado.
3. La tercera es precisamente determinar la localización, magnitud y dirección de los desplazamientos permanentes que aparecen cuando se entra en la zona de carga plástica. Por todo lo expuesto, la distribución será necesariamente heurística, si bien se puede inspirar la estimación en los resultados formales del estudio de Prevost.

En lo que sigue se propondrá un procedimiento de estimación de la deformación plástica que resuelva estas cuestiones, ajustándose a las premisas y requerimientos que se han especificado.

6.3.1 Traducción de los desplazamientos al dominio del volumen

Es fácil comprobar que los movimientos de las partículas no son directamente representables en un modelo voxelizado, en el cual los elementos tienen una posición fija en el espacio.

No obstante, y sabiendo que cada vóxel posee un valor de propiedad que se puede interpretar como el *nivel de ocupación* de material en dicho espacio, se puede considerar que, con el empuje adecuado, el material se *desplaza* de forma macroscópica de un vóxel a otro vecino, realizándose una **transferencia** efectiva de partículas.

Esta idea en la que se asocian vectores de desplazamiento con la redistribución de los valores de propiedad de los vóxeles afectados por los mismos ya se ha probado anteriormente en el trabajo de Dewaele y Cani [DC04b], el cual se describió en el [Capítulo 2](#). Adicionalmente, la idea del trasvase de valor de propiedad se ha usado en otras ocasiones, tal y como se

mostró en la revisión de los trabajos de Arata et al. [ATTY99] y Sakamoto et al. [STK99]. No obstante, en lo que sigue se realizará una equivalencia formal que no aparece en los trabajos anteriores, y que sentará la base para poder abordar la adaptación de la teoría de la plasticidad incremental al modelo volumétrico, objetivo último de esta exposición.

6.3.1.1 Equivalencia de los desplazamientos

Sea \vec{u} un desplazamiento del campo que incide en el espacio ocupado por un voxel v_{ijk} . Para efectuar pues el desplazamiento, se disminuirá en una cantidad w la densidad de v_{ijk} , coincidiendo pues su nuevo valor de propiedad con $d(v_{ijk}) - w$. El material w que abandona dicho voxel deberá ser repartido entre los voxéles vecinos de v_{ijk} , según la dirección que indique \vec{u} . Un ejemplo se ilustra en la Figura 6.2, en la que una herramienta plana presiona sobre la superficie de un modelo en la dirección paralela a la normal.

Un vector de desplazamiento \vec{u} aplicado sobre el centro de un voxel v_{ijk} transferirá *todo* el material a otro voxel v_{lmn} sólo en el caso de que \vec{u} sea:

$$\vec{u} = x(v_{ijk}) - x(v_{lmn}) \quad (6.1)$$

esto es, que cubra exactamente la distancia que hay desde el centro de v_{ijk} hasta el centro de v_{lmn} . Con esta condición, las densidades de los dos voxéles:

$$\begin{aligned} d(v_{ijk}) &\leftarrow 0 \\ d(v_{lmn}) &\leftarrow d(v_{ijk}) + d(v_{lmn}) \end{aligned} \quad (6.2)$$

cambian de forma que v_{ijk} se *vacía*, desplazándose todo el material que contiene a v_{lmn} . Si no se cumple esta condición, el material de v_{ijk} se divide proporcionalmente entre los voxéles inmediatamente más próximos al punto $\vec{u} + x(v_{ijk})$, según la distancia de dicho punto hasta sus centros.

Formalizando lo anterior, se definirá el conjunto de voxéles sobre el que se hace la distribución mediante la función χ :

$$\chi : V \times \mathbb{R}^3 \rightarrow V \quad (6.3)$$

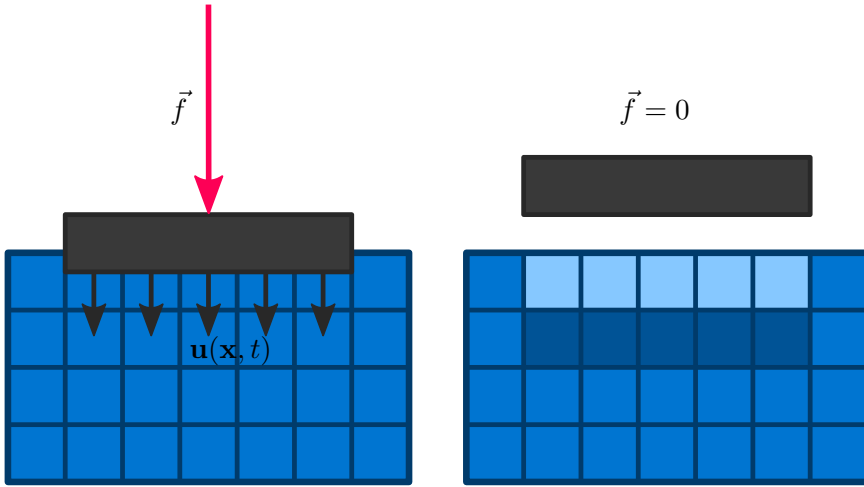


Figura 6.2 – Traducción de un campo de desplazamientos $\mathbf{u}(\mathbf{x}, t)$ en un cambio en la distribución de valores de propiedad del entorno de la herramienta, simplificado en dos dimensiones. A la izquierda, interacción de la herramienta en la que aparecen los desplazamientos. A la derecha, la herramienta retrocede y se muestra el estado del sólido tras la deformación: los vóxeles sobre los que incidió el campo \mathbf{u} se han *vaciado* parcialmente (por lo que se muestran en azul más claro), para transferir su contenido a los vecinos inmediatamente inferiores (vóxeles oscurecidos), que ven incrementado su valor de propiedad.

que para cada vóxel y su desplazamiento asociado determina los 2^3 vóxeles inmediatamente más cercanos al punto indicado. La cantidad de material que recibirá cada voxel $v_j \in \chi(v_i, \vec{u}_i)$ del voxel v_i con el desplazamiento \vec{u}_i vendrá determinada por la fórmula:

$$\Delta d(v_j) = \frac{\|x(v_j) - (x(v_i) + \vec{u}_i)\|}{\sum_{k \in \chi(v_i, \vec{u}_i)} \|x(v_k) - (x(v_i) + \vec{u}_i)\|} d(v_i) \quad (6.4)$$

esto es, la proporción entre la distancia de su centro al punto distribución respecto de la suma de todas las distancias de los vóxeles de χ .

Esto se ilustra para un modelo simplificado en dos dimensiones en la [Figura 6.3](#), en la que el entorno de distribución de cada punto $\vec{u} + x(v_i)$ son $2^2 = 4$ vóxeles en dos dimensiones. El vector de desplazamiento \vec{u}_{c2} lleva todo el material del vóxel v_{c2} al entorno $\chi(v_{c2}, \vec{u}_{c2})$, a saber: v_{b3} , v_{b4} , v_{c3} y v_{c4} , marcados en la imagen con un cuadrado punteado, y donde se

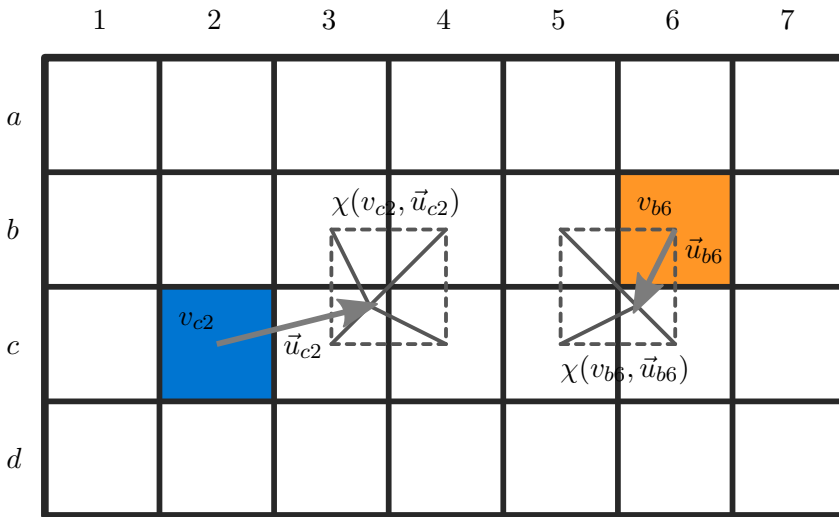


Figura 6.3 – Transferencia de material entre vóxeles según el vector de desplazamiento. El vóxel v_{c2} sufre un desplazamiento de material determinado por el vector \vec{u}_{c2} , mientras que el material de v_{b6} se traslada según \vec{u}_{b6} . En la figura se marcan los entornos de distribución $\chi(v_{c2}, \vec{u}_{c2})$ y $\chi(v_{b6}, \vec{u}_{b6})$ respectivos con una línea gris punteada, así como las distancias relativas de cada vóxel del entorno al centro de distribución.

pueden apreciar también las distancias relativas al punto de distribución $x(v_{c2}) + \vec{u}_{c2}$. El caso del vóxel v_{b6} es similar, excepto por el hecho del que el propio vóxel está en el entorno de distribución $\chi(v_{b6}, \vec{u}_{b6})$. En este caso v_{b6} no se vacía completamente, pues se queda la parte proporcional de su propio material que le tocaría en la distribución realizada según la expresión 6.4.

Un caso especial aparece cuando se usa un desplazamiento como el descrito en la ecuación 6.2. En dicho caso la función χ devolvería un entorno de distribución compuesto de un único vóxel, cumpliéndose así la condición de trasvase completo tal y como ha definido anteriormente.

Se puede comprobar que con esta definición de transferencia existe una relación unívoca entre un desplazamiento \vec{u} aplicado sobre el centro de un vóxel y su correspondiente distribución en un entorno de hasta 8 vóxeles adyacentes, lo cual permitirá más adelante hablar de movimientos de material en el volumen o de desplazamientos de forma indiferente y con una equivalencia siempre exacta.

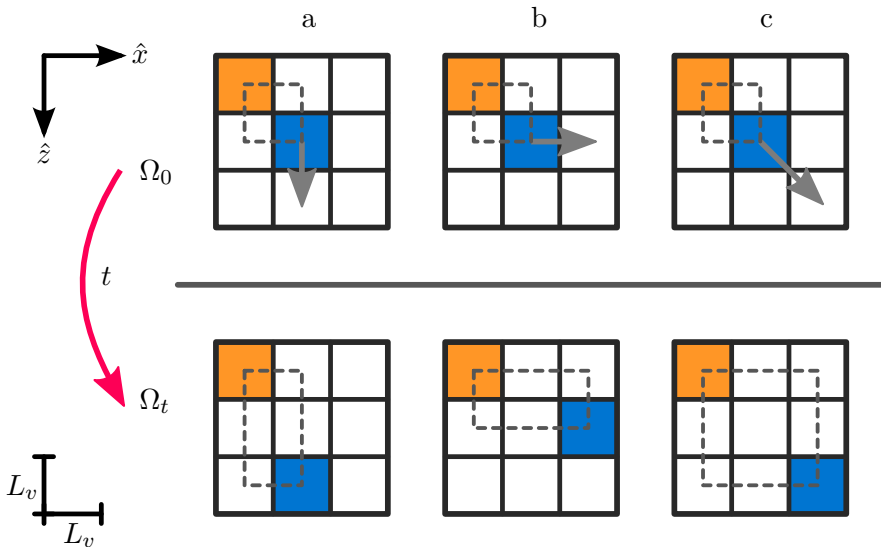


Figura 6.4 – Movimiento de una partícula-vóxel según distintos vectores de desplazamiento. La partícula azul sobre la que se aplica el desplazamiento se mueve respecto de la partícula de referencia (naranja). La distancia entre las partículas se marca con la línea punteada en las dos configuraciones inicial Ω_0 y final Ω_t .

6.3.1.2 Equivalencia de las deformaciones

Se puede además calcular las *deformaciones* asociadas si se estudia el efecto que el desplazamiento \vec{u} tiene sobre un único vóxel, y se considera cada vóxel como una partícula para los efectos de esta exposición. Supónganse la simplificación 2D que se muestra en la [Figura 6.4](#), en la que un vóxel sobre el que se aplican tres desplazamientos distintos: uno en la dirección del eje \hat{x} , otro en la dirección del eje \hat{y} , y otro paralelo a la bisectriz de los dos ejes. Si se considera como partícula de referencia el vóxel vecino (de color naranja en la figura), se comprueba fácilmente que la longitud de los segmentos L_x y L_y que los separan es igual al lado del vóxel, L_v .

La deformación ε_x y ε_y se puede formalizar para cada uno de los casos, sin más que aplicar la definición de *alargamiento unitario* (ver [Sección 4.1](#)) en cada uno de los ejes, siendo esta la relación entre el incremento de longitud de los segmentos entre partículas y su magnitud original. Si (L_x, L_y) es la distancia en la configuración inicial Ω_0 que separa una partícula que se sufre un desplazamiento (u_x, u_y) , y (L'_x, L'_y) es la distancia que las separa

Cuadro 6.1 – Desplazamientos y deformaciones de las partículas de la [Figura 6.4](#).

	L_x	L_y	L'_x	L'_y	u_x	u_y	ε_x	ε_y
a	L_v	L_v	$2L_v$	L_v	L_v	0	$\frac{L_v}{L_v}$	0
b	L_v	L_v	L_v	$2L_v$	0	L_v	0	$\frac{L_v}{L_v}$
c	L_v	L_v	$2L_v$	$2L_v$	L_v	L_v	$\frac{L_v}{L_v}$	$\frac{L_v}{L_v}$

en la configuración deformada Ω_t , entonces se tiene que:

$$\begin{aligned}\varepsilon_x &= \frac{L'_x - L_x}{L_x} \\ \varepsilon_y &= \frac{L'_y - L_y}{L_y}\end{aligned}\tag{6.5}$$

es la descomposición de la deformación en las direcciones \hat{x} y \hat{y} . Se pueden comprobar las deformaciones obtenidas para cada uno de los casos mostrados en la [Figura 6.4](#) en el [Cuadro 6.1](#).

6.3.2 Caracterización de la zona elástica y la zona plástica

La información más crucial que se requiere para poder aplicar las nociones de *carga elástica* y *carga elastoplástica* que se expusieron en el [Capítulo 4](#) es el límite entre la zona elástica y la zona plástica. Es necesario determinar en qué momento la tensión y la deformación dejan de guardar la proporción de la zona elástica y se empieza a invertir energía en deformar el modelo, máxime teniendo en cuenta que en este contexto las deformaciones recuperables no producirán efecto visual alguno.

Formalmente se trata de definir la tensión de fluencia $\sigma_f(\alpha)$ y la variable de endurecimiento α asociada que guarda el historial de la deformación. Al no disponerse de una medida del campo de tensiones $\boldsymbol{\sigma}(\mathbf{x}, t)$ del cuerpo, no es por tanto directamente aplicable la definición [4.41](#), en la que se construye la *función de fluencia plástica* que permite decidir la entrada en la zona de carga plástica. En el mejor de los casos, se podría aplicar el criterio de Von Mises (ecuación [4.48](#)) a las tensiones que se obtienen del

campo $\mathbf{u}_{\tilde{S}}(\mathbf{x}, t)$ del que se dispone. Al ser en cualquier caso una aproximación, y teniendo en cuenta que se ha considerado que en el entorno de la herramienta fuerza y desplazamientos tienen aproximadamente la misma dirección, bien se podría definir el cálculo en términos del modelo lineal de la teoría, lo que simplificaría considerablemente los cálculos y el tiempo de computación necesario para realizarlos.

6.3.2.1 Función de fluencia y endurecimiento

Sea por tanto $f^{\sim} = \|\vec{f}^{\sim}\|$ el módulo de la fuerza estimada y $u^{\sim} = \|\vec{u}^{\sim}\|$ el módulo del desplazamiento usado para su estimación, según lo visto en la [Sección 5.4](#). Sabiendo que la tensión es fuerza por unidad de área, se define la tensión estimada σ^{\sim} como:

$$\sigma^{\sim} = \frac{f^{\sim}}{S_t} \quad (6.6)$$

donde S_t es el área de contacto de la herramienta usada para el cálculo de \vec{f}^{\sim} (ver [Sección 5.4](#)). Sustituyendo la ecuación [5.19](#) se tiene que la tensión estimada por vóxel es:

$$\sigma^{\sim} = C u^{\sim} \quad (6.7)$$

y sabiendo que además el desplazamiento es la proporción entre el estiramiento conseguido y la longitud del segmento deformado (la cual, durante el trabajo con el modelo unidimensional, se entenderá como constante para cualquier dirección y tomará el valor del lado del vóxel L_v) se tiene que:

$$\varepsilon^{\sim} = \frac{u^{\sim}}{L_v} \quad (6.8)$$

por lo que, sustituyendo en [6.6](#):

$$\sigma^{\sim} = C L_v \varepsilon^{\sim} \quad (6.9)$$

aparece la relación lineal que guardan σ y ε y que *siempre* se cumple, ya esté el sólido en régimen de carga elástica, carga plástica o de descarga.

Para sortear la definición del ignoto tensor $\boldsymbol{\sigma}$ se puede por tanto usar lo anterior para calcular el momento en el que se supera la tensión elásti-

ca máxima, utilizando las estimaciones definidas en la función de fluencia descrita en el modelo unidimensional de la teoría (ecuación 4.31):

$$F^{\sim}(\sigma^{\sim}, \alpha^{\sim}) = \sigma^{\sim} - \sigma_f(\alpha^{\sim}) \quad (6.10)$$

obteniéndose una *función de fluencia plástica estimada* $F^{\sim}(\sigma^{\sim}, \alpha^{\sim})$, en la cual se ha usado la definición de la tensión aproximada de la ecuación 6.6, y que requiere igualmente una referencia para la tensión de fluencia σ_f y la variable de endurecimiento α .

La tensión de fluencia se define formalmente en 4.30 como $\sigma_f(\alpha) = \sigma_e + H'\alpha$. Para α puede igualmente tomarse la definición que se hace para el caso uniaxial en la que se equipara con valor absoluto de la deformación plástica acumulada ε_p :

$$\alpha^{\sim} = |\varepsilon_p^{\sim}| \quad (6.11)$$

en la que la *variable de endurecimiento estimada* α^{\sim} es igual al valor absoluto de la *deformación plástica estimada* ε_p^{\sim} , esto es, la parte de la deformación estimada ε^{\sim} que produce cambios permanentes en el sólido.

Con todo lo anterior puede entonces decidirse el fin del comportamiento elástico en términos de la fuerza estimada \vec{f}^{\sim} , que es el dato del que se dispone. La teoría incremental de la plasticidad dicta que el sólido virtual permanecerá en la zona elástica hasta que:

$$F^{\sim}(\sigma^{\sim}, \alpha^{\sim}) = 0 \quad (6.12)$$

y, combinando las expresiones 6.6, 6.10, 6.11 y la definición de la tensión de fluencia de 4.30 queda:

$$\begin{aligned} \sigma^{\sim} &= \sigma_e + H'\alpha^{\sim} \\ \frac{f^{\sim}}{S_t} &= \sigma_e + H'|\varepsilon_p^{\sim}| \end{aligned} \quad (6.13)$$

donde los valores del límite elástico inicial σ_e y del parámetro de endurecimiento H' deberán ser obtenidos de forma experimental mediante comparación con un material real.

6.3.2.2 Resolución de la carga plástica

La primera vez que se alcanza el límite elástico en una interacción (esto es, cuando $\alpha^{\sim} = 0$) la condición es sencilla, pues según lo escrito en 6.13 la condición sería que $\sigma^{\sim} = \sigma_e$. Una vez superado este valor, cualquier incremento en σ^{\sim} generaría una deformación permanente proporcional en la forma:

$$d\sigma = C_{ep}L_v d\varepsilon \quad (6.14)$$

que no es más que la ecuación constitutiva para el régimen elastopástico en carga plástica (ecuación 4.35), cambiando la proporción original por el producto de una *constante de proporcionalidad plástica* C_{ep} y la longitud de un vóxel L_v . Esto permite, por la definición de deformación que se aplicó en la ecuación 6.8, describir el incremento en la tensión en términos del incremento del desplazamiento u :

$$d\sigma = C_{ep}du \quad (6.15)$$

de la cual además se puede conocer la parte permanente, por la descomposición aditiva de la deformación de 4.27, según la cual $d\varepsilon = d\varepsilon_e + d\varepsilon_p$, la relación entre tensión y deformación para el tramo elástico $d\sigma = CL_v d\varepsilon_e$ (ecuación 6.9), y sabiendo que en esta carga plástica se cumple $d\sigma = C_{ep}L_v d\varepsilon$ (ecuación 6.14):

$$d\varepsilon_p = \frac{d\sigma}{C_{ep}L_v} - \frac{d\sigma}{CL_v} \quad (6.16)$$

y que se puede corroborar con el análisis trigonométrico que aparece en la Figura 6.5, pudiéndose además derivar fácilmente de esta expresión el desplazamiento plástico asociado $du_p = L_v d\varepsilon_p$ por la relación constante entre ε y u . Este incremento en la tensión $d\sigma$ una vez que se ha alcanzado el límite elástico inicial $\sigma^{\sim} = \sigma_e$ y su deformación plástica asociada $d\varepsilon$ tienen los siguientes efectos:

- La tensión σ^{\sim} se ve obviamente incrementada, de forma que se actualiza su valor con:

$$\sigma^{\sim} \leftarrow \sigma^{\sim} + d\sigma \quad (6.17)$$

- El desplazamiento creado tiene una parte permanente, la cual se al-

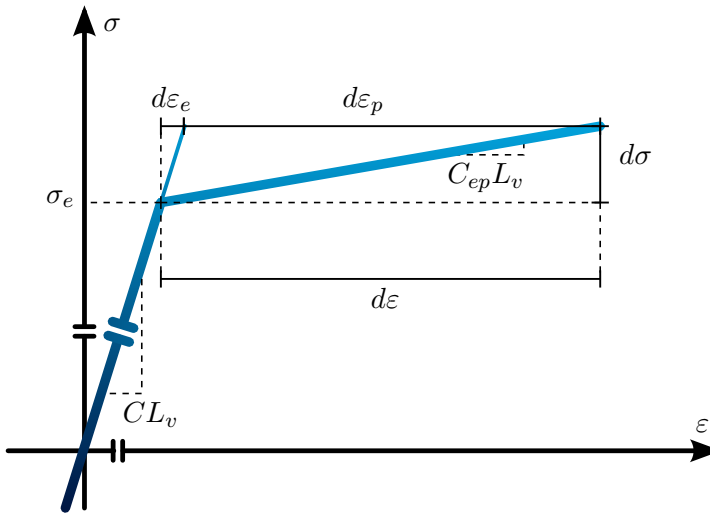


Figura 6.5 – Representación de la relación tensión-deformación ante un incremento de la tensión cuando $\sigma = \sigma_f(0) = \sigma_e$. Se observa que, conocidas las pendientes del tramo elástico (CL_v) y el elastoplástico ($C_{ep}L_v$) se puede calcular por semejanza de triángulos la parte elástica $d\varepsilon_e$ de $d\varepsilon$ que se podrá recuperar y la parte plástica $d\varepsilon_p$ que quedará registrada de forma permanente en el sólido cuando se retire la tensión aplicada.

macena en la variable de endurecimiento estimada:

$$\alpha^{\sim} \leftarrow \alpha^{\sim} + |d\varepsilon_{ep}| \tag{6.18}$$

- La tensión de fluencia cambia automáticamente con α^{\sim} , ya que por definición $\sigma_f(\alpha^{\sim}) = \sigma_e + H'\alpha^{\sim}$.

Tras lo cual se comprueba fácilmente que de nuevo $\sigma^{\sim} = \sigma_f(\alpha^{\sim})$ y hay que esperar a la siguiente variación de u para decidir el estado del sólido:

- Si el incremento vuelve a ser positivo, se aplica de nuevo el procedimiento de carga plástica que se ha descrito, basándose en el incremento de la tensión calculado con la ecuación 6.15.
- Si el incremento es negativo, se vuelve al comportamiento elástico que se rige por la ecuación 6.7, y se comprueba fácilmente que se cumple la desigualdad que caracteriza al estado elástico, $\sigma^{\sim} < \sigma_f(\alpha^{\sim})$.

En cualquier momento y en cualquiera de los estados se puede obtener el valor de la fuerza estimada f^\sim sin más que aplicar la definición de la tensión estimada σ^\sim que se proporcionó en 6.6.

Bajo la premisa de que en el entorno próximo a la herramienta los cambios de dirección entre desplazamiento, deformación, tensión y fuerza son prescindibles, se ha construido el procedimiento de decisión del estado del sólido utilizando exclusivamente el modelo unidimensional de la teoría. Es evidente que esta decisión obvia por completo cualquier rotación que pudiera darse en el modelo tridimensional, si bien este hecho no tiene por qué ser determinante de cara al efecto final de la simulación. Como se verá en lo que sigue, estos cálculos repercuten exclusivamente en los desplazamientos adyacentes a la herramienta, quedando aún pendiente definir qué efecto tendrá todo lo anterior en el interior del sólido.

En cualquier caso deberán estudiarse las consecuencias de dicha decisión una vez implementado el procedimiento descrito, pudiendo no obstante esperarse que los cálculos lineales de la fuerza y la tensión resulten una aproximación lo suficiente significativa, quizás no cuantitativa pero sí cualitativamente.

6.3.3 Estimación del campo de desplazamientos

Para completar el algoritmo de deformación plástica queda aún resolver la tarea de encontrar una nueva distribución en la configuración de valores de propiedad $d(v_{ijk})$ de los vóxeles de un volumen V sometido a una carga plástica, la cual refleje el estado de un sólido elastoplástico sometido a una deformación similar.

La premisa de partida son los desplazamientos estimados $\mathbf{u}_G^\sim(\mathbf{x}, t)$ de la superficie del sólido respecto del estado de reposo (ver Figura 5.5), los cuales son además usados para la estimación de la fuerza \vec{f} que causa la deformación, tal y como se explicó en las secciones 5.3.1 y 5.3.2. Además, tras lo expuesto en la sección anterior, se conoce el momento en el que comienza la carga elastoplástica, marcado por la función de fluencia plástica estimada $F^\sim(\sigma^\sim, \alpha^\sim)$.

Formalmente, y dado que no es viable el cálculo de la solución exacta, habrá que definir el campo de desplazamientos plásticos internos del

material, que aparecerá siempre que suceda una carga plástica en el sistema. Para ello se deberá analizar primero qué sucede en la superficie del sólido, puesto que el movimiento de la herramienta es la base de todos los cálculos que se realizan en la simulación, para, a continuación, decidir cómo evoluciona el interior del material con cada cambio permanente.

6.3.3.1 Análisis de los cambios en la superficie

En el procedimiento propuesto para la estimación de la fuerza \vec{f}^{\sim} que produce la deformación (Sección 5.4) se hizo alusión a una posición especial de la herramienta, \vec{c}_0 , que representaba la localización de la herramienta *justo un ciclo antes* de detectarse una colisión. Esta posición es de especial relevancia en el proceso que aquí se describirá, pues es el punto al que debe volver la herramienta háptica para que anule la tensión y por tanto se termine un ciclo completo de *carga elástica* \rightarrow *carga elastoplástica* \rightarrow *descarga*.

Se realizará en lo que sigue un análisis sobre el modelo de lo que sucede cuando la herramienta llega a una posición \vec{c} tal que se llega al fin del comportamiento elástico, para localizar las traslaciones de material que suceden en la superficie del sólido.

Sea por tanto \vec{c}_f la posición que hace que el desplazamiento estimado \vec{u}^{\sim} (y por tanto la tensión estimada σ^{\sim}) sea tal que $F(\sigma^{\sim}, \alpha^{\sim}) = 0$. Tal y como se explicó en la sección anterior, si la siguiente posición registrada causa un incremento $d\sigma$ en la estimación de la tensión σ^{\sim} se crea una deformación permanente, la cual se ha asumido tiene la misma dirección que el desplazamiento \vec{u} . Este análisis unidimensional que se realizó tiene no obstante una consecuencia directa en la superficie del material, y es el *vaciado* parcial que se realiza en la zona que debería ocupar la herramienta una vez aplicados las deformaciones permanentes.

Los cálculos de la nueva superficie son, con todo lo anterior, relativamente sencillos. Una vez se conoce el desplazamiento permanente du_{ep} asociado a la deformación permanente $d\varepsilon_{ep}$ que se calcula según la ecuación 6.16, se sabe por tanto en qué magnitud se deforma la superficie según la simplificación propuesta: estará en contacto con la herramienta cuando esta esté en $\vec{c}_0 + du_p \hat{u}$, que es precisamente la posición a la que se debe retornar para

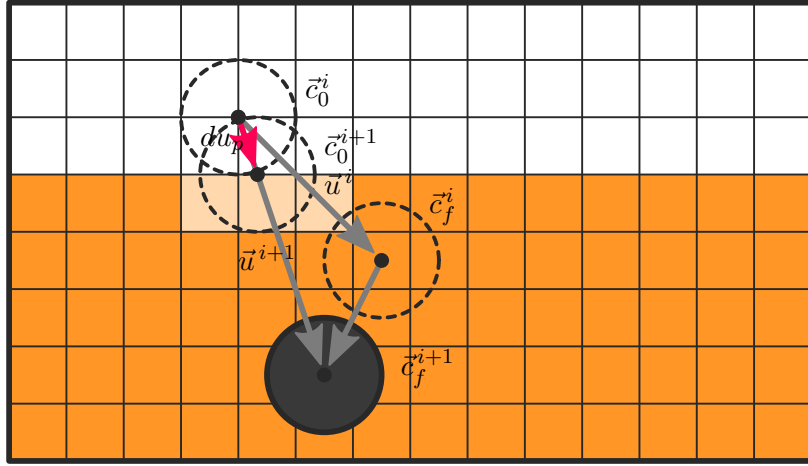


Figura 6.6 – Cálculo de la deformación plástica una vez que se ha sobrepasado el límite elástico. Si \vec{c}_0^i es la última posición en la que no se detectó colisión, y \vec{c}_f^i es la posición en la que se cumple $F(\sigma^\sim, \alpha^\sim) = 0$, entonces en el siguiente movimiento de la herramienta a \vec{c}_f^{i+1} causa la actualización de \vec{c}_0^i a \vec{c}_0^{i+1} , siendo esta última la posición de la herramienta en la que no hay colisión dada la configuración del sólido ya deformado. La distancia de \vec{c}_0^i a \vec{c}_0^{i+1} es precisamente la parte plástica de la deformación u_p .

deshacer la deformación elástica. Puede observarse además que, al acortarse el vector $\vec{u}^\sim = \vec{c}_f - \vec{c}_0$ justo en la parte plástica (du_p), la distancia con la que se estima la fuerza aumenta justo en la parte elástica que se consigue con este incremento en la tensión $d\sigma$, lo cual es coherente con la adaptación del modelo teórico que se ha propuesto.

Esto se puede comprobar en la simplificación bidimensional que se muestra en la [Figura 6.6](#). Inicialmente la herramienta se encuentra en la posición \vec{c}_f^i , la cual, junto con la última posición en la que no hubo colisión \vec{c}_0^i , producen el vector de estimación para la fuerza \vec{u}^i que hace que se llegue al límite elástico. En el siguiente ciclo la herramienta se mueve hasta \vec{c}_f^{i+1} , produciendo por tanto un incremento en la tensión estimada según la ecuación constitutiva para la carga elastoplástica, $d\sigma = C_{ep}du$ (ecuación 6.15), donde du es la diferencia en módulo de los dos vectores de desplazamiento $du = u^{i+1} - u^i$. Con este nuevo valor máximo para σ^\sim se incrementa

automáticamente la tensión de fluencia, y además se genera un tramo de deformación plástica y uno elástico según lo discutido anteriormente. Se puede observar que al *mover* la posición virtual de la herramienta en la superficie \vec{c}_0^{i+1} justo al punto $\vec{c}_0^i + du_p$, el vector para la estimación de la fuerza \vec{u}^{i+1} cumple de nuevo la proporción elástica con el nuevo valor de σ^\sim , cumpliéndose además el criterio de endurecimiento.

Con este procedimiento se puede conocer por tanto el material que se desplaza por contacto directo con la herramienta, pudiendo localizarse consecuentemente los vóxeles que la herramienta desaloja con cada deformación plástica. Este hecho básico (no puede quedar material en la zona por la que pasa la herramienta en una deformación plástica) se usará como premisa para la estimación de las deformaciones internas que se hará a continuación.

6.3.3.2 Obtención de la superficie deformada

Tal y como se ha descrito, el material de la superficie se ve directamente desplazado por el empuje de la herramienta, por lo que los vóxeles que contactan con la misma sufrirán la deformación plástica $d\varepsilon_p$ que se ha calculado. Teniendo en cuenta que el estudio de la resolución de la carga plástica de la [Sección 6.3.2](#) se hizo considerando el comportamiento de un único vóxel, se aplicará un desplazamiento du_p en la dirección del desplazamiento estimado \vec{u}^\sim para cada uno de los vóxeles desalojados por la herramienta.

La cantidad de material desalojado y su destino vendrán determinadas por la equivalencia entre desplazamiento y traslación de material que se definió en la [Sección 6.3.1](#), de forma que la superficie se va *modelando* con el avance de la herramienta en cada nuevo incremento de la deformación plástica.

6.3.3.3 Primera aproximación de la deformación interna

Con todo lo evidente que resulta la idea de que las zonas por las que pasa la herramienta en una deformación permanente deben desalojarse, no está claro hacia dónde debe desplazarse el material, más allá del que se mueve por efecto directo de la herramienta.

El problema de deformar el sólido puede abordarse de dos formas:

1. Aplicando un campo de desplazamientos de forma global a todo el modelo. Este enfoque es el que debería acometerse si se dispusiera de la solución exacta, y es de hecho el método que se usa en el trabajo de Mezger et al. [MTPS08] para ejecutar la deformación de la estructura. La tarea se centraría por tanto en diseñar los movimientos de material y aplicarlos en todo el volumen a la vez con cada deformación plástica $d\varepsilon_p$.
2. Creando un método progresivo de desplazamiento en cadena. De forma a similar a lo realizado en otras heurísticas que se analizaron, tales como la de Dewaele y Cani [DC04a], se puede partir de los movimientos que aparecen en la superficie para, de forma progresiva, expandirlos a lo largo del volumen como si de una *onda* de material se tratara. En este caso necesita definirse la dirección, y, en caso de haberlos, los cambios de orientación durante la propagación, de forma que el resultado final se aproxime lo más posible al resultado real que cabría esperar.

La segunda alternativa tiene una ventaja fundamental sobre la primera, y es que puede realizarse de forma progresiva o fraccionada. Dadas las restricciones de tiempo y la gran cantidad de vóxeles que es necesario procesar, un *algoritmo de cálculo retardado* (del inglés *lazy*) sería un camino efectivo para conseguir un efecto interactivo que posteriormente puede refinarse escalonadamente en momentos de baja carga computacional.

Se considerará pues que el material se *comprime* cuando la herramienta lo desplaza, llenándose los vóxeles adyacentes a la zona que desaloja la herramienta. Esta operación, que puede realizarse rápidamente, vendrá acompañada de un proceso posterior de *expansión* o de *relajación*, en el que la sobrecarga de estos vóxeles se distribuye en los vecinos, y así sucesivamente.

Soluciones anteriores han planteado un sistema parecido para la propagación de la deformación en un modelo voxelizado. No obstante, los trasvases de material propuestos se hacían siempre en las direcciones perpendiculares al sistema de coordenadas, \hat{x} , \hat{y} y \hat{z} , lo cual limita considerablemente el espectro de efectos que se puede conseguir. Teniendo en cuenta que en el sistema propuesto se conoce la dirección con que se realiza el empuje se

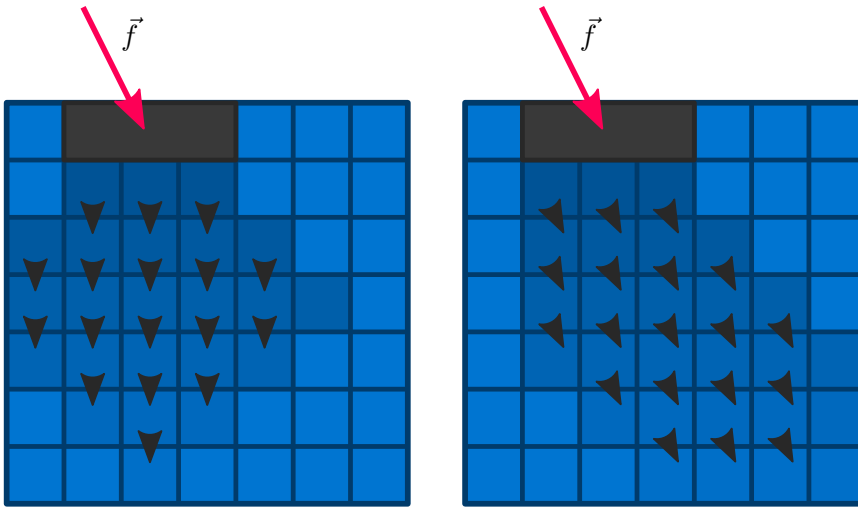


Figura 6.7 – Representación esquematizada del efecto conseguido con una propagación fija y con una propagación orientada según el empuje realizado. A la izquierda, se muestra una expansión en la que el material se reparte de forma equitativa a los vecinos prefijados. A la derecha, se orienta el trasvase de material para conseguir que el avance sea mayor en la dirección del empuje.

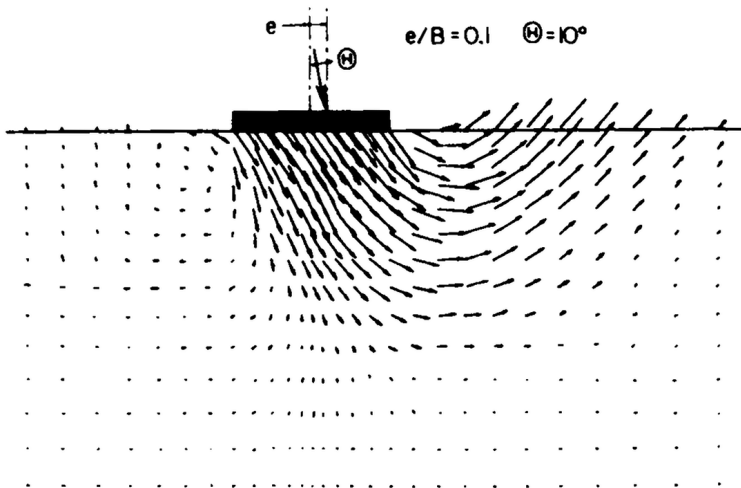


Figura 6.8 – Resultados la simulación bidimensional de Prevost al aplicar una herramienta plana con una tensión no paralela a la normal de la superficie. Se muestra el campo de velocidad obtenido con tres bandas reflectantes en los bordes inferior, izquierdo y derecho.

usará una expansión orientada, en la que la dirección principal de propagación será la del desplazamiento estimado \vec{u}^\sim . En la [Figura 6.7](#) se puede observar que el avance del material es siempre el mismo en una expansión sin orientar, mientras que si se usa la dirección del empuje pueden conseguirse un efecto más parecido al real (que se puede apreciar en la simulación de Prevost mostrada en la [Figura 6.8](#)). Se observa además que cuando se trasvasa material a los vecinos prefijados sin almacenar información sobre la orientación no se puede dirigir la expansión, creándose siempre la misma distribución independientemente de la dirección de \vec{f} .

Para conseguir orientar la expansión del material se redefinirán los vecinos usando la dirección de la deformación plástica $d\varepsilon_p$ que la origina. Con esta finalidad se definirán los *vectores de dispersión* \vec{w}_i , los cuales indican los vecinos a los que cada vóxel debe trasvasar material conforme avanza la expansión de material. Se definirá el vector de dispersión principal \vec{w} como:

$$\vec{w} = \frac{\vec{u}^\sim}{\max(u_x^\sim, u_y^\sim, u_z^\sim)} L_v \quad (6.19)$$

a saber, el vector de desplazamiento estimado \vec{u}^\sim escalado para que la mayor de sus componentes u_x^\sim , u_y^\sim y u_z^\sim coincida con el lado de un vóxel L_v . El vector principal se conocerá como la dirección *adelante*. Los vectores que apunten al vecino *izquierdo* \vec{w}_l , *derecho* \vec{w}_r , de *arriba* \vec{w}_u y *abajo* \vec{w}_d se pueden calcular haciendo el producto vectorial de \vec{w} con un vector arbitrario \vec{p} , de forma que:

$$\begin{aligned} \vec{w}_u &= \frac{\vec{w} \times \vec{p}}{p \sin(\theta)} \\ \vec{w}_d &= -\vec{w}_u \\ \vec{w}_r &= \frac{\vec{w} \times \vec{w}_u}{w} \\ \vec{w}_l &= -\vec{w}_r \end{aligned} \quad (6.20)$$

donde w es el módulo de \vec{w} , p el módulo de \vec{p} , y θ es el ángulo formado por los vectores \vec{w} y \vec{p} . Las operaciones que se realizan en el cómputo de \vec{w}_u y \vec{w}_r son para igualar el resultado del producto vectorial al módulo de \vec{w} . Puede comprobarse fácilmente que todos los vectores de dispersión son

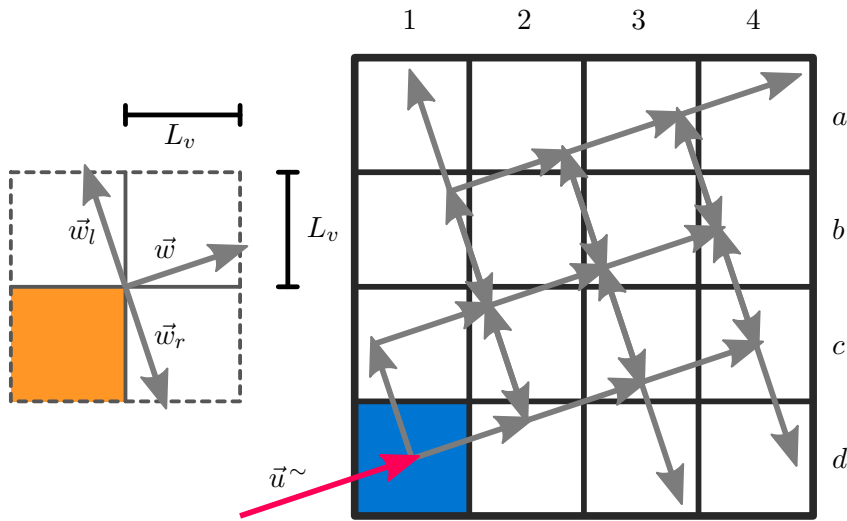


Figura 6.9 – Construcción de los vectores de dispersión principal, izquierdo y derecho en dos dimensiones, y relaciones de vecindad que aparecen con los mismos.

perpendiculares entre sí.

El resultado es un conjunto de vectores que *avanzan siempre uno y no más vóxeles* sumados a cualquier posición \vec{x} que pertenezca a un vóxel determinado. Esto se ilustra simplificado en dos dimensiones en la [Figura 6.9](#). Como se puede observar, al reducir la mayor componente del vector \vec{w} al lado de un vóxel L_v , se tiene que se siempre se puede encontrar un vecino cuando se suma \vec{w} a cualquiera de las posiciones de un vóxel determinado. Aparece por tanto una nueva definición de vecindad, con la que, partiendo del vóxel $1d$, se puede dispersar el material de forma acorde con la dirección de $\vec{u} \sim$. Por ejemplo, los vecinos del vóxel $1d$ son el $1c$ (delante) y el $2d$ (izquierdo), coincidiendo en este caso con la vecindad tradicional, si bien los vecinos del vóxel $2c$ son el $3b$ (delante), el $1b$ (izquierdo) y el $2d$ (derecho) para este reparto concreto.

Cada vóxel desalojado declarará pues su propia red de vecinos, basado en el conjunto de vectores de dispersión que calculan para cada desplazamiento plástico $du_p \hat{u} \sim$ que aparece. El algoritmo de dispersión será el responsable de organizar los trasvases de material según este esquema, procurando maximizar el realismo visual y reduciendo en lo posible la carga

de computación necesaria.

Los desplazamientos de material no sólo hacia el frente sino también hacia los lados simulan parcialmente el cambio de dirección de los desplazamientos que muestran los resultados de Prevost, que se pueden apreciar tanto en la [Figura 6.8](#) como en la [Figura 5.4](#), y que hacen que el material se desplace no sólo hacia *delante* sino también hacia los lados, y que incluso hacen que en ocasiones *retroceda* y rebose por la superficie sobre la que actuó la herramienta. Debe notarse no obstante que con este concepto de trasvase de material se pierde la equivalencia con los desplazamientos que se definió en la [Sección 6.3.1](#) mediante la función χ , alejándose inevitablemente esta solución del modelo formal estudiado, y acabándose gran parte de la fidelidad al modelo teórico en el movimiento inicial de material que realiza la herramienta a su paso.

6.3.3.4 Heurística de deformación: *algoritmo de dispersión recursiva*

Con los vectores de dispersión que se han definido se puede elaborar un mecanismo de lo más básico para expandir el material acumulado. Para cada vóxel que ve alterado su valor de propiedad, se dispersa por igual el sobrante entre los vecinos, según la definición de vecindad aportada por los vectores de dispersión.

El algoritmo se puede componer fácilmente de forma recursiva, ejecutando para cada vóxel modificado por el cambio en la superficie el siguiente procedimiento.

1. Se obtiene como parámetros el punto \vec{x} en el que se realiza la dispersión (según los cálculos de la red de vecinos definida por los vectores de dispersión), y una cantidad sobrante d_s de material.
2. Se comprueba si el sobrante se puede almacenar en este vóxel o está por encima del valor permitido, al que se referirá en lo que sigue como *densidad normal* del material y se notará como d_n .
 - (a) Si no se excede el valor normal, se almacena en este vóxel y no se realiza ninguna llamada recursiva.

- (b) Si se excede el valor normal, se deja en el vóxel sólo la cantidad justa para llegar al normal, a saber, $d_n - d(v(\vec{x}))$, y el resto $d_r = d_s - d_n + d(v(\vec{x}))$ se reparte en cinco partes iguales, llamándose al procedimiento cinco veces con los parámetros:
- i. punto $\vec{x} + \vec{w}_l$ con un quinto de d_r ,
 - ii. punto $\vec{x} + \vec{w}_r$ con un quinto de d_r ,
 - iii. punto $\vec{x} + \vec{w}_t$ con un quinto de d_r ,
 - iv. punto $\vec{x} + \vec{w}_b$ con un quinto de d_r , y finalmente
 - v. punto $\vec{x} + \vec{w}$ con el último quinto de d_r .

Esta implementación, si bien inmediata, sufre de un problema básico, y es el gran coste computacional que supone el considerable número de llamadas recursivas necesarias para resolver cada interacción que modifique la superficie. Se carece además de ningún criterio de prioridad, por lo que además puede que se invierta mucho tiempo en resolver pequeños repartos cuando hay zonas con mucho material acumulado que no se ha comenzado a dispersar siquiera.

Como problema adicional aparece la redundancia de las redes de reparto creadas para cada vóxel sobre el que se inicia el procedimiento, pues estas se solapan generando muchas llamadas duplicadas que no surten ningún efecto. Una segunda llamada sobre un vóxel que ya ha sido manipulado será especialmente ineficiente si este ya ha alcanzado la cantidad normal de material, pues deberá desplazarse todo el material sobrante con nuevas llamadas recursivas.

Este método es válido por tanto para interacciones *pequeñas*, en las que además el material admita cierto tipo de *compresión*: esto es, que el valor habitual de los vóxeles sea inferior al valor normal, de forma que se minimice el número de llamadas recursivas necesarias para resolver cada expansión.

En la [Figura 6.10](#) se puede observar el efecto conseguido con el algoritmo de dispersión recursiva sobre un modelo cúbico, el cual se ha visto sometido a una acción prolongada de empuje por parte de la herramienta, y donde se puede apreciar también el efecto de compresión del material que el algoritmo genera.

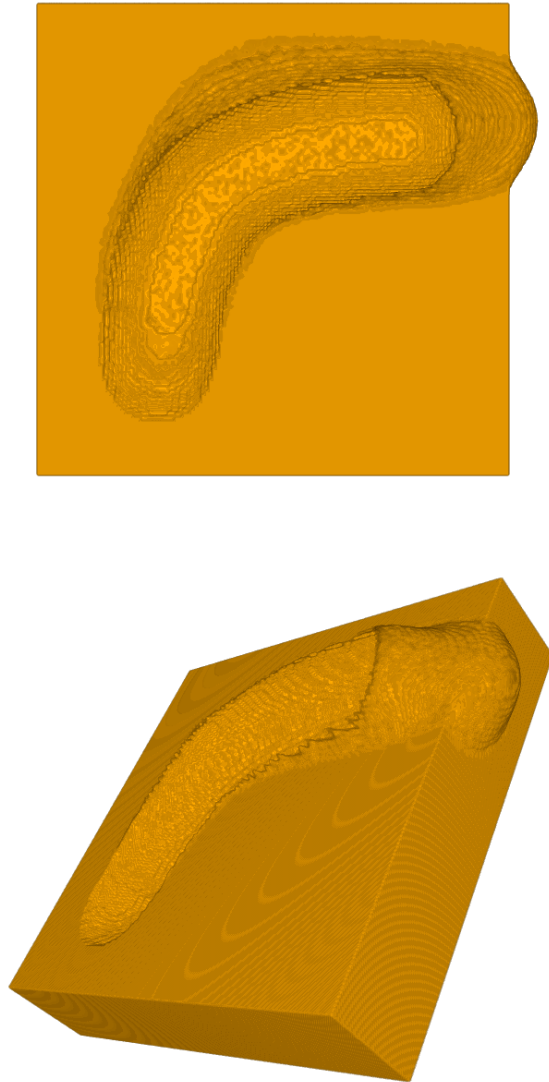


Figura 6.10 – Resultados conseguidos con el algoritmo de dispersión recursiva en un material de densidad uniforme con un tamaño de $128 \times 128 \times 30$ vóxeles. El color más oscuro en las zonas adyacentes al paso de la herramienta denota un incremento en el valor de densidad, obteniéndose por tanto un cierto factor de compresión en el material. Igual sucede con la masa de material arrastrado que se encuentra al final de la trayectoria.

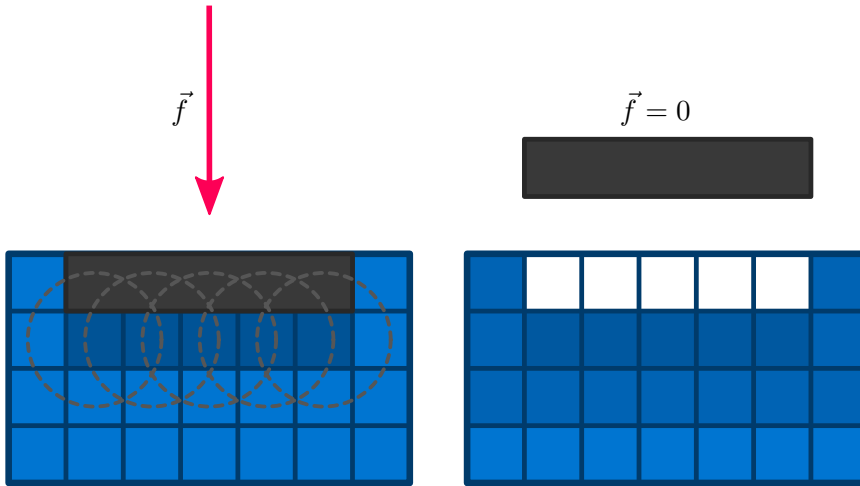


Figura 6.11 – Aplicación de una operación de suavizado en la discontinuidad que aparece al trasladar el material de la superficie por el empuje de la herramienta. Se difuminan las discontinuidades en la vecindad de cada vóxel (marcadas con los círculos punteados en la imagen), de forma que tras varias pasadas del operador el gradiente es más homogéneo.

6.3.3.5 Segunda aproximación de la deformación interna

Siguiendo la misma línea de métodos progresivos que se ha discutido anteriormente, puede diseñarse un segundo tipo de propagación o relajación del material acumulado si se considera la sobrecarga que reciben los vóxeles cercanos a la herramienta cuando se deforma la superficie como un *contorno* o *discontinuidad* a solventar. Aplicando el concepto de *filtro de suavizado* se podría transformar la distribución de material mediante un operador progresivo que promedie las densidades de la zona, convirtiendo de forma incremental los saltos grandes de densidad entre vecinos en incrementos graduales, y minimizándose así la variación del gradiente de densidad. Este concepto se ilustra en la [Figura 6.11](#), donde se puede apreciar de forma cualitativa el efecto de *difuminado* que se consigue al redistribuir los valores de densidad de cada vecindad en torno al promedio encontrado.

Cabe resaltar que esta técnica no hace uso de los vectores de dispersión antes diseñados, y que cualquier operación de suavizado no orientada creará un efecto únicamente basado en los gradientes locales de densidad de

los vóxeles y que se atenuará con la distancia euclídea al foco. El resultado, por tanto, será independiente de la orientación del empuje, dependiendo únicamente de la trayectoria de la herramienta y de los vóxeles desplazados por su acción directa. Sigue además sin haber una equivalencia con un campo de desplazamientos según los términos definidos anteriormente.

En este contexto la tarea más crítica será el diseño del filtro que se aplicará ante una sobrecarga. Su naturaleza no sólo determinará la distribución de final de valores de propiedad, sino que además influirá en la velocidad de convergencia del algoritmo. En una primera aproximación se usará un filtro sencillo que promediará el valor de densidad en la vecindad de cada vóxel y reajustará los valores de propiedad del propio vóxel y de sus vecinos para acercarlos a dicho promedio.

6.3.3.6 Heurística de deformación: *operador de suavizado*

En torno a este concepto de filtrado se construirá el *el algoritmo de suavizado*, un procedimiento iterativo que actuará ante un proceso de carga plástica para expandir o suavizar la carga adicional de material generada por el movimiento de la herramienta. El algoritmo de suavizado comenzará por aplicar el filtro a los vóxeles cuya densidad ha cambiado por efecto de los trasvases de material en la superficie, repitiéndose dicha acción en cadena para cada vóxel que ve alterado su valor de propiedad durante su actuación.

Con esta finalidad se creará la cola de *vóxeles activos*, en la cual se añadirá cualquier vóxel que vea alterado su valor de propiedad, ya sea por los desplazamientos de material de la superficie o por el propio funcionamiento del algoritmo. Se encontrarán por tanto allí los vóxeles cuya vecindad es necesario analizar para comprobar si la distribución de material es irregular (por lo que habrá que aplicar el filtro) o es admisible (y puede por tanto permanecer inalterada).

Debe mantenerse asimismo una lista de *vóxeles prohibidos*, que serán aquellos que se encuentren en el volumen ocupado por la herramienta durante la deformación, que ya de hecho han sido desalojados y a los que no se podrá en ningún caso enviar material durante la relajación. Esto evitará que durante el procedimiento de suavizado el operador envíe material



Figura 6.12 – Resultados conseguidos con el operador de suavizado. Se muestra el resultado de realizar varias incisiones a un volumen cúbico de densidad uniforme con una resolución de 128 vóxeles por lado. La densidad se representa con una gradación de color, siendo el negro el valor más bajo, el blanco el más alto y usándose un tono amarillo para los valores intermedios.

a la zona donde físicamente estaría la herramienta.

Se supondrá además un valor *normal* de densidad, el cual se considerará el cupo de material que tiene un vóxel en el estado de reposo. Este concepto se usará como valor horizonte, para evitar que el material se expanda indefinidamente. De no bloquearse la actuación del filtro ante un valor mínimo, el algoritmo trataría de suavizar la discontinuidad que aparece en la superficie del modelo, en la cual los vóxeles contactan con vecinos con densidad nula.

Con todo esto, el algoritmo se compondrá de los siguientes pasos:

1. Se añaden a la cola de activos todos los vóxeles que hayan recibido material por un cambio en la superficie.
2. Para cada uno de los vóxeles activos:
 - (a) Se saca de la lista el vóxel activo más antiguo v_c
 - (b) Se obtiene su valor de densidad, que se notará por d_c
 - (c) Se calcula la densidad media d_m en los n de los 26 vecinos $N(v_c)$ que *no* estén vacíos y que *no* estén en la lista de prohibidos.
 - (d) Si la densidad del vóxel central d_c está por encima de la media d_m y por encima del valor normal d_n :
 - i. Se deja el vóxel central una cantidad de material igual a la media d_m , o o igual a al valor normal d_n si $d_n > d_m$
 - ii. El exceso se reparte entre los vecinos no prohibidos de forma aleatoria, aumentando su densidad hasta la media o el valor normal, lo que sea mayor. Los vecinos que ya tengan un valor superior a ambos valores no se alteran, excluyéndose del proceso y seleccionándose otro vecino aleatorio para continuar el reparto. Todo vóxel alterado deberá ponerse en la cola de activos.

Cada vez que se modifique la superficie y por tanto aparezcan desplazamientos nuevos de material se añadirán los vóxeles que reciban material a la lista de activos. La modificación de la superficie tendrá prioridad ante

el proceso de relajación, de forma que los cambios realizados por la herramienta se produzcan con la mayor inmediatez.

La tolerancia o criterio de parada del algoritmo será cuando todos los vóxeles analizados tengan una densidad inferior a la media de su vecindad, o al valor normal.

Este procedimiento permite, al igual que el algoritmo de dispersión recursiva, cierto factor de compresión en el material, si bien en este caso el efecto es intrínseco al funcionamiento del algoritmo y no es posible configurarlo. El operador de suavizado permite que los vóxeles que pasan por el proceso vean incrementada su densidad original previa a la modificación de la superficie, siempre y cuando no superen el valor medio de su vecindad. Puede por tanto darse tras una interacción un gradiente de densidad no nulo en el interior, incluso en el caso en el que estado inicial del volumen sea un valor de propiedad homogéneo para todos sus vóxeles.

6.3.3.7 Tercera aproximación a la deformación interna

Todos los métodos que se han propuesto hasta ahora han garantizado la conservación de la cantidad total de material. La integral en volumen de la densidad:

$$\sum_{v_i \in V} d(v_i) \quad (6.21)$$

permanece constante, lo cual es una propiedad que, aunque no estrictamente necesaria, es deseable para cualquier método de deformación.

No obstante, tanto el algoritmo de dispersión recursiva como el algoritmo de suavizado presentan cierta tolerancia a la compresión, lo cual en la práctica resulta en una pérdida efectiva del volumen ocupado: la misma cantidad de material se encuentra almacenada en menos vóxeles con mayor densidad. Esto puede entenderse, a ojos del usuario, como una pérdida de material, lo cual puede ser un efecto poco deseable en determinadas ocasiones.

Se puede proponer para solucionar esta máxima una variación del enfoque anterior. Basándose en el mismo principio de un *operador* que modifica localmente las densidades de los vecinos, y usando el concepto anteriormente definido de densidad *normal* o de reposo, se puede conseguir un efecto de

expansión del material que no permita compresión si se rebaja la cantidad de material en el vóxel procesado hasta la densidad *normal* en lugar de hasta la densidad *media*. El procedimiento será en gran medida similar a su predecesor, si bien el modelo no presentará tras la relajación el efecto de compresión que antes se generaba.

6.3.3.8 Heurística de deformación: *operador de desplazamiento*

Se asume por tanto para esta nueva heurística que el material, en estado de reposo, presenta una densidad igual a la *densidad normal* d_n que antes ya se había definido. Se realiza por tanto un giro en la definición, pues si bien antes se permitían densidades superiores a d_n , teniendo este valor por tanto el estatus de cota inferior, se considera ahora como un límite superior por encima del cual el sólido se entiende comprimido y se procede a su relajación.

El nuevo operador funcionará no obstante de forma similar al *operador de suavizado*, excepto por el hecho de que todo vóxel procesado deberá sesgarse a la densidad normal, repartiéndose el exceso entre sus vecinos. Esto implica un cálculo especial para la cota de reparto. Si bien antes se usaba la densidad media para decidir qué vecinos recibían material y cuáles no, se usará ahora el siguiente cómputo para la densidad límite d_l :

$$d_l = \frac{1}{n} \left(\sum_{v_i \in N} d(v_i) + d_r \right) \quad (6.22)$$

donde N es el subconjunto de los vecinos que no están vacíos y no están en la lista de prohibidos, $n = |N|$ su cardinal y d_r el resto a repartir, diferencia entre la densidad actual del vóxel y la densidad normal con la que quedará tras el procesamiento. Puede observarse que este cálculo es en definitiva una medida de la densidad media en la vecindad a la que se le añade la parte proporcional de resto que debería admitir cada vecino (esto es, el término d_r/n).

Con esta nueva definición se puede diseñar un algoritmo similar al ya propuesto, que sin embargo crea una dispersión sin compresión:

1. Se añaden a la cola de activos todos los vóxeles que hayan recibido

material por un cambio en la superficie.

2. Para cada uno de los vóxeles activos:

- (a) Se saca de la lista el vóxel activo más antiguo v_c
- (b) Se obtiene su valor de densidad, que se notará por d_c
- (c) Se calcula la densidad límite d_l en los n de los 26 vecinos $N(v_c)$ que *no* estén vacíos y que *no* estén en la lista de prohibidos.
- (d) Si la densidad del vóxel central d_c está por encima de la densidad límite d_l y por encima del valor normal d_n :
 - i. Se deja el vóxel central una cantidad de material igual a la normal d_n
 - ii. El exceso se reparte entre los vecinos no prohibidos de forma aleatoria, aumentando su densidad hasta la densidad límite o el valor normal, lo que sea mayor. Los vecinos que ya tengan un valor superior a ambos valores no se alteran, excluyéndose del proceso y seleccionándose otro vecino aleatorio para continuar el reparto. Todo vóxel alterado deberá ponerse en la cola de activos.

Nótese que este algoritmo es idéntico al anterior, excepto por la sustitución de la densidad media por la densidad límite d_l definida anteriormente, y la parte en la que se fuerza al vóxel central a quedarse con la cantidad normal de material d_n . Se comprueba fácilmente, por la condición de procesamiento de cada vóxel ($d_c > d_n$) y por el tratamiento de cada vóxel procesado, que no puede quedar en la zona procesada ningún vóxel con densidad superior a la normal.

Al igual que sucedía en el algoritmo de suavizado, se tratará la acomodación de la superficie como una tarea prioritaria, garantizándose así la agilidad de los cambios en la zona afectada por la herramienta.

El algoritmo produce el efecto esperado, consiguiéndose una dispersión no orientada muy similar a la que producía su antecesor pero sin permitir compresión en el material. No obstante aparece un problema relacionado con la velocidad de convergencia: el hecho de que se envíe el material a una dirección aleatoria repercute en el número de iteraciones necesarias

para que el modelo llegue a un estado de reposo. En muchas ocasiones el material se pasa a un vóxel que ya ha sido procesado y que ya estaba en la densidad normal, debiendo procesarse de nuevo con la correspondiente penalización en tiempo de procesamiento.

6.3.3.9 Heurística de deformación: *operador de desplazamiento orientado*

La velocidad de convergencia no deja de ser un parámetro importante, pues aunque la superficie se deforme de manera inmediata, la redistribución interna del material puede tardar varios segundos en resolverse. Los movimientos internos no dejan de ser un efecto visible en tanto que en ocasiones acaban aflorando en la superficie, por lo que obviamente interesa resolverlos cuanto antes.

Puede optimizarse el reparto de material recuperando la idea de la deformación direccional que se desarrolló para llegar a los vectores de dispersión. Si se asiste el reparto realizado por el operador de desplazamiento de forma que se reconozcan como prioritarios los vecinos que se encuentran en la dirección de los desplazamientos estimados \vec{u}^{\sim} .

Con esta finalidad se creará entonces un índice que contendrá, para cada vecino $v_i \in N(v_c)$, el ángulo que forma con el resto de vecinos $v_j \in N(v_c) - v_i$, considerando las direcciones de los vecinos que aparecen en el [Cuadro 6.2](#). Habrá por tanto 26 listas ordenadas, una por cada vecino, cada una de las cuales permitirá consultar los vecinos con una dirección *más cercana* dada la dirección de un vecino concreto.

El procedimiento de relajación será por tanto parecido sus dos predecesores, excepto por la dirección en la que se reparte el material. Tendrá preferencia siempre el vecino que esté en la dirección más parecida a los desplazamientos estimados \vec{u}^{\sim} , y se comenzará a repartir a ese vecino y a los más cercanos según el orden que se ha descrito. No obstante, y para simular los movimientos laterales, se descartarán aleatoriamente un número determinado de vecinos cercanos al principio del reparto, comenzando a repartirse en una dirección con cierta divergencia que se elige al azar conforme se avanza. La aplicación del operador de desplazamiento orientado queda entonces:

1. Se añaden a la cola de activos todos los vóxeles que hayan recibido

Capítulo 6. Plasticidad

material por un cambio en la superficie, junto con la dirección \vec{m} de la que ha venido el material que se le añadió.

2. Para cada uno de los vóxeles activos:

- (a) Se saca de la lista el vóxel activo más antiguo v_c junto con su dirección de reparto \vec{m}

Cuadro 6.2 – Lista de vecinos y sus direcciones relativas respecto del vóxel central. Los índices se calculan como $1/L_v(3^2(\Delta x + L_v) + 3^1(\Delta y + L_v) + 3^0(\Delta z + L_v))$, con L_v el lado del voxel.

Índice	Δx	Δy	Δz
0	$-L_v$	$-L_v$	$-L_v$
1	$-L_v$	$-L_v$	0
2	$-L_v$	$-L_v$	$+L_v$
3	$-L_v$	0	$-L_v$
4	$-L_v$	0	0
5	$-L_v$	0	$+L_v$
6	$-L_v$	$+L_v$	$-L_v$
7	$-L_v$	$+L_v$	0
8	$-L_v$	$+L_v$	$+L_v$
9	0	$-L_v$	$-L_v$
10	0	$-L_v$	0
11	0	$-L_v$	$+L_v$
12	0	0	$-L_v$
14	0	0	$+L_v$
15	0	$+L_v$	$-L_v$
16	0	$+L_v$	0
17	0	$+L_v$	$+L_v$
18	$+L_v$	$-L_v$	$-L_v$
19	$+L_v$	$-L_v$	0
20	$+L_v$	$-L_v$	$+L_v$
21	$+L_v$	0	$-L_v$
22	$+L_v$	0	0
23	$+L_v$	0	$+L_v$
24	$+L_v$	$+L_v$	$-L_v$
25	$+L_v$	$+L_v$	0
26	$+L_v$	$+L_v$	$+L_v$

- (b) Se obtiene su valor de densidad, que se notará por d_c
- (c) Se calcula la densidad límite d_l en los n de los 26 vecinos $N(v_c)$ que *no* estén vacíos y que *no* estén en la lista de prohibidos.
- (d) Si la densidad del vóxel central d_c está por encima de la densidad límite d_l y por encima del valor normal d_n :
 - i. Se deja el vóxel central una cantidad de material igual a la normal d_n
 - ii. Tomando la lista de los vecinos cercanos al vecino en la dirección más parecida a m , se descartan aleatoriamente entre 0 y 8 vecinos cercanos.
 - iii. El exceso se reparte entre los vecinos no prohibidos tomados en orden a partir del vecino cercano resultante, aumentando su densidad hasta la densidad límite o el valor normal, lo que sea mayor. Los vecinos que ya tengan un valor superior a ambos valores no se alteran, excluyéndose del proceso y seleccionándose el siguiente vecino cercano para continuar el reparto. Todo vóxel alterado deberá ponerse en la cola de activos, junto con su dirección de vecindad tomada del [Cuadro 6.2](#).

Se puede apreciar que es necesario almacenar en la lista de vecinos la dirección en la que se *está realizando* la propagación del material, para que, al repartir el exceso entre los vecinos se sepa qué dirección hay que seguir.

Excluyendo el reparto, el algoritmo es idéntico al que realiza la propagación no orientada. Sin embargo produce una reducción notable del tiempo necesario para resolver cada propagación, pues la información adicional sobre la dirección de la propagación que se arrastra en todo el algoritmo evita en gran medida retornos innecesarios de material, de modo que será poco probable que un vóxel ya procesado reciba de nuevo material de los vecinos a los que él mismo envió material.

El número de vecinos cercanos que se descartan (o, dicho de otro modo, el margen de divergencia de la dirección de propagación) influye directamente tanto en el rendimiento como en el resultado visual. Un número demasiado bajo producirá una propagación demasiado fija que creará un

efecto artificial, mientras que un número demasiado alto hará más probables los retornos, aumentando así necesariamente el tiempo requerido para resolver totalmente la deformación. El número máximo de 8 fue elegido de forma experimental por ser el número más bajo de vecinos cercanos que se pueden descartar sin que se produzcan anomalías claramente visibles.

Los efectos visuales no son significativamente distintos de la versión no orientada del algoritmo, residiendo la mejora más notable de esta revisión en del tiempo necesario para completar la expansión desencadenada por una interacción.

6.3.3.10 Heurística de deformación: *operador de desplazamiento orientado con prioridad*

Una observación clara cuando se analiza la acción del algoritmo orientado es que los vóxeles activos se tratan *por orden* estricto de entrada en la cola de vóxeles activos. La consecuencia directa de esto es que, si el usuario intenta una segunda interacción mientras se está resolviendo una primera, la superficie deformada quedará en el estado intermedio de compresión hasta que no termine de resolverse la parte pendiente de la anterior. Esto puede producir un retardo de algunos segundos en los que puede estar resolviéndose una deformación menor, mientras que en la zona de la nueva interacción no se realiza ningún trabajo.

Sería deseable de hecho *priorizar* la dispersión de forma que el usuario percibiera cierta inmediatez en la zona adyacente sobre la que la herramienta está actuando, mientras que los efectos marginales no tan expuestos se pudieran resolver cuando la carga computacional fuera menor.

Considerando que existe un estado de carga *normal* de material para los vóxeles, representado en forma de la densidad normal d_n , se puede utilizar como criterio de prioridad el nivel de *sobrecarga material* que soportan los vóxeles modificados, de forma que se procesarán primero los vóxeles que tengan una mayor carga, dejando para el final aquellos cuyo monto adicional los deje más cerca de la densidad normal. Este tratamiento de la cola con prioridad se puede especificar fácilmente con una ligera variación sobre su versión anterior a la hora de escoger el siguiente vóxel activo a procesar:

1. Se añaden a la cola de activos todos los vóxeles que hayan recibido material por un cambio en la superficie, junto con la dirección \vec{m} de la que ha venido el material que se le añadió.
2. Para cada uno de los vóxeles activos:
 - (a) Se saca de la lista el vóxel activo más antiguo v_c de entre los que tienen el mayor valor de densidad d_c más alta de la cola, junto con su dirección de reparto \vec{m}
 - (b) Se obtiene su valor de densidad, que se notará por d_c
 - (c) Se calcula la densidad límite d_l en los n de los 26 vecinos $N(v_c)$ que *no* estén vacíos y que *no* estén en la lista de prohibidos.
 - (d) Si la densidad del vóxel central d_c está por encima de la densidad límite d_l y por encima del valor normal d_n :
 - i. Se deja el vóxel central una cantidad de material igual a la normal d_n
 - ii. Tomando la lista de los vecinos cercanos al vecino en la dirección más parecida a m , se descartan aleatoriamente entre 0 y 8 vecinos cercanos.
 - iii. El exceso se reparte entre los vecinos no prohibidos tomados en orden a partir del vecino cercano resultante, aumentando su densidad hasta la densidad límite o el valor normal, lo que sea mayor. Los vecinos que ya tengan un valor superior a ambos valores no se alteran, excluyéndose del proceso y seleccionándose el siguiente vecino cercano para continuar el reparto. Todo vóxel alterado deberá ponerse en la cola de activos, junto con su dirección de vecindad tomada del [Cuadro 6.2](#).

Intuitivamente se puede ver que por una parte este ordenamiento de la lista efectivamente produce resultados más rápidos, y que además aparece un efecto secundario considerablemente ventajoso: al interrumpirse una tarea de dispersión ya avanzada para iniciar otra incipiente puede evitarse parte del procesamiento de la primera si las interacciones fueron cercanas. Si llegan a unificarse las dos ondas de expansión se reducirán de forma notable

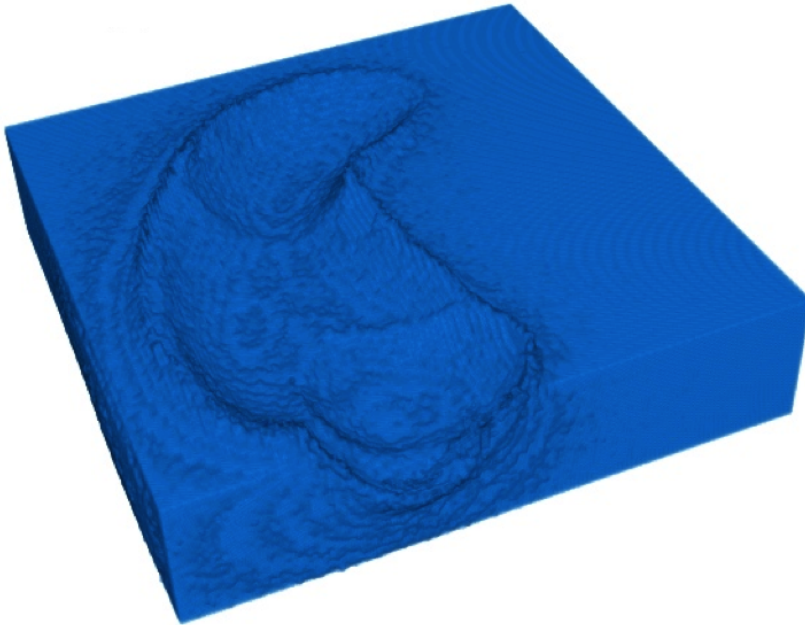


Figura 6.13 – Resultados conseguidos con el operador de desplazamiento. Se muestra el resultado de realizar un trazo curvo con la herramienta en una plancha de material de densidad uniforme con una resolución de 128 vóxeles de lado y 32 de grosor.

el número de operaciones de desplazamiento necesarias, con la consecuente mejora en el tiempo de llegada al estado de completa relajación.

Esta bonificación en el procesamiento de la parte más densa (que suele coincidir precisamente con la zona por donde ha pasado la herramienta), unido al hecho de que la modificación de la superficie sigue siendo una tarea prioritaria, consigue que el avance de la herramienta se resuelva de forma mucho más ágil.

Los efectos de la familia de algoritmos del operador de desplazamiento pueden apreciarse en la [Figura 6.13](#).

6.3.3.11 Mejora del efecto visual

Una cuestión determinante de cara al resultado final de la deformación en la familia de algoritmos basados en filtros (esto es, el algoritmo del operador de suavizado y los algoritmos del operador de desplazamiento) es la dirección

en la que se desplaza el material de la superficie, reubicado por la acción directa de la herramienta.

Se deduce de toda la teoría discutida que deben desplazarse en la dirección de los desplazamientos estimados \vec{u}^{\sim} , si bien experimentalmente se probó una alternativa que mejora conceptualmente el resultado obtenido tras la relajación. Si se envía el material en la dirección perpendicular a la normal de la herramienta esférica la configuración final del material termina siendo más parecida a los resultados obtenidos por Prevost, en los que los desplazamientos cambian de dirección y terminan empujando el material de la superficie sobre la que se actuó (como se aprecia en la [Figura 6.8](#) y en la [Figura 5.4](#)). Aunque este resultado depende del número de bandas reflectantes (inelásticas) que se determinen en la simulación, parece intuitivo que la marca dejada por la herramienta debería *esparcir* cierta cantidad de material hacia los lados, provocando un crecimiento parcial de la superficie en el contorno del valle creado.

Los resultados de la [Figura 6.12](#) y la [Figura 6.13](#) se han conseguido con esta variación del comportamiento inicial propuesto, que, si bien reduce un poco más la fidelidad al modelo teórico analizado, encaja mejor con el funcionamiento al fin y al cabo heurístico que se ha diseñado para resolver el problema de la deformación interna del material.

6.4 Resultados

Para evaluar los resultados de las técnicas aquí descritas se construyó una aplicación propia de escultura virtual, que permite representar modelos volumétricos y que además admite la interacción con el dispositivo háptico que se usó para estudiar e implementar la simulación táctil en el [Capítulo 5](#). Bajo este paraguas se implementan las tres técnicas diseñadas para simular la deformación plástica: el algoritmo de dispersión recursiva, el operador de suavizado y el operador de desplazamiento, en sus tres variantes, integradas en la aplicación como la operación de *empujar* o *desplazar* material. Se realiza también en este contexto la unión de la deformación plástica con el método de estimación de la fuerza, permitiéndose así interactuar con el modelo de forma completa. Esta integración se basa en la supervisión de la

estimación de la fuerza que se ha diseñado para poder acometer los cambios en la superficie en cuanto se llega a la tensión de fluencia. Unificando todas las soluciones expuestas en este capítulo el método general para resolver la deformación plástica queda como sigue:

1. **Monitorización de la deformación elástica.** Se espera hasta que la estimación de la tensión σ^{\sim} alcance el valor de la tensión de fluencia $\sigma_f(\alpha^{\sim})$, atendiendo a la condición que aparece en la expresión 6.13.
2. **Llegada a la tensión de fluencia.** Cuando se cumple dicha condición se controla la siguiente variación en la posición de la herramienta para saber si σ^{\sim} crecerá o decrecerá, de lo que dependerá el signo del incremento $d\sigma$. Si es negativo, se vuelve al paso 1. Si es positivo, se procede con el paso siguiente.
3. **Cálculo de la deformación plástica.** Se calcula el incremento de la tensión y consecuentemente de la fuerza según se indica en la expresión 6.15. Además se calcula qué parte de dicho incremento $d\sigma$ se invierte en un desplazamiento plástico du_p según lo descrito en 6.16.
4. **Deformación de la superficie.** Para ello se aplica el desplazamiento plástico du_p a los vóxeles en contacto con la herramienta, realizándose un trasvase de material efectivo a los vecinos según la equivalencia descrita en la Sección 6.3.1. Se marca todo vóxel que reciba material como *activo*.
5. **Relajación del material.** Se llama al procedimiento de relajación pertinente con la lista de vóxeles activos que se ha compuesto para resolver la compresión generada por la deformación de la superficie y concretar los cambios internos en el material.

La modificación de la superficie es un procedimiento mucho más rápido que la relajación del material comprimido, pues por una parte afecta a un número menor y en cualquier caso acotado de vóxeles y por otra únicamente es necesario realizar un movimiento de material para cada vóxel afectado. Por contra la relajación requiere un número de operaciones de trasvase que es desconocido de antemano, pudiendo tomar mucho más tiempo del que

el usuario invierte en realizar la interacción. El algoritmo del operador de desplazamiento, por ejemplo, requiere en el mejor de los casos del orden de 10-30 segundos para resolver una operación de empuje de mediano tamaño.

Esta situación motiva la inclusión de un mecanismo de interrupción en la resolución de la deformación. De esta forma la tarea más prioritaria será siempre la modificación de la superficie, pudiendo retrasarse o incluso detenerse la relajación si aparece la necesidad de realizar un nuevo cambio superficial. Anteponiendo la resolución de los cambios externos a la modificación interna se genera una mayor sensación de fluidez con la misma potencia de cálculo. Tómese como ejemplo la interacción realizada en la [Figura 6.13](#) y resuelta con el operador de desplazamiento orientado con prioridad. El trazo mostrado se realiza en siete segundos, intercalándose en varias ocasiones durante este intervalo los cambios superficiales y los internos. Cuando se detecta un nuevo movimiento de la herramienta que genera otra deformación plástica se abandona la relajación, para volver a retomarla cuando se ha completado la modificación de la superficie. Este mecanismo evita tener que esperar la resolución completa de los efectos globales generados por cada movimiento de la herramienta, lo cual supondría un grave inconveniente ya que en el caso citado este proceso se alarga hasta casi un minuto. Cabe destacar que esta priorización de las tareas se puede realizar únicamente cuando se utilizan los algoritmos del operador de suavizado y del operador de desplazamiento, pues por la naturaleza procedural del algoritmo de dispersión recursiva no es posible interrumpir el procedimiento de relajación, debiendo esperarse a la resolución completa de los cambios internos antes de poder proseguir con la interacción.

6.5 Experimentos

Los tres algoritmos que se han diseñado para resolver la deformación plástica producen efectos muy distintos y además requieren una cantidad de tiempo variable para resolver una misma interacción. Adicionalmente, los distintos enfoques presentan evoluciones distintas en el tiempo, por lo que se plantea como necesaria una comparación detallada de las tres soluciones propuestas. En lo que sigue se detallan los resultados de una serie de ex-

perimentos que se han previsto para medir el alcance y desempeño de las distintas facetas del método de deformación que aquí se ha diseñado.

6.5.1 Direccionalidad del desplazamiento de material con los vectores de dispersión

Conviene corroborar que la dispersión orientada produce un efecto distinto del que se consigue con la no orientada tal y como se afirmó en el análisis previo al diseño de las soluciones de relajación (ver [Figura 6.7](#)). Con esta motivación se programará una interacción con un modelo, resolviéndose tanto la deformación de la superficie como la relajación del material:

Con información de orientación Se usarán para ello los vectores de dispersión que se diseñaron como apoyo del algoritmo de dispersión recursiva en la [Sección 6.3.3.3](#).

Sin información de orientación La modificación de la superficie y la relajación se limitarán a seguir la estructura básica de vecindad para los trasvases de material, usando el principio heurístico explicado en la [Sección 6.3.3.11](#).

En ambos casos la relajación interna del material se llevará a cabo con la ayuda del algoritmo de dispersión recursiva, efectuándose la interacción con la misma herramienta, a la misma velocidad y sobre el mismo modelo, siendo por tanto la única variable la información de orientación. Se realizará la prueba sobre dos trayectorias distintas: una en curva y otra que realiza un cambio brusco de dirección, en las que el movimiento de la herramienta comenzará en la esquina inferior izquierda del modelo y terminará en la esquina superior derecha del mismo.

Al seguir la herramienta las trayectorias descritas se crean los surcos en el material que se muestran en la [Figura 6.14](#). Se aprecia en las imágenes que existe un salto cualitativo entre el efecto creado por la dispersión orientada (izquierda) y la no orientada (derecha), especialmente cuando se producen cambios bruscos en la dirección del avance. En la propagación dirigida por los vectores de dispersión se acumula mayor cantidad de material en la dirección del avance de la herramienta (y por tanto en la dirección de la fuerza estimada). En la propagación no orientada por contra se envía

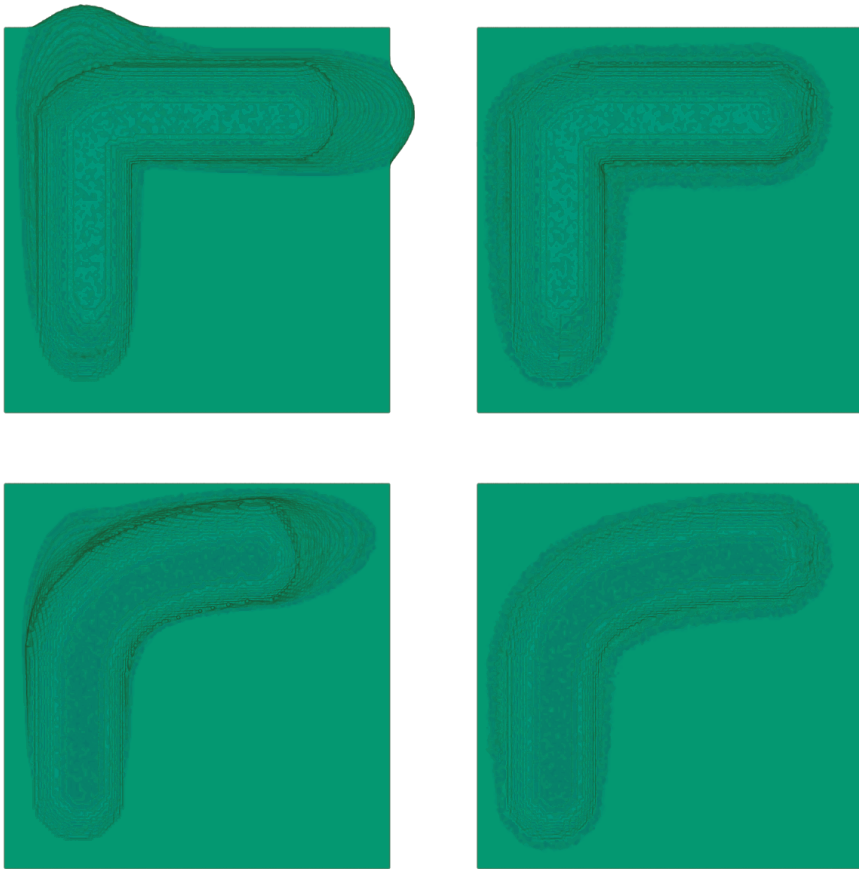


Figura 6.14 – Resultados visuales del experimento de comparación entre la dispersión orientada y la no orientada. Izquierda, resultados obtenidos con dispersión orientada. Derecha, resultados de la dispersión no orientada.

una cantidad homogénea de material en todas las direcciones, al no tenerse conocimiento de la dirección que origina el empuje. La información de la dirección en los repartos de material es por tanto completamente determinante, poniéndose de manifiesto que la dispersión orientada que se ha diseñado es una herramienta potente y que permite conseguir determinados efectos de acumulación de material que de otra forma sería difícil conseguir.

6.5.2 Comparativa de los tres algoritmos propuestos

Para tener una medida objetiva de la variación en tiempo de procesamiento que existe entre las distintas técnicas de relajación aplicadas se someterán los tres algoritmos a una interacción programada que requerirá exactamente los mismos cambios en la superficie, pudiendo así observarse de forma aislada la fase de relajación.

La interacción consistirá en el movimiento rectilíneo de la herramienta, que avanzará realizando una operación de arrastre paralela a la superficie de una plancha de material de $128 \times 128 \times 30$ vóxeles de ancho, largo y fondo. La incisión tendrá una profundidad de 12 vóxeles, y el trazo una longitud de 75 vóxeles. El movimiento será totalmente paralelo al eje vertical y se ejecutará arrastrando una herramienta de radio 16 vóxeles. Los resultados recogidos se presentan en el [Cuadro 6.3](#).

Cuadro 6.3 – Resultados numéricos del experimento de comparación de los tres algoritmos de relajación. Se indica para cada algoritmo el número de vóxeles que se han procesado, el tiempo total invertido en la relajación, la velocidad de procesamiento (en vóxeles por segundo) y el factor de compresión del material desplazado.

Algoritmo	Vóxeles	Tiempo (ms)	Velocidad (vóx/ms)	Compresión (%)
Dipersión recursiva	18851370	6422	2935.44	62.96
Op. suavizado	679594	3938	172.57	82.08
Op. desplazamiento	4695094	27438	171.12	-63.91

A la vista de los datos se comprueba que el algoritmo más rápido es sin lugar a dudas el del operador de suavizado, que invierte para resolver esta interacción poco más de la mitad del tiempo que el siguiente más rápido,

que es la dispersión recursiva. Esto confirma la previsible disminución del tiempo de procesado al permitirse compresión en el material. Por otra parte se observa que la velocidad de procesamiento del algoritmo de dispersión recursiva supera a los otros dos en un orden de magnitud, lo cual tiene sentido si se tiene en cuenta que el procesamiento que la dispersión recursiva hace para cada vóxel apenas es una división de la cantidad de material, mientras que los algoritmos de los operadores necesitan calcular los promedios en la vecindad y realizar un mayor número de repartos. La cantidad de vóxeles que se procesan en la dispersión recursiva es su mayor contrapartida: la creación de una red de reparto aislada para cada vóxel de la superficie que se modifica es lo que deriva en el elevado número de elementos que es necesario procesar. Se puede también resaltar lo desproporcionado del tiempo de procesamiento requerido por el operador de desplazamiento, si bien cabe también añadir que, al priorizarse la relajación, los cambios mayores se realizan en los primeros segundos y que el tiempo total que en estos resultados se muestra incluye la resolución de los cambios menores. La compresión del material se calcula con la relación entre el número total de vóxeles que se vacían al desplazarse su material durante la fase de modificación de la superficie y la cantidad de vóxeles que finalmente quedan vacíos respecto de la configuración inicial. El factor de compresión será por tanto del 100 % si en la fase de relajación no se descomprime nada y del 0 % si se acaban llenando el mismo número de vóxeles que se vaciaron. Se observa entonces que el algoritmo de dispersión recursiva sólo aloja material en 37 vóxeles por cada 100 que fueron vaciados al modificar la superficie (factor de compresión del 63 %), y que el operador de suavizado es el algoritmo que más comprime el material (compresión del 82 %), lo cual era previsible dado que permite incrementos mayores en la densidad del medio respecto de la configuración inicial. El operador de desplazamiento presenta una compresión negativa, lo cual se explica por el hecho de que no permite que ningún vóxel quede tras la relajación con una densidad mayor a la normal. Esta situación deriva en pequeños restos de material que terminan aflorando en la superficie, generándose así una *expansión* efectiva del espacio ocupado por el sólido.

En la [Figura 6.15](#) se muestran los resultados visuales de este experimento. Se puede apreciar allí gracias al código de colores la compresión en

el material que permiten tanto el operador de suavizado como el algoritmo de dispersión recursiva. Se observa de igual modo que el operador de desplazamiento deja todos los vóxeles como mucho a la densidad normal al final del procesamiento, y que además es el único de los tres algoritmos que consigue que el efecto de la relajación se refleje en la parte trasera del modelo utilizado.

6.5.3 Tiempo de convergencia del operador de desplazamiento

Si bien la relajación que realiza el operador de desplazamiento es con diferencia la alternativa que más tiempo de procesamiento requiere, también es cierto que no es directamente aplicable en su caso la comparación en tiempo absoluto de resolución. La relajación por prioridad deriva en una rápida aproximación del resultado final y un procesamiento retardado de los cambios menores, por lo que es especialmente interesante analizar la evolución del modelo a lo largo de la resolución completa de una interacción.

Para ello se analizará la misma interacción programada que se especificó en el experimento anterior, obteniéndose en este caso datos parciales de procesamiento para comprobar el progreso en cada instante. El modelo, del mismo tamaño anteriormente declarado, representará la densidad con un número natural y presentará una densidad inicial uniforme de 8. La incisión se realizará con velocidad uniforme durante los 2550 primeros milisegundos. En el [Cuadro 6.4](#) aparecen una serie de instantes t de tiempo (expresados en milisegundos) para los que se ha medido: el número de vóxeles procesados desde la medición anterior hasta el instante que se indica en cada fila; la densidad máxima que se ha encontrado en los vóxeles analizados en cada tramo; la velocidad de procesamiento en el intervalo medida en número de vóxeles analizados por milisegundo; y el porcentaje de avance en la relajación respecto del número total de vóxeles procesados. La confrontación del tiempo transcurrido y la densidad máxima analizada en cada intervalo se puede además apreciar en la gráfica de la [Figura 6.16](#).

Este análisis aporta una medida cuantitativa de la priorización de los resultados que se ha atribuido hasta el momento al algoritmo del operador de desplazamiento. La densidad máxima analizada en cada tramo es

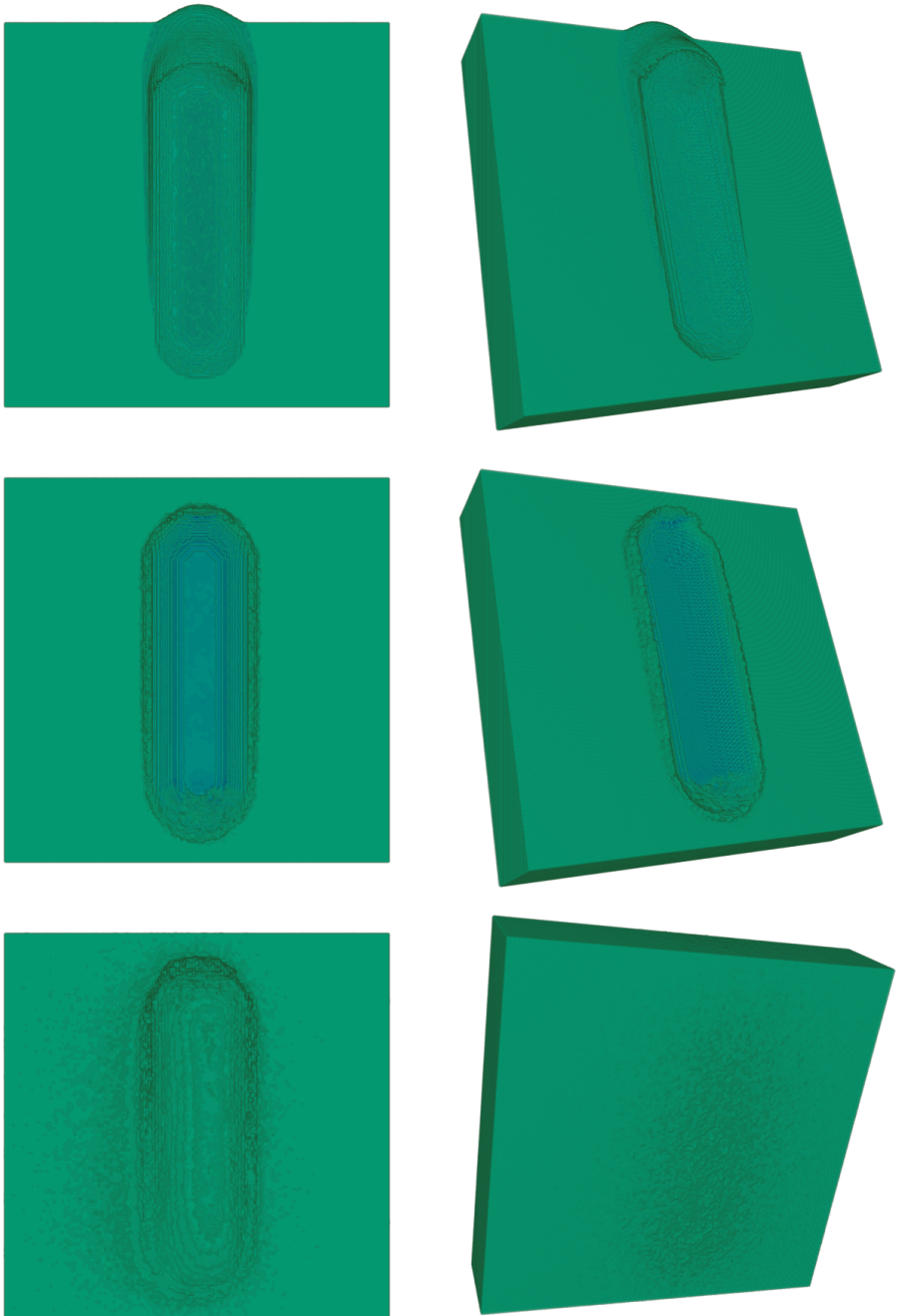


Figura 6.15 – Resultados visuales del experimento de comparación de los tres algoritmos de relajación. De arriba a abajo: algoritmo de dispersión recursiva, operador de suavizado y operador de desplazamiento. La densidad normal se dibuja en azul verdoso, tendiendo al azul los valores superiores de propiedad.

Capítulo 6. Plasticidad

una medida de la compresión existente hasta el instante que se indica en cada fila. Se puede observar que hasta el tercer segundo la densidad máxima encontrada aumenta, pues en paralelo a la relajación la herramienta avanza y sigue comprimiendo el material. El efecto más interesante del experimento es quizá la caída que tiene lugar a partir del tercer segundo, pues en el intervalo $[3000ms, 4250ms]$ no se procesa ningún vóxel con densidad superior a 15, teniéndose por tanto en todo el material como mucho una densidad dos veces superior a la inicial. A partir del quinto segundo se observa una segunda caída que garantiza una compresión máxima de 1.5 veces la densidad original, invirtiéndose gran parte del tiempo restante (intervalo $[12219ms, 2748ms]$) en relajar los vóxeles que quedan al final con una densidad de 9.

En la [Figura 6.17](#), ordenado de izquierda a derecha y de arriba a abajo, se presenta el aspecto del sólido en los instantes $t = 2610ms$ (justo después de finalizar el movimiento de la herramienta), $t = 2950ms$ (un poco antes de que se terminen de procesar los vóxeles más densos), $t = 4970ms$ (poco después de que se relaje todo lo superior a densidad 15), $t = 7920ms$ (poco

Cuadro 6.4 – Resultados del experimento de convergencia del operador de desplazamiento.

Transcurrido (ms)	Vóxeles	Máxima	Velocidad (vóx/ms)	Competado (%)
1000	125772	57	125.77	2.68
2000	247572	56	121.80	5.27
3000	386733	62	139.16	8.24
4250	635806	15	199.26	13.54
5172	801615	12	179.84	17.07
6000	962796	11	194.66	20.51
7360	1181242	11	160.62	25.16
8016	1312823	10	200.58	27.96
9016	1496086	10	183.26	31.86
10578	1718548	10	142.42	36.60
11094	1808473	10	174.27	38.52
12219	1995308	10	166.08	42.50
27438	4695094	9	177.40	100.00

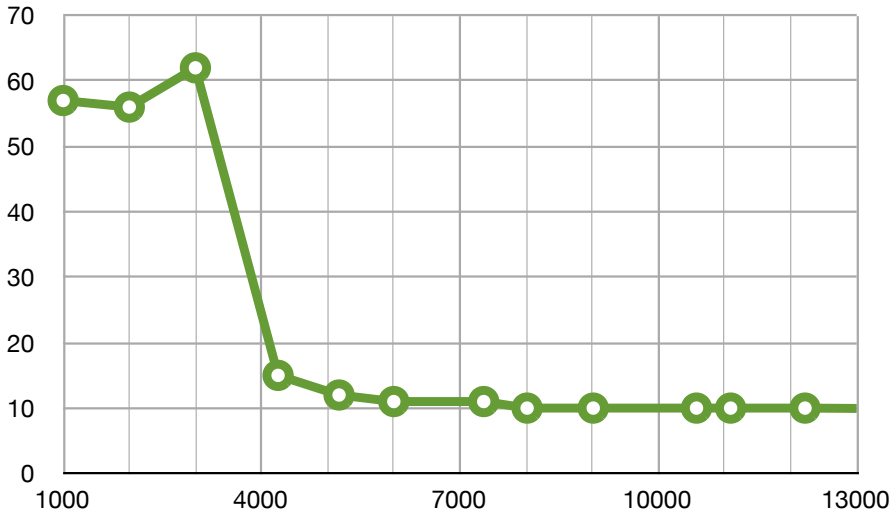


Figura 6.16 – Gráfica con la densidad máxima procesada en el tiempo, donde el eje horizontal indica el tiempo transcurrido en ms y el eje vertical la densidad máxima de los vóxeles procesados en cada intervalo de tiempo. Se representa el intervalo $[0ms, 13000ms]$.

después de terminar con la densidad 11), $t = 12940$ (cuando se ha procesado todo vóxel con densidad 10 o más) y $t = 27438$ (el estado final). Se observa que cuando termina la interacción el sólido está algo comprimido y que la relajación avanza progresivamente. El efecto más notable es quizá el que empieza a aparecer en la cuarta imagen: la expansión de la densidad residual a la superficie. Este fenómeno se agudiza en la quinta imagen ($t = 12940$) y se culmina al final de la relajación, que termina justo cuando afloran en la superficie las últimas sobrecargas que se han ido desplazando por el interior del sólido.

6.5.4 Variación de los tiempos de relajación con el tamaño de la herramienta

Otro aspecto cuyo análisis puede ser interesante es la variación los tiempos de relajación con el incremento del tamaño de la herramienta. Para ello se someterá el algoritmo del operador de suavizado a tres cargas diferentes, que se realizarán sobre la plancha de material definida para los experimen-



Figura 6.17 – Resultados visuales del experimento del análisis de la convergencia del operador de desplazamiento. La densidad normal se dibuja en azul verdoso, tendiendo al azul los valores superiores de propiedad.

tos anteriores con herramientas de radio 8, 16 y 24 vóxeles. Las incisiones tendrán una profundidad proporcional de 6, 12 y 18 vóxeles, suponiendo en cada caso las $3/4$ partes del radio de la herramienta, y seguirán el patrón rectilíneo que se ha usado en los experimentos anteriores.

El [Cuadro 6.5](#) muestra los datos recogidos durante las tres actuaciones. Para cada tamaño de la herramienta se ha anotado el instante en el que se acaba el trazo, y el tiempo transcurrido cuando se han relajado todos los vóxeles con la densidad indicada o mayor. El proceso de relajación termina cuando se han procesado todas las densidades iguales o mayores a 9, llegando por tanto todos los vóxeles a la densidad normal 8. Es interesante la observación de que, en todos los casos, la mayor parte del tiempo se invierte en resolver densidades muy cercanas a la normal: más de la mitad del procesamiento se dedica a mover el material de los vóxeles con densidad 9 o 10. Los datos del [Cuadro 6.5](#) se comparan de forma gráfica en la [Figura 6.18](#). Se puede apreciar allí el crecimiento aproximadamente cuadrático tanto de las resoluciones parciales como del tiempo total de procesamiento, lo cual concuerda con lo que cabe esperar, pues el área de la herramienta crece según el cuadrado del radio y con la misma proporción debe crecer el número vóxeles desplazados en cada paso de la herramienta.

En la [Figura 6.19](#) se muestra la comparación visual de las tres acciones realizadas. Se aprecia cómo crece el efecto de la deformación en la parte trasera del modelo conforme se incrementa el tamaño de la herramienta, y como crece también de forma notable el número de restos (vóxeles con densidad menor que la normal) que afloran por toda la extensión de la superficie al final de la relajación.

6.5.5 Variación de los tiempos de modificación de la superficie con el tamaño de la herramienta

Un factor importante para medir el rendimiento del proceso de deformación es el tiempo que se invierte en modificar la superficie, pues, aunque la relajación se puede realizar de forma retardada, los cambios externos deben reflejarse de forma inmediata para garantizar la sensación de interactividad. Se estudiará por tanto el crecimiento del tiempo requerido para resolver el trasvase inicial de material que se desencadena con el paso de la herramien-

Cuadro 6.5 – Resultados del experimento de la variación de la deformación con diferentes tamaños de herramienta.

Medición	Radio		
	8vóx	16vóx	24vóx
Fin del trazo (ms)	850	2550	8500
Procesado ≥ 15 (ms)	1203	3031	16406
Procesado ≥ 14 (ms)	1203	3453	19141
Procesado ≥ 13 (ms)	1203	3813	24609
Procesado ≥ 12 (ms)	1203	4922	26016
Procesado ≥ 11 (ms)	1203	6891	38766
Procesado ≥ 10 (ms)	1359	12219	74734
Fin relajación (ms)	5156	27438	100672
Vóxeles procesados	963105	4695094	14361201

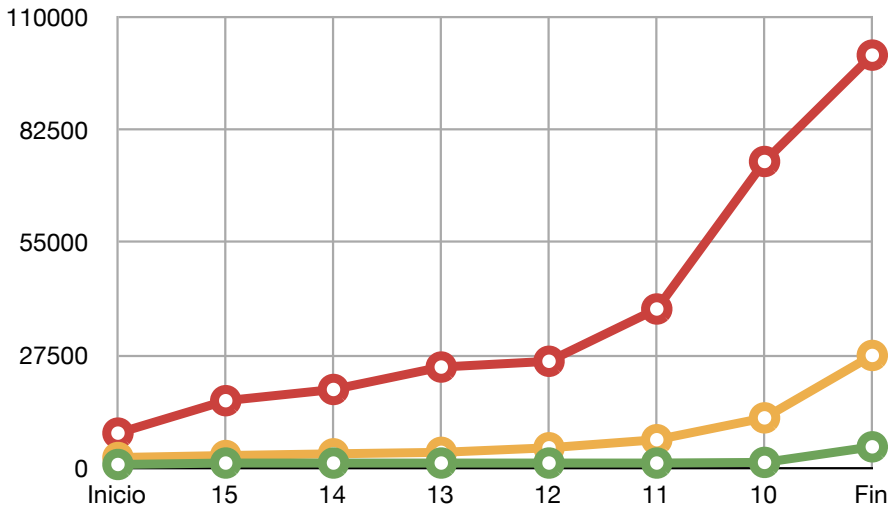


Figura 6.18 – Gráfica con el instante en el que se terminan de procesar todos los vóxeles con densidad igual o superior a la indicada, donde el eje horizontal indica la densidad que se termina de procesar y el eje vertical el tiempo en ms transcurrido hasta ese momento. Las tres líneas representan los tres tamaños de herramienta analizados.

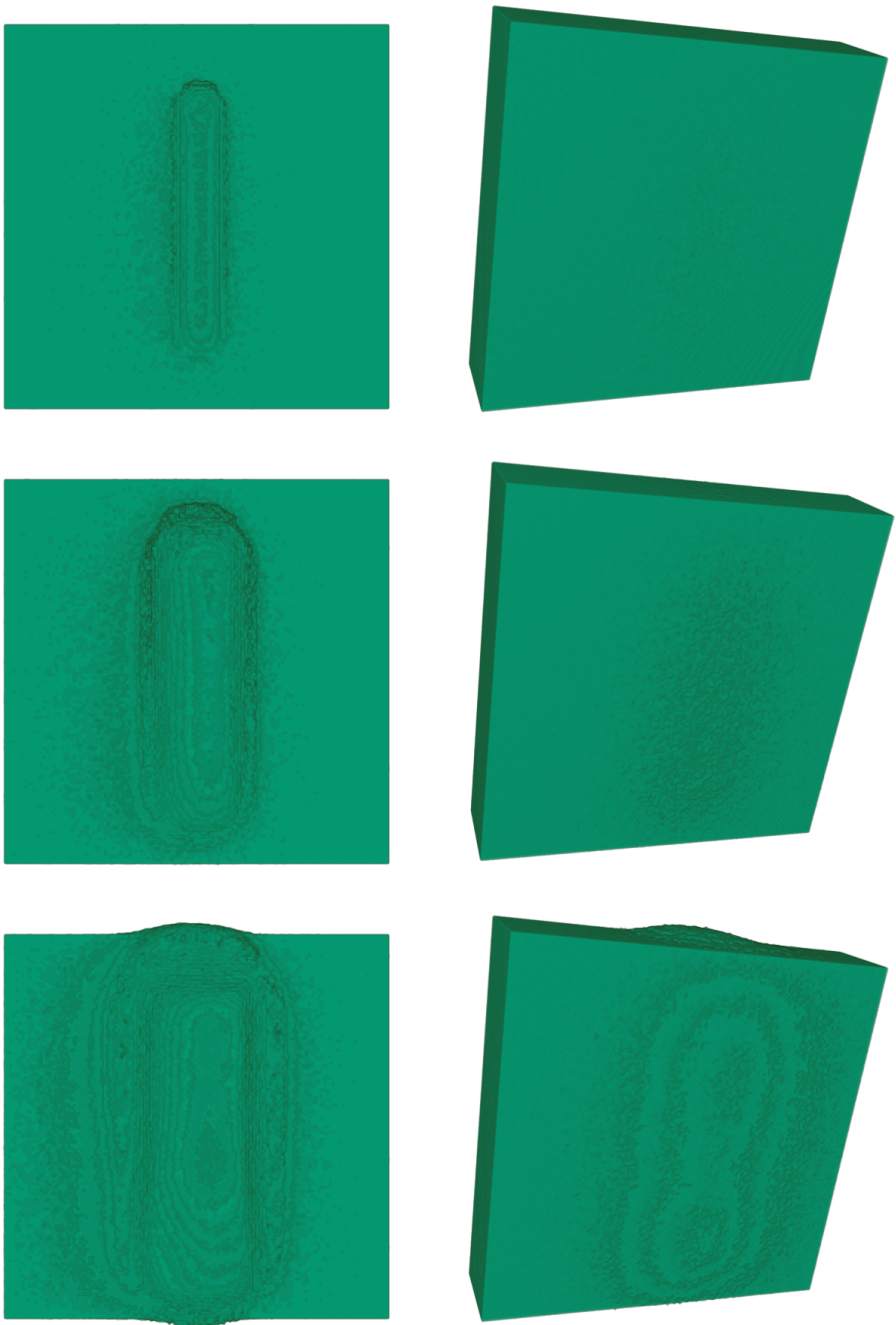


Figura 6.19 – Resultados visuales del experimento de la variación de la deformación con diferentes tamaños de herramienta. De arriba a abajo: efectos de la herramienta de radio 8, radio 16 y radio 24 vóxeles.

Capítulo 6. Plasticidad

Cuadro 6.6 – Resultados del experimento de la variación del tiempo de deformación de la superficie con diferentes tamaños de herramienta.

Radio (vóx)	Tiempo (ms)	Pasos	Tiempo por paso (ms)
4	1	93	0.01
6	62	73	0.85
8	94	86	1.09
10	202	97	2.08
12	375	129	2.91
14	531	137	3.88
16	936	175	5.35
18	1859	181	10.27
20	1797	231	7.78
22	2407	203	11.86
24	4156	164	25.34
26	5422	205	26.45
28	3326	120	27.72
30	6921	144	48.06
32	14266	273	52.26

ta, variando el radio de la misma desde 4 hasta 32 con un incremento por paso de 2 vóxeles.

En el [Cuadro 6.6](#) se muestran los datos recogidos en una interacción de mediano tamaño. La representación gráfica de los datos aparece en la [Figura 6.20](#). Obviando las variaciones que aparecen en la carga computacional por la interferencia de otros procesos, se aprecia un crecimiento cúbico respecto del radio de la herramienta, conforme al incremento del volumen de la misma.

Estas medidas aportan una medición del retardo que se observa de forma experimental a partir de ciertos tamaños de herramienta. Una herramienta de radio 24 vóxeles requiere en cada paso aproximadamente $25ms$ de tiempo de cálculo dedicados exclusivamente a deformar la superficie. Esto implica que en un segundo sólo se pueden resolver 40 pasos de la herramienta, por lo que cualquier movimiento de velocidad superior a 40 pasos por segundo acarreará un cierto desfase inherente. Existe por tanto una cota

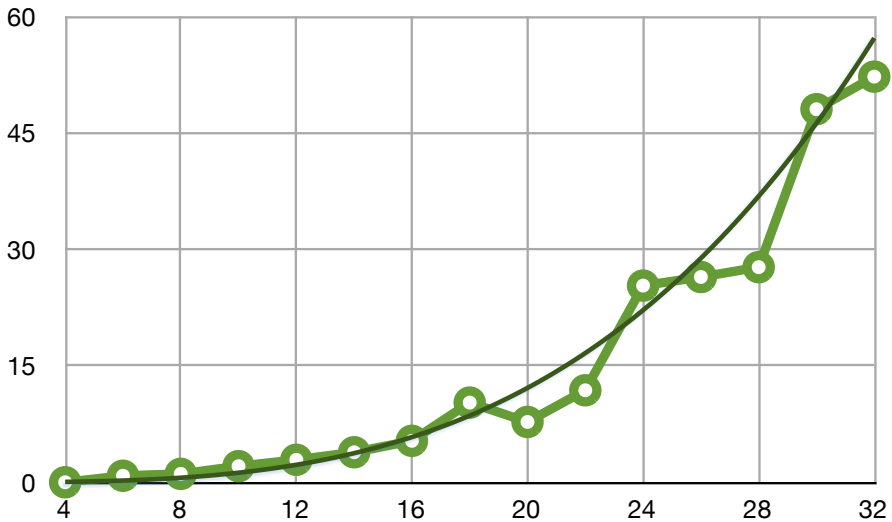


Figura 6.20 – Gráfica con el tiempo invertido para deformar la superficie en cada paso de la herramienta, donde el eje horizontal indica el radio de la herramienta en vóxeles y el eje vertical el tiempo en *ms* necesario para realizar todos los trasvases en la superficie que un movimiento unitario de la herramienta genera. Se dibuja con una línea continua más delgada y oscura la regresión, con un factor de crecimiento potencial $x^{3.2946}$.

superior para la velocidad con la que se puede mover una herramienta de determinado tamaño sin cruzar la frontera de la no inmediatez, siendo esta velocidad máxima precisamente el valor inverso del tiempo medio requerido para resolver un paso.

6.5.6 Cambios topológicos

Si bien todas las interacciones que se han mostrado hasta ahora han sido en mayor o menor medida paralelas a la superficie, es interesante demostrar de forma empírica que el método propuesto puede efectivamente realizar cambios topológicos en el modelo. Como ya se ha expuesto anteriormente, la capacidad de excavar *agujeros* en el sólido debe ser añadida de forma explícita como un caso especial cuando se trabaja con modelos de superficie; no obstante, en el sistema que se ha diseñado, esta capacidad aparece de forma natural.

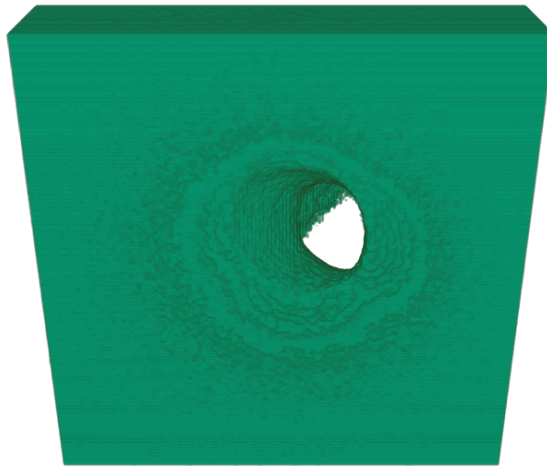
Cuando se realiza un empuje perpendicular a la superficie se inicia un trasvase efectivo de material en dirección a la superficie opuesta. Esto se realiza de manera completamente ordinaria siguiendo el mecanismo de compresión y relajación que se ha descrito. Cuando la herramienta excava lo suficiente como para llegar al límite de la superficie opuesta, los vóxeles se desplazan hacia adelante y hacia los lados creando una pequeña abertura. Si sigue empujando, la hendidura se abrirá cada vez más hacia los lados, expulsando también parte del material hacia afuera en la dirección del avance.

Como ejemplo del funcionamiento de este mecanismo se muestra en la [Figura 6.21](#) una incisión en ángulo de 45° respecto de todos los ejes en la que se ha usado el algoritmo de desplazamiento para la relajación. Se aprecia que, si bien en la cara por la que se realiza la entrada de la herramienta la distribución de los vóxeles es decididamente parecida a las erosiones superficiales que se han mostrado hasta ahora en las imágenes ([Figura 6.21a](#)), a la salida se aprecia un efecto distinto, generado por el mecanismo de expulsión de material a la salida que se acaba de describir. Los vóxeles cercanos a la superficie de salida se ven obligados a *apartarse* hacia los lados cuando la herramienta avanza hacia el exterior del modelo, creándose el efecto *túnel* o *caverna* que se puede observar en la [Figura 6.21b](#).

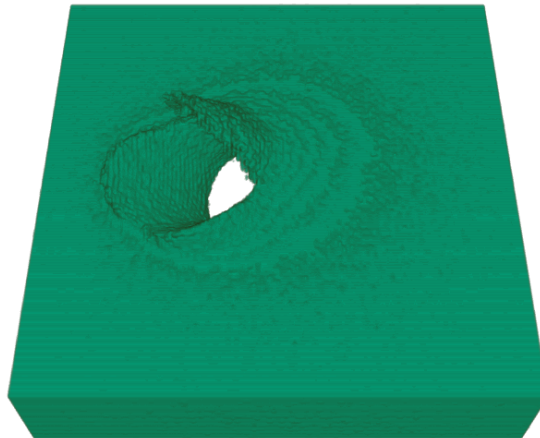
6.6 Conclusiones

La complejidad de la realidad que en este trabajo se pretende simular hace que sea complicado encontrar el equilibrio entre la fidelidad al proceso real, la credibilidad de los resultados y la eficiencia computacional. Durante la creación del método de deformación plástica ha sido necesario renunciar a la aplicación estricta de algunos de los principios mecánicos descritos por la teoría, reduciéndose pues el cumplimiento de la misma al cómputo de la fuerza, a la utilización del modelo de tramos elásticos y elastoplásticos, y, en algunos casos, a la modificación de la superficie.

La propuestas hechas finalmente usan la *aproximación* como principio de diseño. Las heurísticas de deformación se han inspirado en todo caso en los resultados esperados para una simulación formal y exacta, ofreciendo



(a) Entrada de la horadación



(b) Salida de la horadación

Figura 6.21 – Cambios topológicos conseguidos con el operador de desplazamiento orientado, atravesando una plancha de material con una incisión en 45° . En (a) se aprecia la parte de la plancha por la que entra la herramienta, mientras que (b) muestra el efecto creado en la salida del túnel, en el que parte del material sobresale creando un efecto de *caverna*.

como contrapartida a su limitada exactitud una forma ágil y progresiva para simular deformaciones permanentes.

En los experimentos se ha corroborado el carácter *interactivo* de la técnica de deformación creada, pudiéndose realizar deformaciones de mediano tamaño en modelos cúbicos de 256 vóxeles de lado. Las distintas soluciones aportadas permiten un abanico de efectos que pueden combinarse para simular uno u otro material. Se ha puesto además de manifiesto la posibilidad de producir *cambios topológicos* en el sólido. Mientras que la creación de *agujeros* cuando se trabaja con representaciones de superficie es una tarea tediosa que implica cálculos específicos de colisión y ampliación de las superficies conectadas, las técnicas aquí propuestas abrazan este efecto de forma natural.

Sobre el endurecimiento cabe también destacar que, al no estar reflejada función de fluencia plástica $\sigma_f(\alpha\sim)$ que representa el límite del comportamiento elástico en los valores de propiedad de los vóxeles, no es posible en principio *recordar* los parámetros de endurecimiento de una interacción a otra. La variable de endurecimiento $\alpha\sim$ se reinicia cada vez que se abandona la interacción actual, usándose además valores predeterminados para el límite elástico inicial σ_e y el parámetro de endurecimiento H' , los cuales deben obtenerse de forma experimental.

Se observa que, si bien los resultados no coinciden necesariamente con la configuración que se hubiera podido obtener con el desarrollo formal del problema, el efecto conseguido no deja de ser *creíble* de forma macroscópica, lo cual es una conclusión cuanto menos satisfactoria para un método que, por definición, no deja de ser aproximado.

Capítulo 7

Elasticidad

El estudio de la plasticidad que se ha realizado en el [Capítulo 6](#) deja pendiente la representación visual de la elasticidad. Para atacar la resolución de las deformaciones permanentes se separaron los efectos visuales elástico y plástico, con la intención de que el análisis fuera más sencillo, y bajo la pretensión de que, al ser las deformaciones elásticas recuperables no sería necesario reflejar estos cambios en la distribución del material en los vóxeles, pudiendo así prepararse algún mecanismo que trabajara en un nivel superior independiente.

Como asistencia en la búsqueda de una vía para representar los movimientos recuperables de las partículas del sólido aparece la herencia de los estudios del cálculo de la fuerza y de la plasticidad. Ya se conocen las fuerzas estimadas que generan la carga elástica, la carga elastoplástica y la descarga, conociéndose además con exactitud el punto en el que termina el comportamiento elástico y comienza la deformación permanente. Se puede prever también que calcular la solución exacta del problema será una ardua y poco eficiente tarea, pues no existe para el comportamiento elástico ninguna ventaja que lo haga más sencillo de resolver que el de las deformaciones plásticas.

Para bordear el planteamiento formal se propondrá también un método heurístico que imite los resultados previstos en un proceso elástico real. No hay que perder de vista que el proceso es totalmente reversible, lo cual quiere decir que en cualquier momento una anulación de la fuerza ejercida

por la herramienta deberá dejar el sistema exactamente en el mismo estado en el que se encontraba antes de comenzar la interacción. Se explicarán pues en este capítulo los pasos que llevarán a la estimación de la parte elástica de la deformación de un sólido elastoplástico, culminándose con esto la construcción del entorno de simulación.

7.1 Aislamiento del efecto elástico

Paralelamente a lo que se determinó cuando se estudió la deformación permanente, se analizará en este capítulo la visualización de los movimientos elásticos de forma aislada. Será necesario no obstante tener en cuenta que posteriormente habrá que unir la deformación elástica con la plástica.

El tramo elástico termina cuando se alcanza la tensión de fluencia, después de lo cual la deformación recuperable irá aumentando proporcionalmente con cada incremento de tensión en la zona plástica. En caso de detectarse un incremento negativo en la estimación de la fuerza que causa la deformación el modelo deberá regresar paulatinamente al estado original. Se puede sin embargo abstraer este límite y observar un comportamiento puramente elástico, pues la conexión con la teoría incremental de la plasticidad podrá hacerse de forma automática activando los cálculos descritos para la tensión de fluencia y el endurecimiento sobre la configuración elástica que se describieron en el [Capítulo 6](#).

Entendiéndose en este contexto la **deformación** como el proceso de cambio temporal de la estructura de datos que representa el material, se centrarán los esfuerzos ahora en diseñar una heurística que efectúe los cambios pertinentes en el modelo para que se pueda apreciar de forma visual la mencionada transformación.

7.2 Viabilidad del modelo teórico

La aplicación directa de la teoría de la elasticidad plantea exactamente los mismos problemas que el cómputo exacto de la deformación permanente. No en vano es necesario solucionar la parte elástica de la deformación *antes* incluso de poder considerar los desplazamientos permanentes de material.

Una resolución formal de la deformación acarrea exactamente los mismos problemas que ya se analizaron en la [Sección 5.2](#) durante la adaptación de la teoría al entorno háptico y en la [Sección 6.2](#) cuando se discutió lo propio para los cambios permanentes. La primera contrariedad es que no se conoce con exactitud la fuerza ejercida por el usuario, por lo que no puede conseguirse una noción exacta del campo de tensiones $\sigma(\mathbf{x}, t)$ ni de las deformaciones $\epsilon(\mathbf{x}, t)$. Por otra parte se tiene que incluso superando este obstáculo la resolución del sistema de ecuaciones en derivadas parciales plantearía un coste computacional difícil de asumir, debido principalmente a las restricciones de tiempo y la cantidad de nodos a procesar.

Se actuará por tanto bajo la misma premisa que se usó para la estimación de la fuerza y de la deformación plástica, y es la certeza de que la superficie se mueve acorde a los movimientos de la herramienta una vez la segunda contacta la primera, lo cual derivará irremediabilmente, una vez más, en un procedimiento aproximado.

7.3 Aproximación heurística de la deformación elástica

Vuelven a ser el campo de desplazamientos estimado $\mathbf{u}_{\tilde{S}}$ generado por el contacto de la herramienta y la fuerza $\vec{f}_{\tilde{S}}$ que de él se deriva los únicos puntos de apoyo a los que se puede recurrir, si se tiene en cuenta lo anteriormente expuesto.

Al ser todo el procedimiento reversible no se antoja muy práctico reflejar los cambios en los valores de propiedad tal y como se hizo para la resolver la plasticidad. Esto requeriría en principio trazar los cambios efectuados en la distribución del material con cada cambio de los desplazamientos estimados $\mathbf{u}_{\tilde{S}}(\mathbf{x}, t)$, para poder deshacerlos al detectarse la reducción de la tensión. Otra alternativa sería mantener dos *copias* del volumen y aplicar todos los cambios de forma conjunta con cada cambio, método que si bien ha sido aplicado satisfactoriamente en otros trabajos que simulan elasticidad como el de Aldrich et al. [[APH11](#)], supondría aplicar todos los desplazamientos a todos los vóxeles implicados en cada ciclo de refresco de la imagen. Esta alternativa terminaría irremediabilmente con la línea de método progresivo

que se instauró en la resolución de la plasticidad, sin tener en cuenta que cualquiera de las dos opciones expuestas empeoraría la ya de por sí considerable carga computacional que soporta el algoritmo de deformación.

Sea como fuere aparece la necesidad de crear un recubrimiento o capa adicional de información sobre el modelo volumétrico que se ha propuesto pues, se elija la solución que se elija, todo cambio reversible necesita por definición el historial de modificaciones para poder deshacerse.

Se requiere por tanto aclarar dos cuestiones principales relacionadas con el diseño de la solución heurística planteada:

1. En primer lugar será necesario idear una *superestructura* que trabaje sobre el modelo volumétrico y que permita representar cambios espaciales fácilmente recuperables, de forma que se puedan ejecutar las deformaciones elásticas de forma independiente y con el mínimo coste computacional posible.
2. En segundo lugar habrá que diseñar un método de estimación de los desplazamientos elásticos. Dicho método tendrá que coordinar los cambios de la estructura adicional diseñada con los desplazamientos estimados $\mathbf{u}_{\tilde{S}}(\mathbf{x}, t)$ de los que se tiene constancia, inspirándose en los estudios formales sobre deformación como el ya mencionado de Prevost [Pre84], o incluso alguno de los múltiples que se pueden encontrar sobre deformaciones elásticas puras como los de Bathe et al. [KLB93, SB88].

En lo que sigue se describirán los pasos seguidos para resolver estas dos tareas, una vez completadas las cuales se podrá crear un algoritmo de estimación que cumpla las premisas que hasta ahora se han descrito.

7.3.1 Superestructura de deformación elástica

Habiéndose comprobado que la opción de reflejar los desplazamientos recuperables en los vóxeles desembocaría en una sobrecarga indeseable para el procedimiento de deformación, se puede buscar una alternativa que actúe en un nivel superior, deformando el espacio que envuelve al modelo volumétrico sin intervenir para nada en la distribución de material.

Existe de hecho una técnica ampliamente conocida que permite deformar el espacio y todo lo contenido en él, y es la llamada *Free-Form Deformation* que crearon Sederberg y Parry [SP86] y que se revisó en los trabajos relacionados en el [Capítulo 2](#). Mediante un modelo FFD que englobe al modelo voxelizado que representa el material se puede transformar el espacio interno y, consecuentemente, la estructura que este contiene según la configuración de los puntos de control (tal y como se ilustró en la [Figura 2.5](#)).

Esta idea de *volumen transformable* ya fue usada por Winter [Win02] para realizar operaciones espaciales sobre modelos volumétricos. Utilizando la noción de espacio deformable en la que se basó Serderberg [Bar84], Winter define lo que denominó *funciones de transferencia espacial* [CSW⁺03], entidad que permitía aplicar una transformación $\psi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ a todos los puntos del espacio y desplazándolos de forma arbitraria:

$$P' = \psi(P) \tag{7.1}$$

donde $P \in \mathbb{R}^3$ es un punto del espacio y $P' \in \mathbb{R}^3$ es la localización del punto P tras aplicarle la transformación inducida por ψ . De esta forma pueden conseguirse efectos tales como compresión, extensión, o torsión en el volumen eligiendo la transformación ψ indicada en cada caso.

Mezclando los dos conceptos se creará un envoltorio deformable conformado por un conjunto de $l \times m \times n$ puntos X_{ijk} ordenados en rejilla regular en el espacio tal y como se determinó en la ecuación 3.18. Esta estructura, que ya fue descrita en la [Sección 3.3](#), conformará una partición de prismas de base rectangular que *recubrirá* completamente al modelo voxelizado (ver [Figura 7.1](#)). Los puntos X_{ijk} se deformarán mediante una función de transferencia espacial ψ que creará una transformación del espacio según el desplazamiento elástico estimado.

Al igual que sucede al mover los puntos de la rejilla de control de un modelo Free-Form Deformation, cualquier movimiento de los puntos X_{ijk} redefinirá de forma implícita los puntos interiores de los paralelepípedos por interpolación, consiguiéndose una deformación efectiva para todos los puntos X del modelo con la aplicación de la transformación a los puntos X_{ijk} de la rejilla.

Puede observarse que esta definición trabaja con dos sistemas de coor-

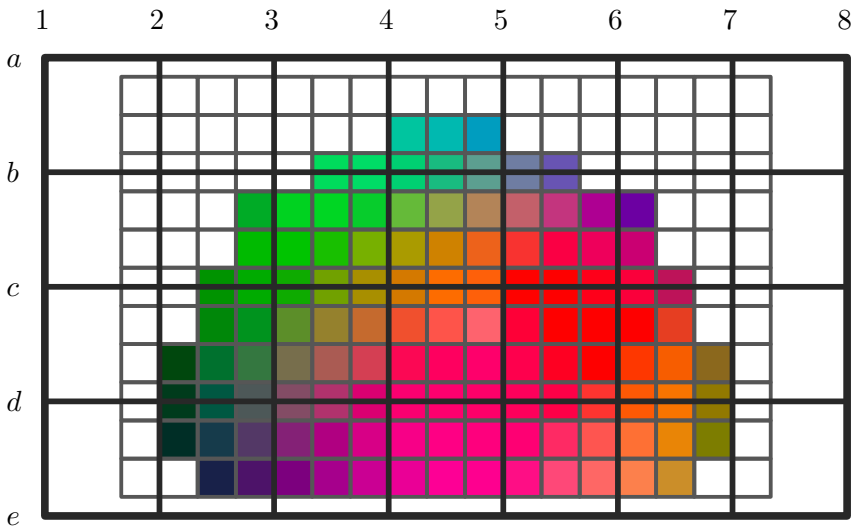


Figura 7.1 – Versión plana de una superestructura de deformación. La rejilla del modelo de muestras se ilustra en gris, mientras que los puntos X_{ij} se encuentran en la intersección de las líneas negras. Los puntos X_{ij} no tienen por qué estar necesariamente alineados con las muestras m_{ij} ni con la frontera intervoxel. Además no tiene por qué coincidir la dimensión de la estructura voxelizada con la del recubrimiento en ninguna de las direcciones.

denadas concurrentes: el de los puntos X que se corresponden al espacio del modelo de vóxeles y el de los puntos X' de la nueva organización del espacio tras aplicar la transformación ψ .

7.3.2 Estimación de la deformación elástica

Teniendo a la disposición una estructura que se puede deformar libremente, queda entonces encontrar una estimación para los desplazamientos elásticos $\mathbf{u}_e^{\sim}(\mathbf{x}, t)$ que se parezca lo más posible al proceso real de deformación de un sólido elástico.

En la búsqueda de un efecto visual lo más acertado posible se encuentra, mediante la observación directa de un sólido elástico, que con el empuje realizado sobre la superficie no sólo se mueve la zona presionada, sino que además se *arrastra* parte de la superficie circundante, y en algunos casos y dependiendo del tamaño se deforma el cuerpo entero. Este efecto es en realidad parcialmente reconocible también en la deformación plástica (ver

Figura 5.4 de los resultados de Prevost) si bien no pudo reflejarse con mucha exactitud cuando se realizó la estimación, debido en parte a que la naturaleza de los algoritmos elegidos realizaba un desplazamiento progresivo del sólido que hace difícil controlar la orientación de la deformación de forma macroscópica.

Este efecto se corrobora en estudios formales como los de Bathe et al. [SB88], que muestran evidencias en este sentido tras resolver el problema de forma numérica. En la Figura 7.2 se puede apreciar la deformación de una pieza elástica de goma que está anclada en su lateral izquierdo, mientras que se aplica una presión en el lado opuesto (Figura 7.2a). Se comprueba en la Figura 7.2c que la malla original se deforma en casi toda su dimensión, atenuándose el desplazamiento vertical infligido conforme los nodos de la malla se acercan a la zona de anclaje.

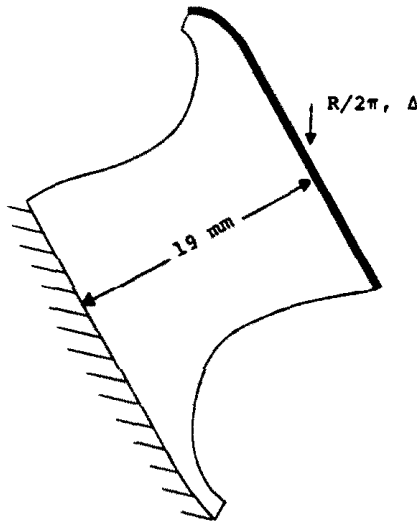
En el sistema virtual que se ha construido no existen los puntos de anclaje, si bien puede asumirse que el sólido está apoyado en la zona opuesta a la que se está actuando, lo cual es una asunción que entra dentro de lo razonable, teniendo en cuenta que el sólido debe estar apoyado en algún sitio para poder ejercer resistencia. Se puede procurar entonces una estimación de la deformación elástica \mathbf{u}_e^\sim de largo alcance que simule el movimiento de la pieza anterior definiendo una transformación ψ que se adecue al principio descrito, en el que se produce un desplazamiento aproximadamente homogéneo que se atenúe con la distancia al punto de acción.

Se formalizará entonces la transformación ψ con la función $e : \mathbb{R}^9 \rightarrow \mathbb{R}^3$ que se definió en 3.17:

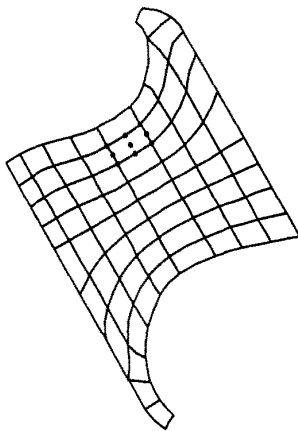
$$\mathbf{X}' = e(\mathbf{X}, \vec{c}_0, \vec{u}^\sim) \tag{7.2}$$

a la que se conocerá por *transformación espacial elástica*, y que se usará para desplazar los puntos X_{ijk} de la rejilla a una nueva posición deformada X'_{ijk} , según la variación del desplazamiento estimado \vec{u}^\sim y el punto de incisión de la herramienta en el volumen \vec{c}_0 .

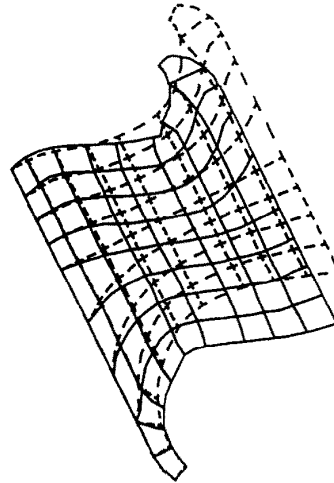
Teniendo en cuenta la especificación que se ha hecho para la estimación de la deformación, se puede definir la transformación espacial elástica como la parte proporcional del desplazamiento estimado \vec{u}^\sim correspondiente a



(a) Esquema de una pieza de goma



(b) Malla creada



(c) Deformación de la malla

Figura 7.2 – Simulación de la deformación elástica de una pieza de goma de Bathe et al. [SB88]. Arriba, la representación esquemática de la pieza, que está sujeta por la parte derecha, y sobre la que se ejerce la fuerza indicada con la flecha sobre la superficie de la izquierda. Abajo a la izquierda, malla con la que se simula la deformación. Abajo a la derecha, deformación conseguida. La malla original se muestra rayada, y la deformada con líneas sólidas.

cada punto X según la cercanía al borde de la herramienta:

$$e(X, \vec{c}_0, \vec{u}^\sim) = X + \min\left(\frac{r}{\|X - \vec{c}_0\|}, 1\right) \vec{u}^\sim \quad (7.3)$$

donde r es el radio de la herramienta, constante, y donde se observa que el desplazamiento máximo se consigue en los puntos tangentes a la herramienta, atenuándose este según la distancia a la superficie de la misma. Se sesga además el factor de reducción cuando supera la unidad, para evitar que los puntos del interior de la herramienta se desplacen más allá de los de la superficie. El efecto conseguido con la función e se representa de forma esquemática en la [Figura 7.3](#). Se comprueba a la vista de la imagen que, dado que en la posición \vec{c}_0 deja por definición a la herramienta en una posición tangente a la superficie, solamente habrá un punto X que sufra el desplazamiento máximo \vec{u}^\sim . Un cálculo exacto de cómo se modifica el espacio con la función e se puede apreciar en la [Figura 7.4](#).

Con todo lo anterior se puede formalizar el campo de desplazamientos elásticos \mathbf{u}_e^\sim en función de las partículas \mathbf{x} del sólido, identificadas con notación tensorial:

$$\mathbf{u}_e^\sim(\mathbf{x}, t) = e(\mathbf{x}, \vec{c}_0, \vec{u}^\sim) - \mathbf{x} \quad (7.4)$$

esto es, el vector resultante de medir la distancia entre cada punto \mathbf{x} del modelo y su homólogo transformado. Utilizando la definición de la transformación elástica de [7.3](#) se puede expresar campo como:

$$\mathbf{u}_e^\sim(\mathbf{x}, t) = \min\left(\frac{r}{\|\mathbf{x} - \vec{c}_0\|}, 1\right) \vec{u}^\sim \quad (7.5)$$

quedando correctamente definida con esto la deformación elástica estimada que se había propuesto.

7.4 Resultados

La superestructura de deformación elástica se puede incorporar al sistema diseñado con poco esfuerzo si se planifica correctamente su integración en la visualización del modelo volumétrico, pues no deja de ser una transformación del espacio similar a las que se realizan de forma ordinaria al

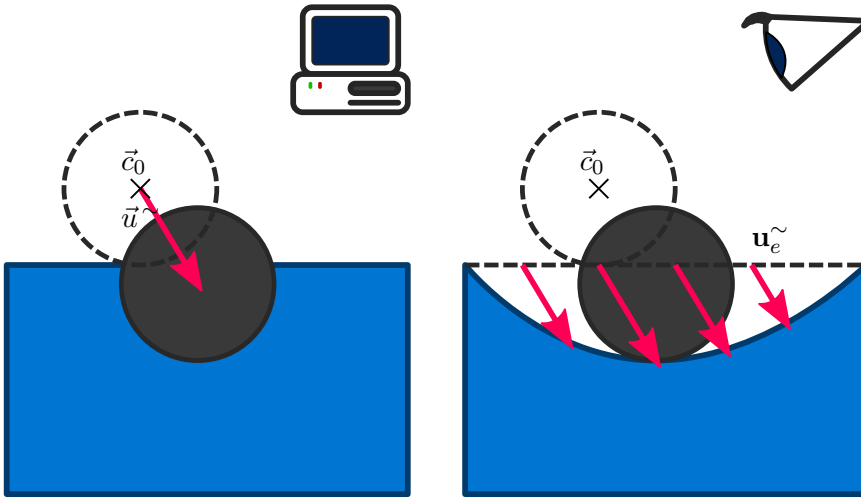


Figura 7.3 – Representación esquemática bidimensional de la transformación elástica sufrida por el material con la aparición de un desplazamiento estimado \tilde{u} . A la izquierda, la representación interna del modelo volumétrico. A la derecha, modelo transformado que se presenta al usuario, en el que los puntos se desplazan según el campo de desplazamiento elástico estimado \tilde{u}_e .

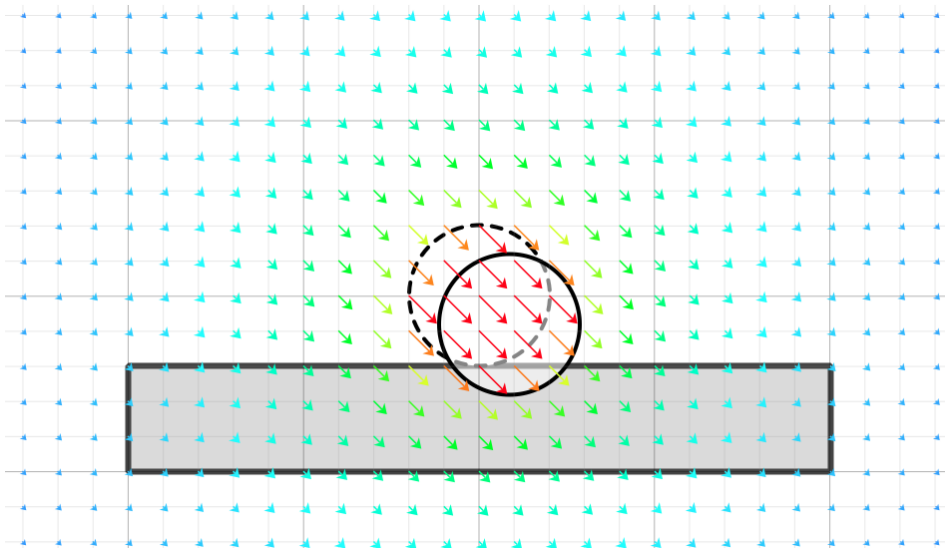


Figura 7.4 – Cálculo de la deformación espacial e en dos dimensiones para un desplazamiento estimado de $\tilde{u} \equiv (0.8, -0.8)$, con una rejilla de lado unitario y una herramienta de radio 2. Se muestra el campo de desplazamientos conseguido para los puntos de la rejilla, usándose un gradiente de color para una mejor apreciación de la variación del módulo los vectores del campo.

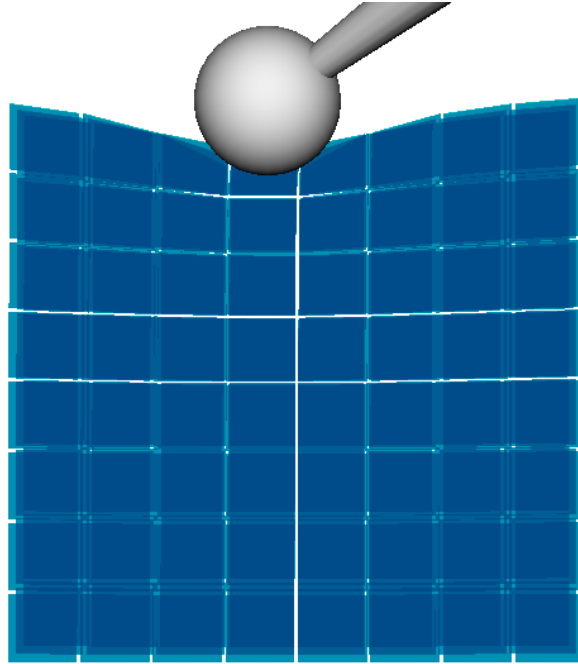


Figura 7.5 – Superestructura de deformación elástica organizada en planos. Se aprecia en la imagen que el modelo (la misma plancha de material de la [Figura 6.13](#)) está representado con $n = 3$ planos, cada uno de ellos con $l = 9$ puntos de anchura y $m = 9$ puntos de altura, creándose por tanto 8×8 polígonos o parches por plano. Se observa que, al calcularse la intersección de los planos con el volumen V antes de desplazar los puntos X_{ijk} , se crea la ilusión de deformación al aplicar el desplazamiento elástico.

componer las vistas en la computadora. Cabe observar, a la vista de los métodos de visualización expuestos en la [Sección 1.2.2.5](#), que la estructura poligonal que se requiere para implementar la visualización basada en *cortes en profundidad* se complementa especialmente bien con la superestructura elástica descrita si se agrupan los $l \times m \times n$ puntos del recubrimiento en n planos apilados en profundidad para que coincidan con los cortes necesarios para componer la vista. Cada uno de los cortes se dividirá además en una partición de $(l - 1) \times (m - 1)$ polígonos, estando las divisiones de los mismos definidos por los $l \times m$ puntos X_{ij} de cada plano k . Se calcula entonces la intersección de todos los polígonos con el modelo volumétrico, consiguiéndose el efecto de profundidad al apilar las *rebanadas* extraídas. Esta organización se puede apreciar en la [Figura 7.5](#), donde se muestra una

plancha de material con $n = 3$ cortes en profundidad, $l = 9$ puntos de anchura y $m = 9$ puntos de altura, a la que se ha añadido un espacio vacío en las líneas que unen los 9×9 puntos X_{ij} de cada plano para una mejor comprensión de la estructura.

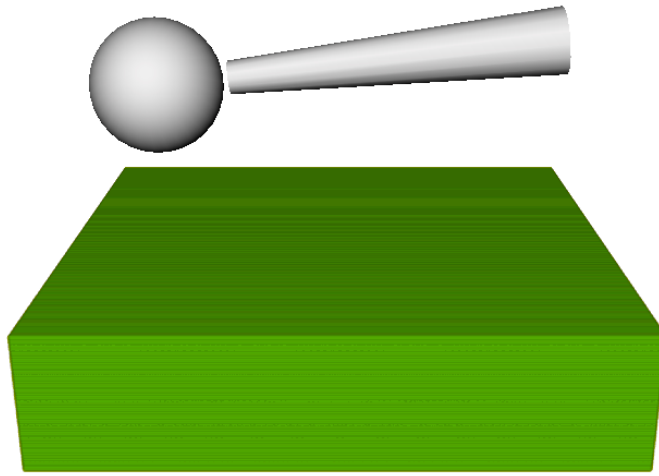
Teniendo esto en cuenta el algoritmo de cálculo de la deformación elástica se reduce pues a calcular la posición deformada de los puntos X_{ijk} mediante la transformación espacial elástica e , consiguiéndose pues el efecto que se persigue ejecutando tres sencillos pasos:

1. **Comprobación del estado de deformación elástico.** Se obtiene el desplazamiento estimado \vec{u}^{\sim} y el punto de incisión de la herramienta \vec{c}_0 .
2. **Transformación de la superestructura de deformación.** Se calcula la deformación elástica haciendo $X'_{ijk} \leftarrow e(X_{ijk}, \vec{u}^{\sim}, \vec{c}_0)$ para todos los puntos X_{ijk} de la superestructura de deformación.
3. **Envío de los resultados.** Se muestra en pantalla la superestructura dibujando todos los polígonos implícitamente definidos con los puntos X'_{ijk} .

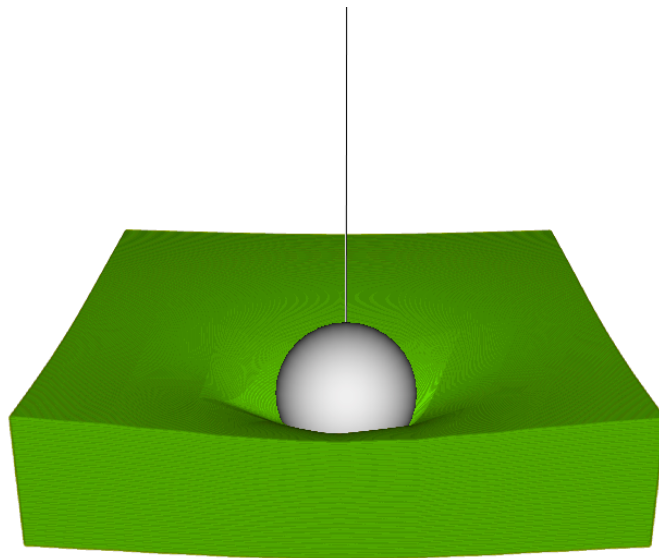
Nótese que en caso de no interacción (esto es, $\vec{u}^{\sim} \equiv 0$) se tiene que $X_{ijk} = X'_{ijk}$ para todos los puntos de la estructura, mostrándose el volumen V inalterado. Esto permite introducir el algoritmo de deformación elástica que se ha descrito en el ciclo normal de dibujado, deformándose automáticamente la doble estructura vóxeles—recubrimiento en cuanto se detecta una colisión de la herramienta con el sólido.

Los detalles concretos de la implementación se ofrecen en el [Apéndice A](#). De igual modo se describe allí la relación de la superestructura elástica con el método de visualización de volúmenes que se utiliza.

Los resultados visuales del algoritmo se pueden apreciar en la [Figura 7.6](#), en la que se presiona con la herramienta sobre un bloque de material elástico. Los puntos de la superestructura de deformación se desplazan con la aparición del desplazamiento estimado inducido por el movimiento del dispositivo háptico, arrastrando con ellos al modelo volumétrico que se ha asociado a los polígonos de la misma. En la [Figura 7.7](#) se puede apreciar

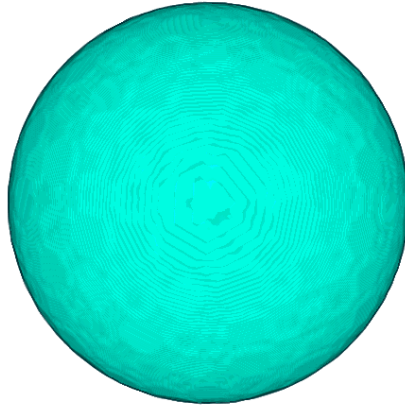


(a) Plancha de material sin deformar

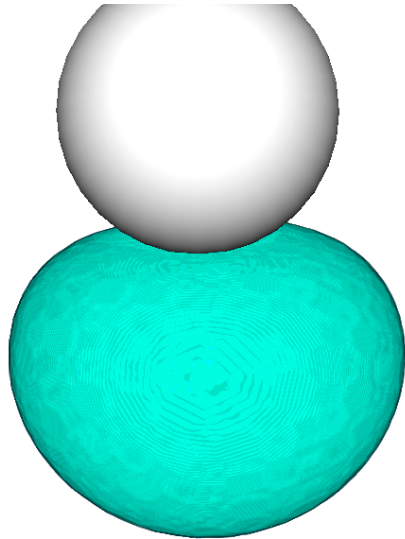


(b) Plancha de material deformado

Figura 7.6 – Deformación elástica estimada de una plancha de material, representada con un modelo cúbico de 256 vóxeles de lado. Arriba, el modelo intacto, en un estado de no interacción. Abajo, la herramienta presiona hacia abajo en una dirección normal a la superficie (se muestra el vector con la fuerza generada como una línea gris).



(a) Esfera sin deformar

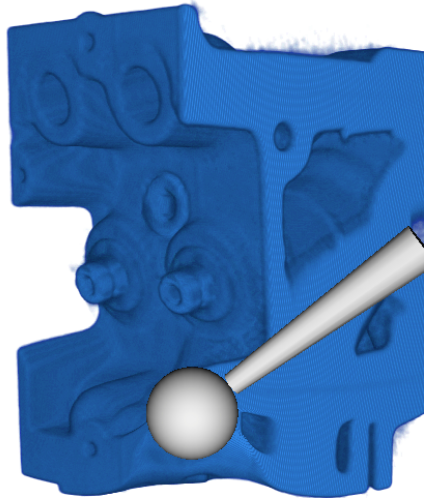


(b) Esfera deformada

Figura 7.7 – Deformación elástica estimada de un cuerpo esférico, simulada con un modelo cúbico de 128 vóxeles de lado. Arriba, el modelo intacto, en un estado de no interacción. Abajo, la herramienta presiona hacia abajo para deformar el modelo.



(a) Motor sin deformar



(b) Motor deformado

Figura 7.8 – Deformación elástica estimada del modelo de un motor, simulado con un modelo cúbico de 256 vóxeles de lado. Arriba, el modelo intacto, en un estado de no interacción. Abajo, la herramienta presiona la zona inferior para deformar el modelo. (Modelo cortesía de General Electrics USA).

la deformación de un cuerpo esférico, mientras que en las [Figura 7.8](#) se muestra la deformación de un modelo algo más complejo.

Para los resultados que se muestran se ha usado una superestructura con $k = 512$ planos de fondo, $i = 16$ puntos de anchura y $j = 16$ puntos de altura. Esto crea un total de $16^2 \times 512 = 131072$ puntos X_{ijk} cuyo desplazamiento tiene que ser explícitamente calculado, con $(16-1)^2 \times 512 = 115200$ parches cuadrados que muestran el volumen deformado mediante interpolación.

7.5 Experimentos

Una vez construido el método de deformación elástica se abre la puerta a su validación práctica, poniendo a prueba su desempeño al enfrentarse a modelos de distintos tamaños, así como comprobando la variación del rendimiento al cambiar la configuración de la superestructura elástica. Es también necesario comprobar la compatibilidad con el método de deformación plástica que se diseñó en el [Capítulo 6](#), de forma que se permita una interacción conjunta que refleje la operación con materiales que presenten tanto propiedades elásticas como plásticas. En esta sección se mostrarán los resultados de las pruebas que se han llevado a cabo con la intención última de evaluar las cuestiones que se han descrito.

7.5.1 Sobre el rendimiento de la deformación elástica

Es obvio que tanto la calidad de los resultados como la cantidad de tiempo requerida para calcular la deformación de la estructura completa dependen directamente del número de puntos usados en la superestructura. Experimentalmente se ha podido comprobar que un número bajo de parches por plano no implica necesariamente malos resultados si se compensa con una densidad alta de planos de fondo (tal y como se puede comprobar en las imágenes de las [Figuras 7.6](#), [7.7](#) y [7.8](#)). Esto garantiza además una representación más detallada del modelo volumétrico, puesto que su visualización se basa en la superposición de dichos planos. Para comprobar el rendimiento del método de deformación elástica se medirán las frecuencias de actualización para las siguientes variaciones del sistema:

Tamaño del volumen Se medirá el desempeño con volúmenes cúbicos de lado 128, 256, 384 y 512 vóxeles.

Número de cortes Se usarán dos configuraciones distintas para la superestructura de deformación, con 256 y 512 planos de fondo.

Número de parches por corte Se variará la configuración de la superestructura además en el número de parches, poniéndose a prueba una distribución de 8×8 parches y otra de 16×16 .

Influencia de la superestructura Para comprobar la repercusión directa que tiene la superestructura en el rendimiento se medirá, para todas las combinaciones de las variables anteriores, la frecuencia de actualización de la pantalla con la estructura y sin la estructura. De este modo se podrá confrontar la caída de rendimiento generada exclusivamente por la presencia de la superestructura a la reducción en el mismo derivada de otros factores externos.

El tamaño de la herramienta no afecta a los tiempos de cálculo de la deformación elástica, pues el radio es únicamente una constante en la ecuación que calcula el desplazamiento de los puntos (ver ecuación 7.3).

Cuadro 7.1 – Resultados del experimento sobre el rendimiento de la deformación elástica con distintas configuraciones del sistema.

Modelo	Configuración superestructura			
	$8 \times 8 \times 256$	$8 \times 8 \times 512$	$16 \times 16 \times 256$	$16 \times 16 \times 512$
128^3	21/21	21/21	21/22	10/21
256^3	20/21	19/21	11/21	10/21
384^3	11/11	11/12	10/11	7/11
512^3	7/7	6/7	6/7	6/7

Las mediciones realizadas se pueden apreciar en el Cuadro 7.1. Para cada combinación de un tamaño de volumen y una distribución de la superestructura determinada se muestran dos cifras: la primera es la frecuencia de actualización de la vista dibujando la superestructura completa, la segunda es la frecuencia pintando únicamente los cortes de fondo sin los parches de

cada plano. Todas las frecuencias se han redondeado al entero más bajo y expresan en número de frames generados por segundo.

A la vista de estos dos datos se puede concluir para empezar que la reducción drástica del rendimiento que aparece para los modelos de 384 y 512 vóxeles cúbicos no se debe a la presencia de la superestructura, sino a las limitaciones del hardware a la hora de manejar modelos de ese tamaño. También aparecen diferencias importantes cuando crece el número de polígonos que conforman la superestructura. No es apenas perceptible el cambio de la estructura de $8 \times 8 \times 256$ a $8 \times 8 \times 512$ polígonos, pero sí supone una reducción drástica de la frecuencia aumentar el número de parches de 8×8 a 16×16 . Al multiplicarse por cuatro los polígonos presentes cae dramáticamente el rendimiento en prácticamente todas las configuraciones.

Existe pues un límite impuesto por el hardware en el detalle con el que la deformación elástica se puede representar, asociado principalmente al número de polígonos que se usan para construir la superestructura de deformación elástica. La configuración con $8 \times 8 \times 512$ polígonos es el mejor compromiso entre la agilidad de la deformación y la complejidad de la estructura subyacente dada la configuración del hardware sobre el que se han realizado las pruebas, y es de hecho la que se ha elegido para representar los resultados anteriores a estos experimentos.

7.5.2 Sobre la unión con el método de deformación permanente

El método de deformación elástica se construyó sobre una base existente y funcional: un sistema de estimación de la fuerza y de deformación elastoplástica que, no obstante, omitía la visualización de las deformaciones recuperables. El diseño pues se ha realizado teniendo siempre presente estas condiciones de partida, por lo que la integración de los dos métodos de deformación, el elástico y el elastoplástico, se realiza de forma natural y sencilla.

El pilar fundamental sobre el que se organiza la integración es la *tensión de fluencia estimada* $\sigma_f(\alpha^{\sim})$, que como se explicó en el [Capítulo 6](#) es la que marca el límite entre la deformación elástica recuperable y la deformación plástica que imprime cambios permanentes en el modelo. Para fusionar

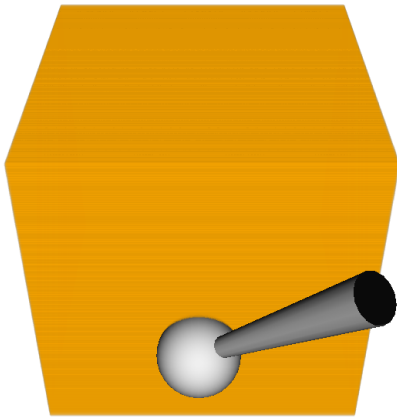
los dos mecanismos basta con activar la superestructura de deformación elástica mientras se controla el crecimiento en la tensión estimada. Cuando se alcanza la tensión de fluencia, se inicia el procedimiento de deformación permanente tal y como se hacía antes de introducir la superestructura, sin que haya que realizar ningún cálculo especial para ajustar el tramo elástico: en toda deformación elastoplástica se calcula la parte proporcional que se añade al tramo recuperable, quedando esta última siempre reflejada en el desplazamiento estimado \vec{u}^{\sim} (tal y como se explica en la [Sección 6.3.2.2](#)). Al ser precisamente el desplazamiento estimado el pilar sobre el que descansa la estimación de la deformación elástica los cambios de la superestructura se consiguen de forma automática tras la actualización de los parámetros cuando se resuelve una carga elastoplástica.

Para ilustrar la combinación de los dos modelos se han dispuesto una prueba de interacción realizada sobre el mismo modelo y con el mismo tamaño de herramienta, en la que se ataca el modelo con las cuatro configuraciones distintas conseguidas activando y desactivando los procedimientos de deformación elástica y plástica. Los resultados conseguidos se pueden apreciar en la [Figura 7.9](#). El método combinado se puede observar en la [Figura 7.9d](#), donde se comprueba que, además de conseguirse una deformación permanente, el modelo se deforma debido a los desplazamientos elásticos que no se reflejan deforma definitiva en su estructura.

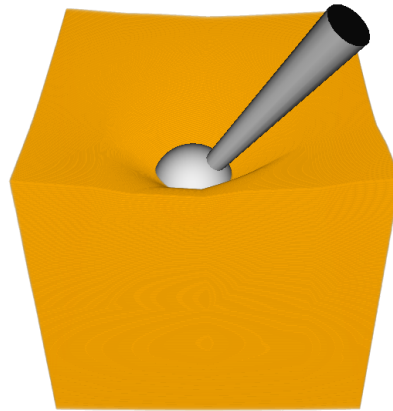
7.6 Conclusiones

La deformación elástica suponía todo un reto dadas las limitaciones del presente entorno de simulación. Descartada la vía de la solución exacta, se requería encontrar una heurística que permitiera representar los cambios de largo alcance que se producen en los materiales elásticos. Finalmente se llega a una propuesta de deformación inspirada en la idea de las estructuras *Free-Form Deformation* y que permite imitar los movimientos elásticos que se observan en los estudios formales sobre la elasticidad como el que se ha utilizado como referencia [[SB88](#)].

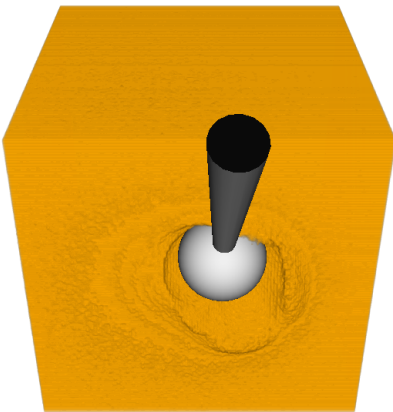
Es posible por tanto representar materiales elásticos puros mediante la *superestructura de deformación* y la *transformación espacial elástica* que



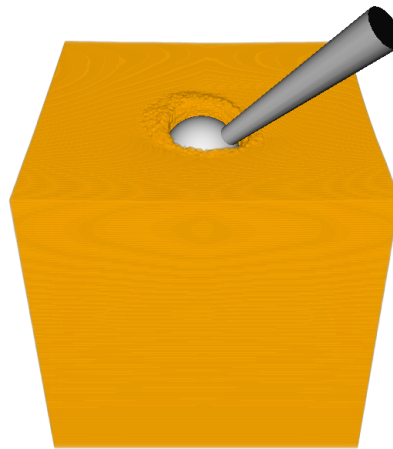
(a) Sin deformación



(b) Deformación elástica



(c) Deformación plástica



(d) Deformación elastoplástica

Figura 7.9 – Resultados visuales del experimento de combinación de la deformación elástica y plástica.

se ha diseñado, así como completar la visualización de la deformación de materiales elastoplásticos que se diseñó en el [Capítulo 6](#).

El método que se presenta tiene ciertas limitaciones en tanto que se considera el volumen subyacente como un medio homogéneo e isotrópico. La transformación espacial que se propone se aplica por igual a todos los puntos de la superestructura, sin importar si los vóxeles asociados están *vacíos* o no. Se deforma el espacio completo de forma indiscriminada (como se puede comprobar en la representación del campo de desplazamientos de la [Figura 7.4](#)) por lo que se pueden producir efectos extraños si el volumen subyacente contiene huecos internos. Los desplazamientos no se atenúan en el espacio vacío como cabría esperar, produciéndose un efecto extraño en el que parece que el vacío *arrastra* al material circundante.

Paralelamente a lo que sucedió cuando se estudió la deformación permanente, los resultados no son todo lo exactos que podría esperarse, si bien la estimación propuesta crea una sensación agradable y creíble. La heurística permite además ofrecer resultados interactivos, consiguiéndose un método robusto que podría ser satisfactorio para determinadas aplicaciones que no requieran un realismo total pero que sin embargo puedan beneficiarse de un efecto rápido y natural como el que se ha construido en este capítulo.

Chapter 8

Conclusions

The present work focuses on the creation of a simulation model based in the theory of deformable solids, which moreover runs in an environment with limited resources and lacks access to basic information — such as the forces acting on the system. The development of such a system was far from linear: there were many problems which had to be solved in parallel, paths that opened and closed, many were also the dead ends which forced a backward march and a reassessment of the whole situation.

Actually, two of the aforementioned impasses heavily affected the entire design process. First, the fact that the force applied by the user was unknowable prevented the direct application of the physical principles. Second, the intrinsic complexity of the calculi required to solve the formal solution limited the proximity to the exact, physically correct deformation.

The simulation system has been structured in three branches, attacking each of them a certain principle:

- **Force.** Probably the first problem encountered in this process was the generation of the force output. The force-related issues comprehended the estimation of the user-applied pressure and the generation of the haptic response. The solution of this problem settled the foundations for the upcoming deformation tasks.
- **Plasticity.** Figuring out how to link the generated force with the permanent changes in the material was with no doubt the hardest

part of this research. The adaptation of the incremental theory of plasticity took most of the efforts put in the design of the plastic changes. This was done in parallel with the design of the surface and inner material deformation algorithms, resulting in what has been called the first approach to the deformation — and more precisely, recursive dispersion algorithm.

- **Elasticity.** Temporary changes in the solid were left unconsidered until the very last part of the research. Whereas permanent changes required a considerable amount of trial and error elastic deformations were solved at once with a visual transformation. The reversible nature of elastic forces and the fact that the rest of the system was well defined and working made much easier their translation to the system.

The most significant conclusion was that theory cannot be directly applied to a simulation system including haptic force rendering (at least in environments without device-provided force estimation). Once this was discovered efforts were placed into creating an approximate solution, trying to keep the process as realistic as it was possible at every step.

In the end the resulting system met most of the expectations put into the research back into its origin. The requirements presented as desirable for the simulation in the introductory essay were, as it will be explained in this chapter, as well fulfilled.

8.1 Achievements

At the end of Chapter 1 a descriptive list of the contributions of this work was given. Throughout the exposition the mentioned achieved goals were explained, giving a much deeper view of how those claims were accomplished. Here all these details will be synthesized, comparing the different developed technics in order to show their strengths and flaws, how well they combine and how close they are to the initial expectations.

The entire simulation system is based and supported by the incremental theory of plasticity, a part of mechanics of deformable solids that models

the behaviour of elastoplastic solids. This theory was studied, adapted and used to create a physically inspired tactile interaction and a natural deformation effect.

Force is estimated using a simplification in the displacements that should occur in the solid, for there is no way to create a reliable estimation based only in the kinetic parameters of position and speed offered by the haptic device. The force output, though built on a simplification, is a robust, plausible solution for the haptic interaction with deformable models.

This force estimation subsystem is the foundation for the deformation method for elastoplastic solids that has been built.

The main concern after the haptic part was cleared was the creation of the algorithm to inflect permanent changes to the model. Plastic deformation was resolved by redistributing property values in a voxelized model, using the movement of the virtual tool (controlled by the haptic device) and the force information to drive the changes both in the surface and the inner material. Reallocation of the moved material is achieved through a process of compression and decompression. This mechanism also improves global performance, for changes in the voxel distribution can be prioritized and solved asynchronously with a lazy algorithm.

Only after plastic changes were solved elasticity was taken into consideration. Whereas the plastic deformation procedure affects the property value distribution, elastic changes are performed with a spatial transformation, affecting only the structure used to visualize the model and not the model itself.

Plastic changes are close to the behaviour of a pure plastic material such as clay or plasticine. The elastic effect that has been proposed is on the contrary closer to rubber-like materials. The combination of the two effects can be adjusted by modifying the elastic limit (also known as flow function) and the elastic constants.

Both of the two deformation mechanisms are a simplification of the formal solution to the problem. Sometimes considering an adaptation of the real circumstances, sometimes merely being inspired by the theoretical results, the proposed simulation is for the most part an heuristic method. Always trying to maximize performance and interactivity, a compromise be-

tween realism and efficiency has been found, considering that the resulting simulation shows a natural, plausible, interactive effect.

Experiments have been carried out to test the performance and scalability of the designed algorithms. The algorithms were implemented in a prototype running on an Intel Core 2 Quad CPU at 2.93GHz with 3.50GB of RAM computer, supported by a nVidia Quadro FX 3700 graphics card. Using this hardware configuration interactive results have been achieved for medium-sized models and tools.

Performance of the force computation algorithm strictly depends on the tool size, since collisions need to be checked all over the tool voxelization. As for the deformation algorithms, the time required to solve a single step of a plastic deformation is proportional to the tool surface, whereas the update rate of the elastic structure has no relation with the model or tool size and strictly depends on the number of polygons used to represent it.

The whole simulation system has been designed so it can be parallelized and executed in a thread pool: the different tasks can be run in parallel using synchronization mechanisms so processor time is more efficiently used.

Partial results of this work have been sent to conferences and journals of international scope. The first results considering the haptic interaction were published in the IV Iberoamerican Symposium in Computer Graphics, SIACG 2009 [PST09]. A study concerning the issues of estimating acceleration in haptic devices was unveiled at the XX Spanish Conference in Computer Graphics, CEIG 2010 [PST10] (which has also been attached to this thesis and can be found in Appendix B). The elastic and plastic algorithms found their way into the Volume 3 of the International Journal of Creative Interfaces and Computer Graphics, IJCICG [PST11b] and the 32nd annual conference of the European Association for Computer Graphics (Eurographics 2011) [PST11a], respectively.

8.2 Comparison with Previous Works

The proposed simulation method makes use of a volumetric structure for the material representation, and furthermore carries out the permanent changes by transferring property values from one voxel to another. This

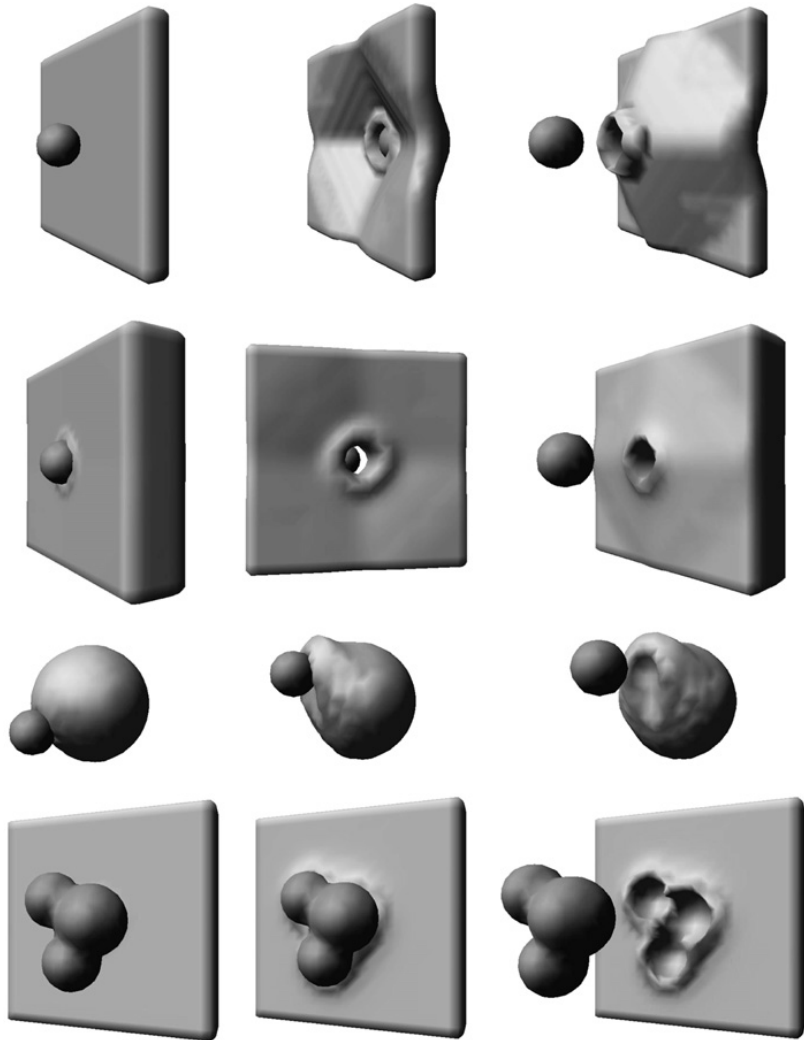


Figure 8.1 – Results from the work by Dewaele and Cani, 2004.

essential mechanism is quite equivalent to the solutions offered by Dewaele and Cani [DC04a, DC04b], Arata et al. [ATTY99] and Sakamoto et al. [STK99]. Even though that neither a haptic interaction mechanism or elasticity are considered there, a detailed comparison with these previous solutions is however convenient.

The first limitation overridden by the deformation method proposed in this work is the orthogonality to which property value redistribution is restrained in the alternate solutions. This leads to the limited effect that

was discussed in the experiment driven in Section 6.5.1. Figure 8.1 shows different interactions of the Dewaele proposal [DC04b]. In the bottom row an interaction similar to that found in Figure 6.14 can be seen, so the different material transfer approaches can be compared.

Secondly the here proposed simulation includes a prioritization mechanism that, combined with the lazy resolution of the minor changes, guarantees interactive results for larger interactions.

The method by Dewaele et. al also proposes a long-distance displacement field that allows torsions and rotations in the material. The effects of this technique can be perceived in the interactions in the first two rows of Figure 8.1. As the tool passes through the material the displacement field twists the model, also creating artifacts due to the orthogonal and discrete treatment of the distances. This long distance plastic deformations, though inspired in the intuitive behaviour of clay, create a interesting effect that has no equivalent in the solution developed in this work.

The work of Arata and that of Sakamoto use the same approach to create permanent deformations, but no improvement over the presented system or the work by Dewaele and Cani has been found.

As to the author's knowledge, no similar work to the designed elastic deformation method has been built (see the related works in Chapter 2). No time or force measurement has neither been found in the works presenting alternatives to the haptic rendering algorithm, so no comparison can be done whatsoever.

8.3 Future Work and Outlook

Many were the solved problems, but also many were the paths left opened which could be the way to go in a further stage of this research.

It would be for instance very interesting to recreate a exact simulation of the deformation using a limited integration scheme and a simplified model as discussed in Section 6.2. It might be possible to use the elastic superstructure to compute the exact solution to the problem following the idea found in the work of Mezger et al. [MTPS08]. It would be then necessary to assume that the estimated force $\tilde{\vec{f}}$ is precise enough to be used as the

surface tension that triggers the deformation process, and solve possible divergences that could appear in the displacements due to force estimation errors.

Hardening in the plastic deformation has been considered only partially. The hardening pattern is computed along with the interaction but it is not reflected in any form in the material or its properties. It could be possible to record the history of plastic changes in the model so hardening between interactions is consistent.

An open path is also the deformation of anisotropic media. All along this work materials have been supposed to be isotropic, so deformation processes do not change regardless of the orientation. Mixed environments would require a detailed analysis and a mechanism to extract material properties from the model.

Close to the previous line of work is the problem of the reflected boundaries that appeared in the study of Prevost [Pre84]. These boundaries acted as inelastic bands that did not admit any degree of deformation, therefore *reflecting* the forces and the displacements. A wider study on the deformation of composite media could formalize the mentioned behaviour.

The friction forces generated by the movement of the tool were left unconsidered. In future works they could be incorporated to the system, thus increasing the plausibility of the simulation.

Considering that elasticity has been implemented with a spatial transformation ψ embedded in the visualization process, it could be possible to transfer the computation of the displacements to the graphics processing unit (GPU) to take advantage of the dedicated graphics hardware. The elastic displacements $e(P)$ of the points P in the structure could be then performed in a per-vertex approach, therefore improving the quality of the visualization and removing the artifacts derived from the linear interpolation performed in the polygons of the elastic superstructure.

Finally it would be also challenging to migrate the plastic compression and relaxation problem to the GPU, to check if deformation times could be improved.

These were the ideas that were left aside throughout the development of this work due to the lack of time or resources. All the mentioned proposals

Chapter 8. Conclusions

would make however an appealing quest, for this thesis laid the foundation for a further research on deformable materials. Future works based on the principles described here could also include other phenomena such as hysteresis, or even take the road to the simulation of human tissue, which was actually one of the motivations that engaged the present study back on its origin.

Apéndice A

Implementación

Los esfuerzos teóricos que todo proceso de investigación acarrea tendrían poco o escaso valor si no se sometieran a *experimentación*. Esto resulta especialmente cierto en el ámbito de la informática gráfica, donde se espera en todo caso resultados *visuales* de todo trabajo, tratamiento o interacción que se realice sobre los modelos objeto de estudio.

Las referencias a técnicas de implementación, lenguajes de programación y bibliotecas usadas han sido omitidas deliberadamente de los capítulos anteriores, con la intención de realizar una separación tácita entre el diseño abstracto de la metodología de simulación y los detalles concretos que han materializado los procedimientos descritos en rutinas comprensibles para un ordenador. En este anexo se darán estos detalles de programación, complementando así la visión exclusivamente cualitativa de la fase de experimentación que en los anteriores capítulos se ha realizado.

A.1 La aplicación *Volume Works*

La validación de los conceptos que se han desarrollado a lo largo de este trabajo se ha realizado en el marco de una aplicación de modelado de volúmenes a la que se bautizó como *Volume Works*.

Esta aplicación, desarrollada usando el lenguaje C++ con el apoyo de la plataforma *Qt* en su revisión 4.x [Mol07], integra las técnicas de visualización e interacción que se han expuesto. Para la visualización de los modelos

volumétricos se ha usado la conexión que *Qt* ofrece con la biblioteca gráfica *OpenGL* (*Open Graphics Library*). *Qt* ofrece unas clases predefinidas que hacen particularmente fácil la creación de una rutina gráfica que muestra los resultados en un *widget* o elemento de una ventana. La interacción se diseñó conforme a los principios que se indicaron en la [Sección 1.2.3](#), utilizando tanto un ratón de escritorio como el dispositivo háptico *PHANTOM*[®] *Desktop*[™] de *Sensable Technologies*. El ratón controla por tanto la rotación del modelo, el cual gira sobre un punto central para que pueda examinarse desde diferentes ángulos, mientras que el dispositivo háptico permite mover la herramienta virtual que se simboliza en la pantalla como una esfera con un mango.

A.1.1 Interfaz de usuario

Volume Works compone todos los elementos necesarios para la visualización y la interacción en una única ventana. Dispone de un menú y botones para las opciones de la aplicación en la parte superior, una lista de opciones de visualización en la parte izquierda, y en la parte derecha, ocupando la mayor parte del espacio, se emplaza la paleta de colores (de la cual más tarde se hablará) y lo que en *Qt* se tiene a bien denominar el *GLWidget* (por ser el enlace con *OpenGL*), al cual aquí se referirá como *elemento gráfico*. Es este elemento gráfico el protagonista de la aplicación, pues es el marco en el cual se visualiza la representación gráfica tanto del modelo como de la herramienta, teniendo lugar por tanto ahí toda la interacción y procesado visual de la simulación. En la [Figura A.1](#) se muestra una visión de la pantalla principal de *Volume Works*.

La aplicación permite activar y desactivar por separado los procedimientos de la deformación elástica y plástica, así como cambiar el módulo elástico que se usa para la estimación de la fuerza y el límite entre la zona elástica y la zona de carga elástica. Además, se puede mediante el menú seleccionar el algoritmo que se usa para la relajación en la deformación permanente, de forma que se pueden comparar fácilmente los efectos producidos por las tres soluciones que se han desarrollado.

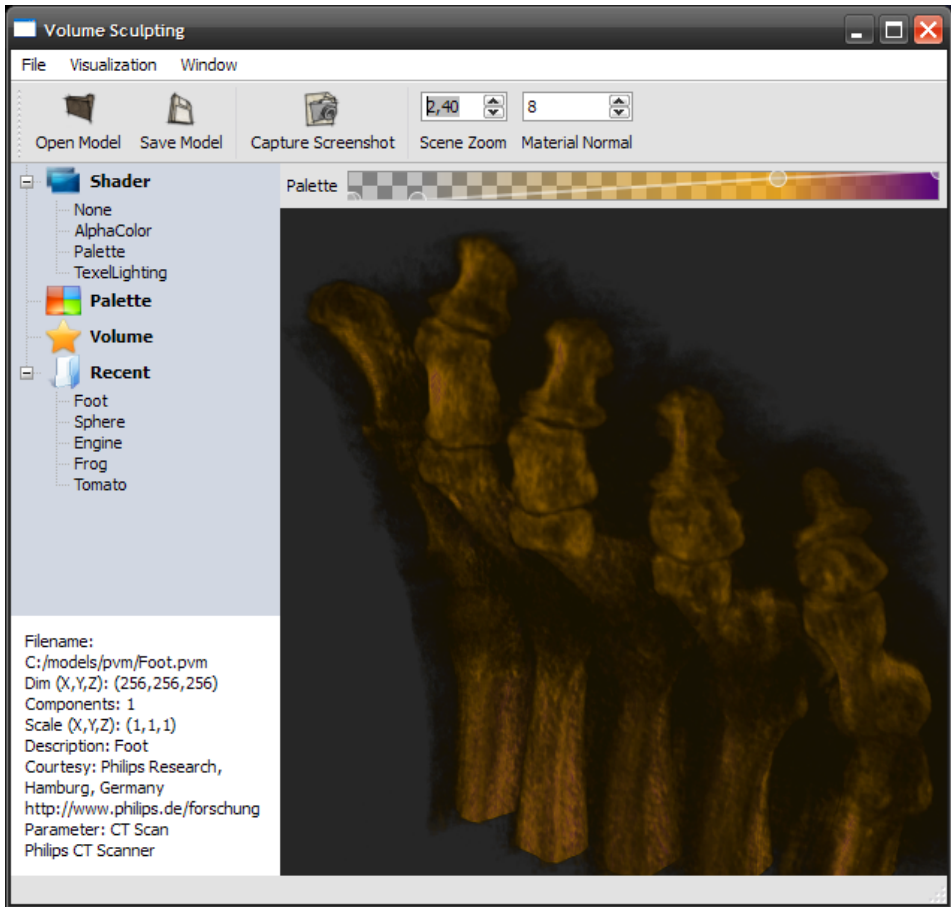


Figura A.1 – Ventana de la aplicación *VolumeWorks*. Se aprecia que el protagonista de la composición es el modelo, en este caso representando la imagen TAC de un pie. En el lateral están las opciones de visualización, y debajo del menú otros controles de la aplicación, tales como el zoom o las propiedades del material. A la derecha, ocupando la mayor parte de la ventana, queda el elemento gráfico. En la parte superior del elemento gráfico aparece la *paleta de colores*, que se usa para asociar un color concreto y un nivel de transparencia a cada valor de propiedad. En ella se observa que los valores bajos de propiedad son transparentes, mientras que los valores en torno al 80 % del rango se marcan en amarillo. Los valores máximos deberían aparecer en morado, de no estar ocultos por zonas circundantes más opacas.

A.1.2 Arquitectura multitarea

Para organizar las distintas tareas que es necesario acometer *VolumeWorks* se organiza en varias hebras. Éstas se ejecutan en paralelo, disponiendo todas ellas de acceso compartido a las estructuras de datos que representan el modelo, la herramienta y los parámetros de la estimación de la deformación. La coordinación entre las mismas se realiza sobre las mismas estructuras de datos, que disponen de etiquetas que sirven de mensaje para iniciar y parar los distintos procesos. La estructura de hebras queda pues como sigue:

Hebra principal Es el proceso principal que lanza *Qt*, y que se encarga de manejar los eventos del ratón y teclado. Maneja toda *interacción con el usuario* salvo la del dispositivo háptico, activando los procedimientos que corresponden en cada caso cuando se activan los controles en pantalla (como por ejemplo, abrir o cerrar un modelo o cambiar el zoom de la visualización).

Hebra gráfica Es la que se encarga de la *visualización del modelo* y de la *superestructura de deformación elástica*. Haciendo uso de la funcionalidad que aporta *OpenGL*, interpreta la estructura que contiene la información del modelo volumétrico y lo muestra en el elemento gráfico de la ventana. Implementa el método de visualización directa basado en *cortes en profundidad* que se describió en la [Sección 1.2.2.5](#). El cálculo de la deformación elástica se realiza durante el proceso de visualización, tal y como se explicará más adelante. Además recibe la información de posición del puntero háptico para dibujar en pantalla la herramienta virtual.

Hebra de colisión Lleva a cabo el *cálculo de las colisiones* de la herramienta con el material y el *cálculo de la fuerza estimada* \vec{f}^{\sim} . La hebra computa en todo momento la intersección entre la herramienta y el volumen tal y como se describió en la [Sección 5.4](#). Además mantiene el control sobre la tensión de fluencia estimada $\sigma_f^{\sim}(\alpha^{\sim})$, estando por tanto al cargo de notificar los cambios en la superficie cuando se entra en la zona de carga elastoplástica y es necesario registrar una deformación permanente. Se delega cualquier operación que deba realizarse

sobre la representación del volumen, de forma que se evita la carga computacional que esto conlleva y se maximiza así la frecuencia de actualización de las estimaciones relacionadas con la respuesta táctil.

Hebra háptica Envía la *respuesta háptica* al usuario, usando la fuerza estimada \vec{f}^{\sim} que estima la hebra de colisión. Si la hebra de colisión a una frecuencia menor a la frecuencia de actualización que exige el dispositivo háptico ($1kHz$), el vector de fuerza que se envía como respuesta se obtiene por interpolación entre valores consecutivos de \vec{f}^{\sim} , insertándose valores intermedios entre estimaciones para evitar la posible percepción de una baja frecuencia de actualización.

Hebra de deformación Por su parte, la hebra de deformación recoge las indicaciones de la hebra de colisión cuando la última apunta cambios que deben suceder en la superficie. Se ocupa por tanto de, en primer lugar, realizar los pertinentes *cambios en la superficie*, para a continuación aplicar bien el *algoritmo de dispersión recursiva*, el *algoritmo del operador de suavizado* o el *algoritmo del operador de desplazamiento* (según la opción marcada por el usuario en las opciones de la aplicación) para relajar el material y efectuar el reparto de material.

La arquitectura basada en hebras se debe principalmente a la necesidad de distribuir la carga computacional. En algunos casos, como es el de la interacción háptica, es estrictamente necesario mantener un subproceso paralelo que se engargue exclusivamente de la interacción, y en otros, como en la separación entre la detección de colisiones y la ejecución de la deformación plástica, porque la ejecución en paralelo permite la resolución retardada y progresiva de los cambios permanentes que deben efectuarse. En cualquiera de los casos la ejecución en hebras supone un mejor aprovechamiento de las capacidades de cálculo paralelo de los equipos actuales, teniendo en cuenta que la práctica totalidad de los procesadores comerciales modernos incorporan múltiples núcleos que el sistema operativo gestiona de forma automática y transparente para el programador.

En la implementación concreta que se ha realizado no obstante el cuello de botella no es la carga del procesador, sino el acceso a la estructura de datos que representa el material. Todas las hebras acceden de forma

concurrente a dicha estructura, y, debido a que deben realizarse lecturas y escrituras el acceso debe realizarse en exclusión mutua, siendo esta la mayor limitación en el rendimiento del sistema implementado.

A.2 Las rutinas hápticas

Tal y como se ha comentado brevemente en la sección anterior la interacción háptica requiere la programación de una hebra independiente que procese la entrada y la salida de datos del dispositivo. Dicha gestión se realiza a través de las bibliotecas *OpenHaptics* [Sen08], que han sido desarrolladas por *Sensible Technologies* para facilitar el control de los dispositivos hápticos que comercializan y entre los cuales se encuentra el *PHANTOM[®] Desktop[™]* que se ha utilizado para los experimentos (ver [Figura A.2](#)).

OpenHaptics ofrece varias vías para controlar el dispositivo háptico, que se diferencian en el grado de abstracción en el que sitúan la generación de la sensación táctil: la interfaz *QuickHaptics*, que realiza una gestión semiautomática de la respuesta en fuerza según los modelos enviados a *OpenGL*; la capa de abstracción *HLAPI* (de la voz inglesa *Haptic Library Application Programming Interface*, interfaz de programación de alto nivel), que permite definir *fronteras táctiles* que acompañen a los modelos de superficie que se visualizan; y la *HDAPI* (de *Haptic Device API*, interfaz del dispositivo háptico), que ofrece un acceso directo a las funciones más esenciales del dispositivo, permitiendo un limitado conjunto de operaciones de bajo nivel tales como leer los valores de los sensores de posición o enviar un vector de fuerza.

Tras el estudio de las tres alternativas de programación se estimó que la *HDAPI* aportaba herramientas suficientes para atacar el problema propuesto. Se concluyó además que el resto de interfaces aportaban utilidades y capas de abstracción innecesarias que no aportarían funcionalidad y que por contra podrían repercutir en una reducción del rendimiento. La interfaz *HD* de *OpenHaptics* permite:

- Acceder a la posición \vec{c} del cursor háptico en cada ciclo.
- Acceder a la orientación \hat{o} del mango en cada ciclo.

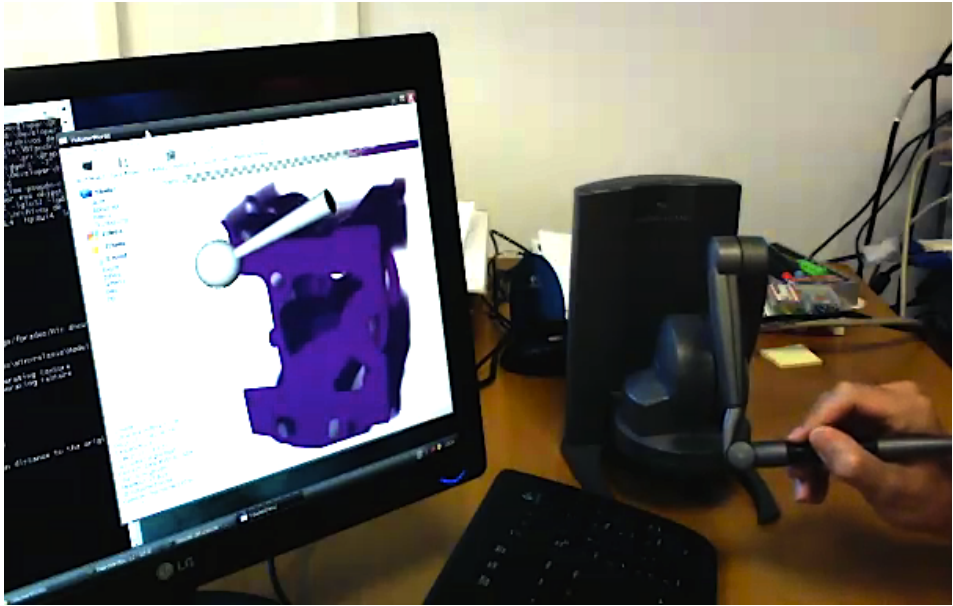


Figura A.2 – Entorno de trabajo en el que se ha desarrollado y experimentado el modelo de simulación propuesto. Se aprecia la correspondencia entre el puntero y mango háptico y la herramienta virtual dibujada en la ventana de *VolumeWorks*.

- Enviar un vector de fuerza de respuesta \vec{f} en cada ciclo.
- Un marco para implementar el *callback háptico* $h(\vec{c}, \hat{o}, \omega)$ del que se habló en el [Capítulo 5](#), permitiendo realizar operaciones personalizadas según las necesidades.

Teniendo en cuenta además que la *HLAPI* y la interfaz *QuickHaptics* están enfocadas principalmente a la adición de interacción táctil a modelos de superficie, se antoja especialmente oportuno el uso de la interfaz de bajo nivel *HDAPI*. Es además por su simplicidad un marco ideal para generar campos vectoriales (de fuerza) calculados en función de otro campo definido en el espacio \mathbb{R}^3 (como es el caso del campo escalar al que equivale el modelo volumétrico que se usa para la representación del material).

La fuerza estimada \vec{f}^{\sim} que se enviará al usuario se calcula en la hebra de colisión. Los cálculos con la posición muestreada \vec{c} y con los parámetros del entorno que en el [Capítulo 5](#) se definieron como ω son por tanto responsabilidades ajenas, por lo que las operaciones que debe realizar la hebra que comunica con el dispositivo háptico se limitan a, en cada ciclo de mues-

treo, leer los valores de \vec{c} , enviar el vector de fuerza 1000 veces por segundo e interpolar la respuesta si fuera necesario. Como se ha comentado anteriormente, esta interpolación de \vec{f}^{\sim} se realiza para minimizar la diferencia entre muestras de cara a una posible baja frecuencia de actualización en su valor. Esto produce, como es evidente, un desfase en la respuesta, pues es necesario esperar a tener al menos dos valores consecutivos de la estimación para poder enviar al dispositivo el *camino interpolado* entre los dos. Este retardo es admisible mientras la estimación no baje de los $10Hz$ (lo que implicaría una desincronización en la respuesta de $0.1s$), pues se ha probado experimentalmente que pasado este límite los efectos de la interpolación se empiezan a hacer evidentes al usuario [PST10]. En cualquier caso una estimación tan lenta no es admisible en el sistema de simulación, por lo que la interpolación puede mantenerse como principio general sin temor a que sus efectos sean percibidos.

A.3 Técnica de visualización

Como ya se adelantó en el [Capítulo 7](#), para la visualización del modelo volumétrico se ha elegido el procedimiento que muestra el volumen como una serie de *cortes en profundidad* por la compatibilidad que éste presenta con la técnica de deformación elástica que se ha diseñado. En lo que sigue se detallarán los pormenores de la visualización, lo que permitirá más adelante explicar cómo se adapta la superestructura de deformación a la misma.

A.3.1 Implementación del método de visualización

El procedimiento de los cortes apilados resulta especialmente fácil de implementar gracias a las capacidades del hardware gráfico actual y las herramientas ofrecidas por *OpenGL*. La clave está en el concepto de *textura 3D* que se introdujo en la especificación 1.2 de *OpenGL*, y que permite enviar la representación volumétrica directamente a la biblioteca gráfica si se utiliza la estructura de datos adecuada. Esta asociación directa entre volumen y textura se realiza determinando que cada vóxel del primero corresponderá a un tétel (abstracción del píxel según la nomenclatura de *OpenGL*) de la segunda [MLM03, CCF94]. La biblioteca considera entonces el volumen como

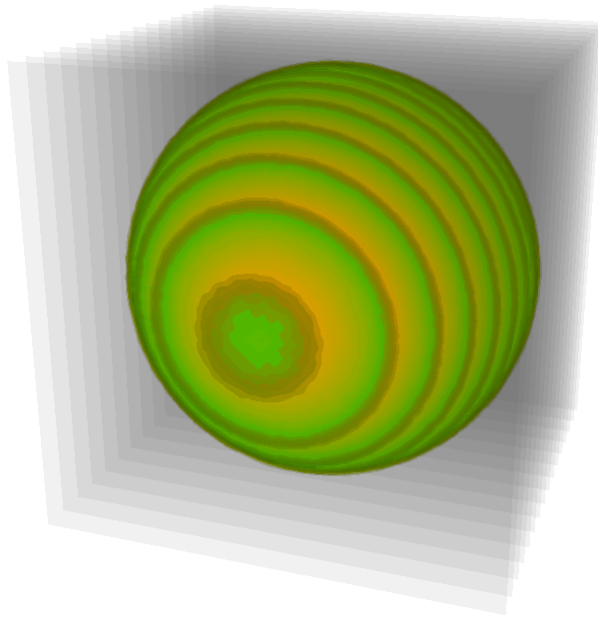


Figura A.3 – Resultado de aplicar el método de visualización directa basado en cortes que se ha usado a la representación volumétrica de una esfera, que se ha coloreado además en su interior de amarillo, de color verde en las zonas más exteriores y de negro en su frontera para que se puedan apreciar los contornos. Los polígonos que intersectan el volumen y que normalmente están dispuestos perpendicularmente a la vista se han rotado aquí ligeramente para que se aprecie mejor su distribución.

una imagen 3D que se puede asociar a cualquier polígono, calculándose automáticamente la intersección entre la imagen y los polígonos de la escena cuando se indican correctamente los puntos de anclaje (lo que se conoce como *coordenadas de textura*).

Para componer la imagen final en profundidad usando esta funcionalidad basta con situar un número determinado de polígonos perpendicularmente a la dirección en la que se observa separados de forma equidistante, asociarles a todos la misma textura 3D e indicar a *OpenGL* a qué coordenada de la textura 3D corresponde a cada vértice de cada polígono. Este concepto se ilustra en la [Figura A.3](#), en la que aparece un modelo que se visualiza con 16 cortes. Si bien se desvirtúa el efecto de profundidad con un número tan bajo de planos se aprecia por contra mejor la idea que se acaba de exponer.

Cabe también destacar que las rotaciones y traslaciones del modelo se pueden realizar aplicando una *transformación geométrica* a la textura. *OpenGL* recalcula las intersecciones cuando se realizan operaciones espaciales sobre las texturas, lo que permite que los planos permanezcan siempre perpendiculares a la dirección en la que mira el observador y facilita considerablemente los cálculos que se realizan para cambiar el punto de vista. Por ejemplo, si se quiere observar el modelo de frente pero con un levantamiento de 45° basta con *rotar* la textura hasta conseguir dicho ángulo. Los polígonos obtendrán las secciones que correspondan a la nueva posición de la textura sin necesidad de cambiar nada en su definición, pues *OpenGL* calculará automáticamente sus coordenadas de textura tras realizar la rotación.

A.3.2 Función de transferencia

El mecanismo que permite mostrar, ocultar y colorear el modelo volumétrico se implementa también sobre *OpenGL*, en este caso con la programación directa de funciones en el procesador gráfico mediante el lenguaje *GLSL* (acrónimo de *OpenGL Shading Language*). Basado en el lenguaje C, *GLSL* permite un control total sobre lo que se dibuja en la pantalla, lo cual posibilita por ejemplo cambiar el color de una determinada zona de la escena atendiendo a su posición en el espacio o al valor de su coordenada de textura.

Es este camino precisamente el que se sigue para implementar lo que de forma abstracta se conoce como la *función de transferencia* del volumen [PLB⁺01], y que no es más que la correspondencia entre los valores de propiedad encontrados en el modelo y el color y valor de transparencia que mostrarán en pantalla. La posibilidad de definir una función de transferencia no sólo es una vía para *colorear* el volumen sino que es una potente herramienta de exploración de los datos. En el caso concreto de *Volume-Works* la función de transferencia se define manualmente por el usuario sobre paleta de colores que se mostró en la [Figura A.1](#). El eje horizontal equivale al dominio de la función, esto es, el valor de propiedad de los vóxeles del volumen, estando a la izquierda los valores bajos y a la derecha los más altos. El eje vertical representa el valor de transparencia asignado, que

oscila entre totalmente opaco (arriba) y totalmente transparente (abajo). El color se añade insertando puntos de control entre los que se realiza interpolación lineal, generando una curva de color que es enviada directamente al programa *GLSL*. Este último se limita a sustituir en pantalla los valores sencillos de propiedad del modelo por el color que les corresponde según la función de transferencia definida, manifestándose de esta forma en la visión del volumen la ordenación de colores que el usuario define.

A.3.3 Interpolación

El resultado final de la visualización no coincide *exactamente* con la reconstrucción del modelo que se desarrolla en el [Capítulo 3](#), en la que se indica que todos los puntos del espacio tomarán el valor de la muestra más cercana. Según los resultados mostrados en la [Figura 1.9c](#) el modelo debería visualizarse como un conjunto de cubos apilados, mientras que en las imágenes que muestran los resultados la apariencia es más parecida a lo que se muestra en la [Figura 1.9d](#). Esto se debe a que *OpenGL* permite realizar una interpolación lineal en todas las dimensiones de la textura, resultando en una interpolación trilineal para las texturas 3D que *suaviza* considerablemente el resultado final dando una mayor sensación de continuidad.

Si bien al admitir la interpolación de la textura aparece un desfase entre la interpretación interna del modelo y la imagen de él que se obtiene en pantalla no se produce, sin embargo, interferencia alguna con el funcionamiento de la simulación. Su utilización afecta exclusivamente a lo visual y que no interfiere para nada con la manipulación de la estructura que realizan las demás hebras.

A.4 Superestructura de deformación elástica

Ya se adelantó en el [Capítulo 7](#) que la superestructura de deformación se integra en el proceso de visualización, usando los cuadriláteros de la visualización por cortes como base para construir la estructura. Queda pendiente, no obstante, explicar cómo se traduce la superestructura en polígonos interpretables por *OpenGL*, proceso que como se verá es idéntico al que se ha expuesto anteriormente al detallar la implementación de la visualización.

Tal y como se explicó en su momento los $l \times m \times n$ puntos de la superestructura se subdividen en n planos, organizándose además los $l \times m$ puntos X_{ijk} asignados a cada plano $k \in [1..n]$ en una unión de $(l - 1) \times (m - 1)$ cuadriláteros. La implementación de dichos parches se consigue por tanto haciendo coincidir los n planos de la superestructura con los polígonos que se usan para la visualización, *fragmentando* además estos últimos en subpolígonos que sean coincidentes con la subdivisión establecida para cada plano k . Cada corte de la visualización vendrá por tanto definido por los puntos X_{11k} , X_{1mk} , X_{lmk} y X_{l1k} (esto es, los puntos de las esquinas de cada plano k).

Como se ha comentado en la sección anterior, a los polígonos de la estructura creada para la visualización se les asigna unos puntos concretos de la textura 3D, que como antes se comentó en *OpenGL* se conocen como *coordenadas de textura*. Se notarán para esta exposición como V_{ijk} las coordenadas de textura que corresponden a cada punto X_{ijk} de la superestructura elástica y de visualización. El mecanismo de deformación elástica funciona precisamente por esta dualidad en los sistemas de coordenadas. Al crearse la estructura se asocia de forma unívoca cada polígono con una zona concreta de la textura, definida por la asociación entre sus cuatro puntos X_{ijk} y las coordenadas de textura V_{ijk} que le corresponden. Al dibujarse los polígonos en las posiciones X'_{ijk} calculadas con la transformación espacial elástica *OpenGL* normalmente les asignaría los puntos V'_{ijk} de la textura; sin embargo, puede asociarse cada punto transformado X'_{ijk} a su coordenadas de textura originales V_{ijk} , produciéndose de esta forma un *desplazamiento* efectivo de los puntos de la textura según lo establecido por la transformación elástica.

No es necesario declarar ninguna acción adicional para desplazar los puntos interiores de cada parche, pues *OpenGL* los calcula automáticamente por interpolación según las coordenadas de textura asignadas a los vértices de los polígonos.

Este funcionamiento basado en el desplazamiento de unos pocos puntos que definen parches cuadrados dentro de los cuales se interpola la textura es lo que deriva en las discontinuidades que se pueden ver en las imágenes que muestran las deformaciones elásticas. En la [Figura A.4](#) se puede apre-

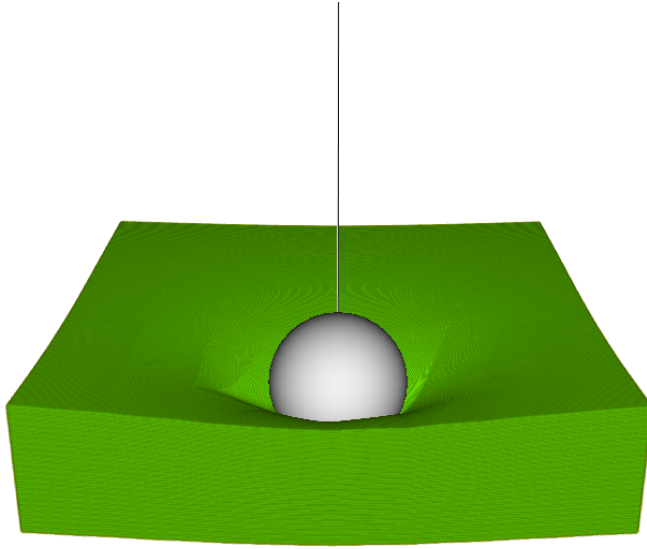


Figura A.4 – Deformación de una plancha plana de material en la que se pueden apreciar las discontinuidades que aparecen en las uniones entre polígonos.

ciar que si bien la curvatura en profundidad sí cambia suavemente, debido al gran número de cortes que se usan para la visualización, la pendiente conseguida a lo ancho presenta sutiles cambios en ángulo en las zonas de unión de los polígonos. Esto se debe, en definitiva, a que todas las curvas en la deformación elástica se consiguen componiendo los tramos rectos que definen los polígonos transformados.

La simplicidad con la que se puede representar la deformación elástica permite delegar en la hebra gráfica el cálculo de la posición deformada de los puntos X_{ijk} . La única operación adicional que se debe realizar en cada ciclo gráfico es precisamente calcular los desplazamientos que moverán cada punto X_{ijk} hasta su homólogo X'_{ijk} y asignar las coordenadas de textura originales V_{ijk} a cada uno de ellos. Carece de sentido calcular la deformación elástica a mayor frecuencia de lo que se actualiza la visualización, pues es este un efecto que, tal y como se ha estructurado el sistema de simulación, afecta exclusivamente a la respuesta visual.

A.5 Conclusiones

La implementación del sistema de simulación de sólidos elastoplásticos que en este trabajo se ha descrito no sólo ha servido para validar las premisas que se enunciaron ante el estudio de las soluciones anteriores y de la teoría, en las que se anticipaba la dificultad de conseguir un método exacto e interactivo de deformación, sino que además ha permitido, mediante ciclos de prueba y error, avanzar hacia el compromiso heurístico al que se ha llegado.

El proceso de ingeniería se ha ejecutado siempre en paralelo al estudio de la teoría. La búsqueda de soluciones técnicas para los problemas que se planteaban permitió en muchos casos descartar caminos poco viables y elegir soluciones más prácticas. Este trabajo es en definitiva un estudio híbrido entre lo teórico y lo práctico que en todo caso ha procurado, siendo consecuente con la motivación que lo inició, maximizar la fidelidad al proceso mientras que se conseguían resultados interactivos aceptables en el marco tecnológico sobre el que se ha trabajado.

Apéndice B

Cinemática háptica

Los dispositivos hápticos proporcionan una sensación realista e inmersiva cuando se trata de *tocar* o percibir objetos del mundo virtual. Sin embargo, el paradigma de trabajo plantea algunos problemas cuando lo que se pretende es simular interacciones fieles a las leyes de la cinemática, tales como velocidad, aceleración o las fuerzas de inercia.

La cinemática deriva todas las magnitudes de las ecuaciones del movimiento. Estas leyes plantean serios problemas a la hora de trasladarlas a modelos de interacción háptica: las trayectorias están lejos de ser uniformes, observándose desviaciones y vibraciones de alta frecuencia en la misma. La alta precisión de los sensores del dispositivo por una parte y la naturaleza discreta del sistema de referencia conllevan discontinuidades en las trayectorias ejecutadas por el usuario, lo que se traduce en medidas imprecisas y discontinuas de la aceleración y la velocidad. Este problema se agrava cuando los parámetros estimados han de ser usados en la propia respuesta háptica, lo que puede llevar a tirones y vibraciones indeseadas.

En lo que sigue se estudiará el problema de la discontinuidad en los parámetros cinemáticos del mundo virtual, y más en concreto cuando éstos se utilizan para realimentar la salida háptica.

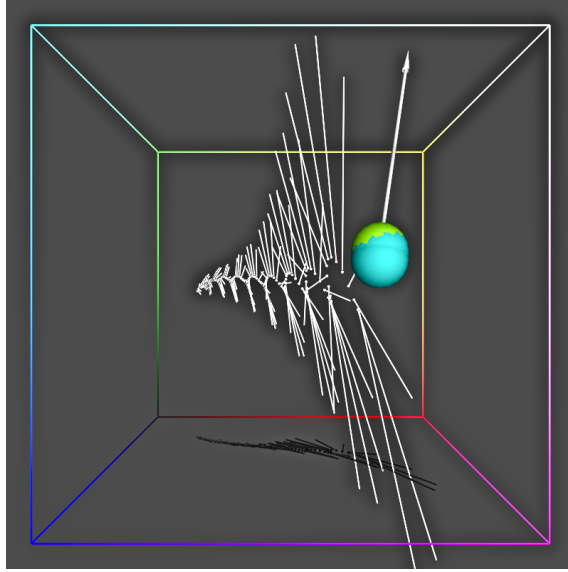


Figura B.1 – Efecto rebote provocado por la sucesión de fuerzas opuestas en la salida háptica. Se muestran los puntos de la trayectoria y la aceleración medida en cada uno.

B.1 Introducción

Históricamente, el método de exploración de objetos virtuales usado en interacción háptica se ha reducido a la aplicación de la Ley de Hook,

$$\vec{F} = -k \cdot \vec{d} \quad (\text{B.1})$$

calculándose la fuerza enviada al usuario como una magnitud proporcional en módulo a la distancia entre la posición de la herramienta que representa al dispositivo háptico y la superficie del objeto virtual. Esta fuerza se empieza a enviar en cuanto se detecta una colisión entre la posición de la herramienta y la superficie.

Esta técnica deriva de los tempranos trabajos realizados por Adachi et al. [AKO95], en los que el empuje del objeto virtual es proporcional a la distancia recorrida hacia su interior. Igualmente, el enfoque del *proxy* ideado por Ruspini et al. [RKK97] consigue un método eficiente de exploración usando una fuerza de repulsión proporcional a la distancia del *proxy* a la posición real de la herramienta.

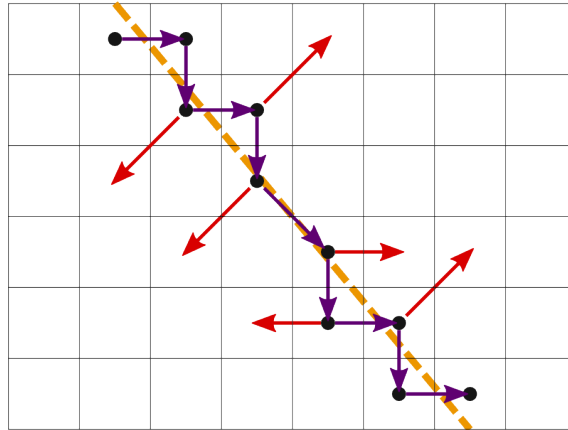


Figura B.2 – Esquema simplificado en 2D de un espacio discreto de posicionamiento. La recta naranja punteada es una trayectoria real. Los puntos muestreados se muestran en negro. Se marcan además los vectores de velocidad (en magenta, unen los puntos en los que se detecta) y de aceleración (vectores rojos).

Las fricciones [MRF⁺96], deformaciones [KP04, CSH03] e incluso la exploración de modelos volumétricos [AS96, LYG02] también han sido ampliamente estudiadas, siempre presentando modelos que se oponen al movimiento del usuario basados en el concepto de *proxy* o en métodos *spring-mass*.

La creación de respuestas hápticas basadas en otros parámetros cinemáticos tales como la velocidad o la aceleración plantean dos limitaciones básicas:

1. *La naturaleza discreta de los sensores.* Los sensores de posición devuelven la localización en un dominio discreto y discontinuo. Esto produce que las trayectorias no sean derivables y que se produzcan fuertes discontinuidades en la medición de la velocidad y la aceleración en mediciones consecutivas (ver figura B.2).
2. *El paradigma de trabajo en ciclos.* La respuesta en fuerza se devuelve al final de un ciclo de movimiento. Esto quiere decir que, recorrida una distancia \vec{d} , se devuelve una fuerza \vec{F} . Consecuentemente la fuerza siempre tiene, en el mejor de los casos, un ciclo de retraso respecto del movimiento.

Estos dos handicaps se aúnan en el *problema de los rebotes*, cuyos efectos se pueden apreciar en la figura B.1. Los *rebotes* se producen cuando el sistema produce fuerzas que se oponen a la velocidad o la aceleración de la herramienta. Si la respuesta en fuerza es lo suficientemente grande, el sistema no sólo opondrá resistencia sino que desplazará el puntero (y presumiblemente la mano del usuario) en la dirección opuesta a la deseada. Si el sistema se vuelve a oponer al movimiento, repetirá este desplazamiento en la dirección opuesta, posiblemente amplificado.

Cualquier entorno virtual que requiera una oposición al movimiento y no sea susceptible de usar las soluciones basadas en la Ley de Hook está expuesto a estos problemas. En este apéndice se hará un estudio de diferentes métodos ideados para estimar los parámetros cinemáticos. Para situar el contexto de la aportación, en primer lugar se hará una revisión de anteriores trabajos que han atacado el problema de la medición de la velocidad y aceleración. Posteriormente, se hará un breve apunte teórico sobre cinemática, lo que sentará las bases del posterior desarrollo del estudio.

B.2 Trabajos previos

Janabi-Sharifi et al. [JsHsJC00] presentaron un método de promediado adaptativo para obtener medidas suavizadas de la velocidad. El método actúa como un filtro variable que atenúa las frecuencias según la propia medida de la velocidad. No obstante, las pruebas fueron realizadas en un dispositivo háptico de dos grados de libertad, e incluso con los promedios más amplios aparecen vibraciones e irregularidades en los resultados. Si bien esto puede no resultar especialmente significativo para la medida de la velocidad, puede resultar dramático para la aceleración en un entorno con tres grados de libertad, puesto que aparecerían aceleraciones perpendiculares a la trayectoria en tramos rectilíneos.

Una solución parecida fue la que presentó Liu [Liu02], en la que trata de reducir el ruido haciendo un promedio inteligente de la velocidad instantánea. Utiliza las características de los propios sensores para reducir la resolución dinámicamente en cada paso y proporcionar una medida suave de la velocidad. Si bien este método reduce el ruido de las curvas de veloci-

dad, siguen apareciendo picos, llevando al mismo problema que presentaba el método de Janabi-Sharifi et al.

Aunque no perteneciente al entorno de la interacción háptica, más reciente es el trabajo de Merry et al. [MvdMS10], que ataca directamente el problema del suavizado de la trayectoria. Propone una reconstrucción de la trayectoria original, perdida en la discretización que realizan los sensores ópticos. Para ello se utilizan splines, lo que proporciona una medida continua de la velocidad y de la aceleración. Este método sin embargo no es directamente aplicable al dominio de la respuesta háptica, pues se obvia el problema de las vibraciones de alta frecuencia, lo que puede derivar en rebotes.

Zhou et al. [ZSPG08] presentaron unos prometedores resultados teóricos, aplicando un filtro de Kalman para estimar la velocidad y aceleración de la trayectoria seguida por un puntero manejado por un usuario. Sin embargo, no presentaron resultados prácticos en un entorno de simulación real.

B.3 Cinemática

En un sistema de referencia inercial la velocidad se puede obtener midiendo el incremento del vector de posición en el tiempo:

$$\vec{v} = \frac{d\vec{x}}{dt} \quad (\text{B.2})$$

Además la fuerza es la derivada en el tiempo de la cantidad de movimiento \vec{p} , que se puede calcular asimismo con la velocidad:

$$\vec{F} = \frac{d\vec{p}}{dt} = m \frac{d\vec{v}}{dt} \quad (\text{B.3})$$

Considerando un conjunto de $n + 1$ muestras de posición $\{\vec{x}_0, \dots, \vec{x}_n\}$ y un intervalo de muestreo constante Δt , entonces:

$$\vec{F}_i = m \frac{\Delta \vec{v}_i}{\Delta t} = \quad (\text{B.4})$$

$$= m \frac{\vec{v}_i - \vec{v}_{i-1}}{t_i - t_{i-1}} \quad \forall i \in \{0, \dots, n\} \quad (\text{B.5})$$

Y sustituyendo en la ecuación B.5 la velocidad por $\vec{v}_i = (\vec{x}_i - \vec{x}_{i-1}) / (t_i - t_{i-1})$ se obtiene que:

$$\vec{F}_i = m \frac{\vec{x}_i - 2\vec{x}_{i-1} + \vec{x}_{i-2}}{(\Delta t)^2} \quad (\text{B.6})$$

Con esta expresión se observa que, obviando la masa del dispositivo (en la práctica, masa del brazo mecánico más la masa del brazo del usuario), se puede computar la aceleración instantánea en cada paso de cálculo, con las tres últimas posiciones del la herramienta del háptico \vec{x}_i, \vec{x}_{i-1} y \vec{x}_{i-2} .

B.4 Métodos para el cálculo de la inercia

A continuación se expone un estudio comparativo de distintos métodos que se han diseñado para estimar la aceleración y la velocidad en un mundo virtual con respuesta háptica. El sistema simula un sistema inercial newtoniano, en el que para mover la herramienta es necesario vencer la resistencia de la misma a ser acelerada. En todo sistema inercial se debe cumplir el equilibrio dinámico, expresado por la siguiente ecuación:

$$\sum_{j=1}^n \vec{F}_j - m\vec{a} = 0 \quad (\text{B.7})$$

en la que la suma de todas las fuerzas que actúan sobre un cuerpo debe ser igual a la masa por la aceleración. Por tanto, para acelerar la herramienta será preciso vencer la llamada *fuerza de inercia*, $-m\vec{a}$.

Los resultados de las pruebas se muestran en imágenes que representan la interacción del usuario con el sistema háptico, considerando lo siguiente:

- El usuario realiza el mismo movimiento en todas las pruebas: dos movimientos curvos con cambio en la dirección de giro y un tramo

rectilíneo al final.

- Se visualizan las muestras de posición de la trayectoria recorrida por la herramienta en un segundo con puntos blancos, mostrando 1 de cada 8 posiciones muestreadas.
- Los vectores de aceleración estimados parten de cada una de las posiciones mostrando la magnitud y módulo con una línea blanca.
- La trayectoria y vectores están proyectados en el plano inferior en gris oscuro.
- La esfera verde muestra la posición del puntero háptico en el instante de la captura.
- Tres esferas cyan muestran las tres únicas posiciones \vec{x}_n , \vec{x}_{n-1} y \vec{x}_{n-2} usadas para el cómputo de la aceleración en el momento de la captura.
- Se marca el vector de fuerza en el momento de la captura con una flecha.
- En los métodos que presentan discontinuidades en la dirección de la aceleración se ha atenuado el módulo para que la respuesta en fuerza sea menos abrupta.

La comparativa de los diferentes métodos se ha realizado con un dispositivo háptico PHANTOM[®] Desktop[™] en una estación gráfica de sobremesa.

A continuación se pasa a describir los diferentes métodos de estimación.

B.4.1 Aceleración instantánea

El enfoque inmediato es aplicar la expresión de la ecuación B.6 para calcular el vector de aceleración en cada ciclo, con la muestra actual \vec{x}_n y las dos anteriores \vec{x}_{n-1} y \vec{x}_{n-2} . Como resulta esperable tras lo explicado en la exposición del problema (ver figura B.2) aparece ruido de alta frecuencia, existiendo fuertes discontinuidades en la dirección del vector aceleración.

En la figura B.3 se muestran las aceleraciones obtenidas para una trayectoria curva. Se puede apreciar que aparecen vectores de aceleración perpendiculares a la trayectoria y con fuertes cambios de dirección. La fuerza de inercia así calculada es totalmente incoherente.

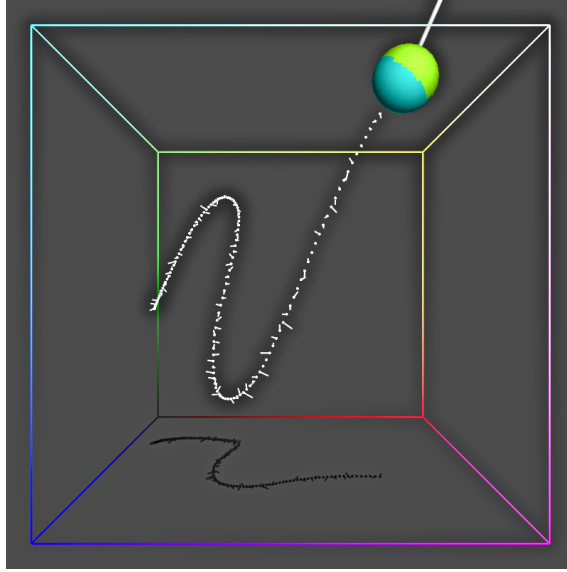


Figura B.3 – Cálculo de la aceleración usando la aceleración instantánea en las últimas tres muestras.

Este método padece además de vibraciones de alta frecuencia. Cuando se mantiene la posición de la herramienta, las pequeñas variaciones de la posición repercuten en vectores de aceleración que cambian de orientación rápidamente y con discontinuidades, lo que hace el método de estimación inútil para calcular la inercia en los movimientos usuales de un dispositivo háptico.

B.4.2 Simplificación de trayectoria

En un segundo plano de aproximación al problema se puede utilizar el concepto de velocidad media para calcular la aceleración:

$$\vec{v}_{media} = \frac{\vec{x} - \vec{x}_0}{t - t_0} \quad (\text{B.8})$$

lo que permite calcular la velocidad en tramos más extensos y eliminar así el ruido de alta frecuencia.

Para esto se almacena únicamente una muestra \vec{x}_n cada k muestreadas, y se calcula una aceleración constante en cada tramo con \vec{x}_n , \vec{x}_{n-1} y \vec{x}_{n-2} . A efectos prácticos se realiza una simplificación de la trayectoria en tramos

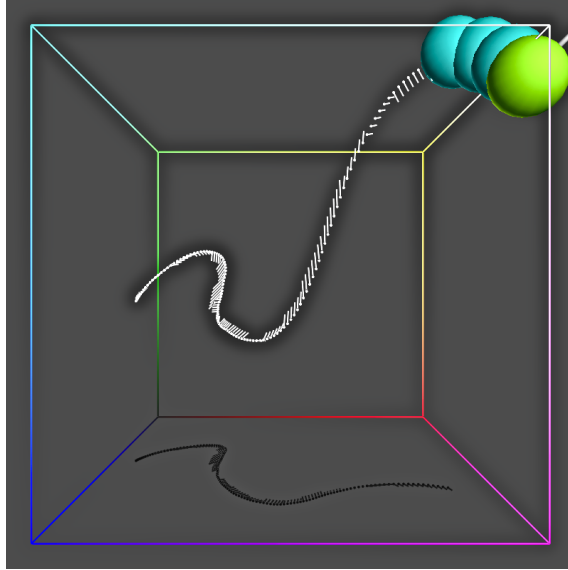


Figura B.4 – Cálculo de la aceleración usando simplificación de trayectoria, tomando una posición de cada 50.

rectilíneos de distancia variable.

Si bien el método es más estable que el anterior al eliminarse el ruido de alta frecuencia, la trayectoria no es derivable en los puntos que coinciden con las muestras \vec{x}_n , lo que produce discontinuidades en la estimación de la aceleración. Esto se puede apreciar en la figura B.4. Las discontinuidades, unidas a la baja frecuencia de actualización de la aceleración, producen fuertes efectos rebote, tanto al estar la herramienta en reposo como en movimiento.

B.4.3 Trayectoria promedio

Si en lugar de calcular las velocidades medias con las posiciones \vec{x}_n muestreadas se utiliza el promedio vectorial de la posición cada k muestras:

$$\vec{x}_n = \frac{1}{k} \sum_{i=1}^k \vec{x}_i \quad (\text{B.9})$$

el efecto rebote se atenúa para pequeñas vibraciones, siendo además estable cuando la herramienta está en reposo. Si las oscilaciones son menores

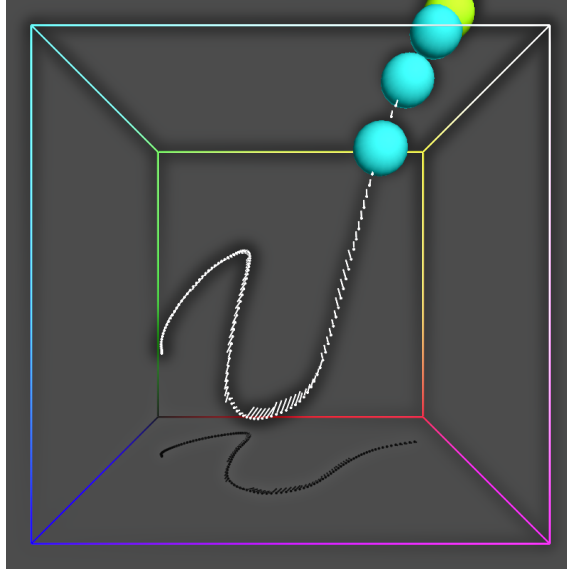


Figura B.5 – Cálculo de la aceleración usando el promedio de la posición cada 50 muestras.

que una cierta distancia, el promedio de la posición los atenúa y acaban convergiendo en 0.

No obstante, se siguen apreciando las mismas discontinuidades que aparecerían en el método anterior cada vez que se toma un nuevo promedio \vec{x}_n y cambia la estimación de \vec{a} : los tramos tienen una estimación constante, que cambia de forma abrupta en los extremos (ver figura B.5).

B.4.4 Aceleración promedio

Para atenuar el ruido también puede estimarse la aceleración como el promedio vectorial de las aceleraciones instantáneas en las k posiciones anteriores:

$$\vec{a}_n = \frac{1}{k} \sum_{i=n-k}^n \vec{a}_i \quad (\text{B.10})$$

lo cual, si se desarrolla la expresión de la aceleración tal y como se hizo para llegar a la ecuación B.6, se comprueba es equivalente a promediar las k velocidades o las k posiciones anteriores y posteriormente calcular la aceleración.

La aceleración promedio, actualizada en cada ciclo, produce una esti-

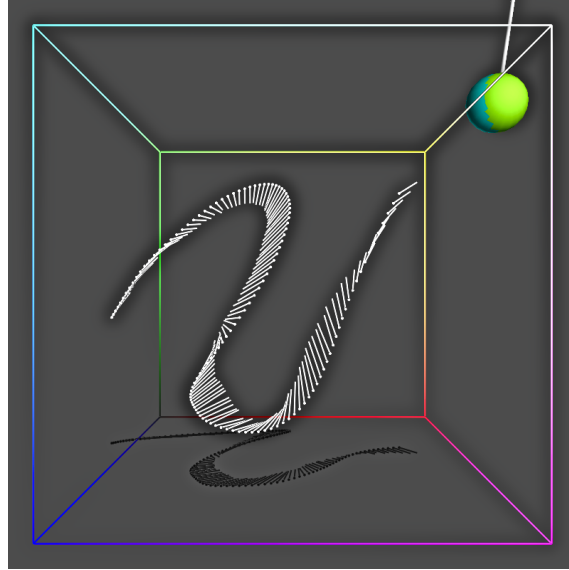


Figura B.6 – Cálculo de la aceleración usando el promedio de la aceleración instantánea en las 100 muestras anteriores.

mación de la inercia relativamente suave y sin cambios bruscos de módulo o dirección, lo cual elimina los rebotes por completo. No obstante, plantea dos problemas:

1. Cuanto más grande es la ventana en la que se calcula el promedio, más se atenúa el ruido de alta frecuencia. Sin embargo, una ventana de mayor tamaño también introduce más retardo, llegando a ser perceptible si el retraso es mayor de 0.1 segundos.
2. Aunque la variación de la aceleración es suave, se siguen produciendo discontinuidades e irregularidades, debido a que no se elimina todo el ruido de alta frecuencia (ver vectores de la parte inferior de la curva en la figura B.6).

B.4.5 Suavizado estático de trayectoria

Dado que el promedio no resuelve de forma definitiva el problema y la reducción de la frecuencia de las muestras produce discontinuidades, se puede reconstruir y simplificar la trayectoria para luego estimar los parámetros ci-

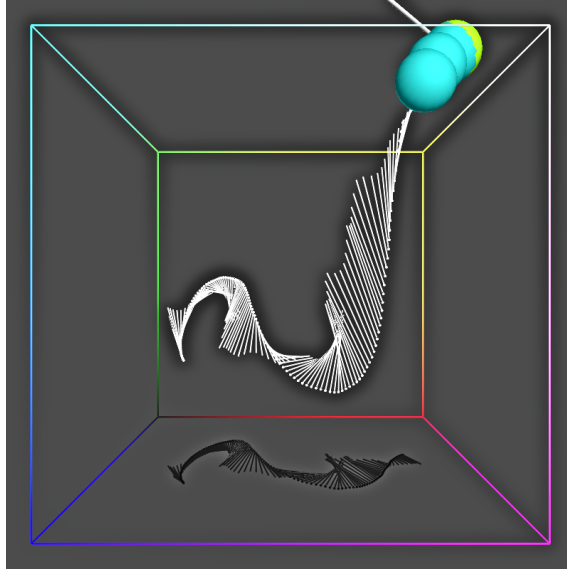


Figura B.7 – Cálculo de la aceleración usando el suavizado estático de trayectoria.

nemáticos. Esto permite la reducción del ruido por una parte y proporciona continuidad en la estimación por otra.

En un primer intento se sustituirá la trayectoria en los $2k$ pasos anteriores por dos tramos rectilíneos. En cada muestreo se actualizan los dos tramos. Si \vec{x}_n , \vec{x}_{n-1} y \vec{x}_{n-2} son las posiciones que definen los dos tramos, $k\Delta t$ es el intervalo de tiempo que transcurre cada k pasos y \vec{x} es la posición actual, calcularemos \vec{a} como:

$$\begin{aligned}
 \vec{x}_0 &\leftarrow \vec{x} \\
 \vec{x}_1 &\leftarrow \vec{x}_{n-1} + i \cdot \frac{\vec{x}_n - \vec{x}_{n-1}}{k\Delta t} \\
 \vec{x}_2 &\leftarrow \vec{x}_{n-2} + i \cdot \frac{\vec{x}_{n-1} - \vec{x}_{n-2}}{k\Delta t} \\
 \vec{a} &\leftarrow m \frac{\vec{x}_0 - 2\vec{x}_1 + \vec{x}_2}{(k\Delta t)^2}
 \end{aligned} \tag{B.11}$$

para todo $i = 1 \dots k$. Se observa que \vec{x}_2 y \vec{x}_1 recorren los dos tramos rectilíneos, hasta que en k llegan al final y coinciden con el \vec{x}_{n-1} y \vec{x}_n , respectivamente. Al final de cada tramo k se actualizan \vec{x}_n , \vec{x}_{n-1} y \vec{x}_{n-2} .

La variación de la aceleración es continua al computarse siempre sobre

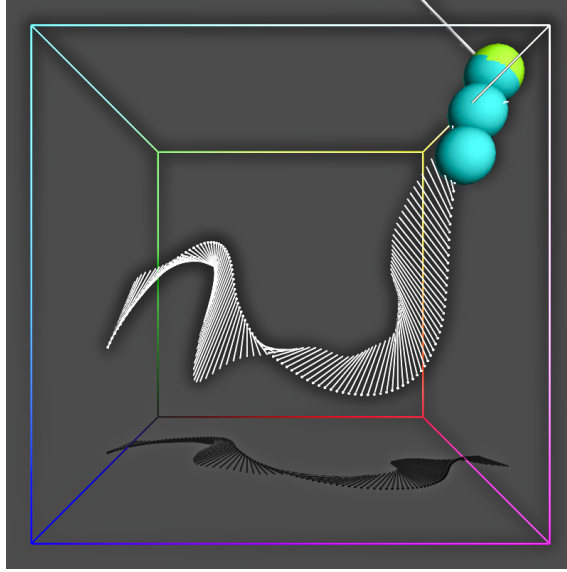


Figura B.8 – Cálculo de la aceleración usando el suavizado dinámico de trayectoria.

la trayectoria simplificada, y sólo es perceptible el ruido introducido por \vec{x}_0 , el cual se mueve con \vec{x} en cada ciclo. Sin embargo, el módulo de \vec{a} oscila (ver figura B.7), ya que la distancia entre \vec{x}_0 , \vec{x}_1 y \vec{x}_2 no es constante en todo el tramo k en las trayectorias curvas. Además no se evitan los rebotes, ya que la trayectoria simplificada sigue teniendo ángulos que acaban por amplificarse.

B.4.6 Suavizado dinámico de trayectoria

Para solventar el problema de los rebotes del método anterior es necesario encontrar una trayectoria simplificada que sea derivable. Esto se puede conseguir actualizando los tramos rectilíneos *en cada uno de los ciclos*, siempre siguiendo la trayectoria marcada por \vec{x} . Si los tramos rectilíneos están definidos por las muestras \vec{x}_n , \vec{x}_{n-1} y \vec{x}_{n-2} , entre las cuales hay una

distancia temporal de $k\Delta t$, los cálculos:

$$\begin{aligned}
 \vec{x}_n &\leftarrow \vec{x} \\
 \vec{x}_{n-1} &\leftarrow \vec{x}_{n-1} + \frac{\vec{x}_n - \vec{x}_{n-1}}{k\Delta t} \\
 \vec{x}_{n-2} &\leftarrow \vec{x}_{n-2} + \frac{\vec{x}_{n-1} - \vec{x}_{n-2}}{k\Delta t} \\
 \vec{a} &\leftarrow m \frac{\vec{x}_n - 2\vec{x}_{n-1} + \vec{x}_{n-2}}{(k\Delta t)^2}
 \end{aligned}
 \tag{B.12}$$

acercan \vec{x}_{n-1} a \vec{x}_n , y \vec{x}_{n-2} a \vec{x}_{n-1} en cada ciclo.

Si bien es cierto que la trayectoria se altera (\vec{x}_{n-1} y \vec{x}_{n-2} no pasan siempre por \vec{x} en las curvas), a la vista del algoritmo y tras las pruebas se observan los siguientes beneficios:

1. Una respuesta inmediata ante un cambio en la trayectoria. Los incrementos en módulo y dirección se reflejan en la estimación de la aceleración al final de cada ciclo, pues \vec{x}_n siempre se actualiza a \vec{x} .
2. Una variación continua y suave de \vec{a} (ver figura B.8).
3. La eliminación total de los rebotes, gracias a la variación continua de la fuerza de inercia.
4. La reducción del ruido de alta frecuencia hasta ser prácticamente imperceptible.

Los resultados de la estimación con suavizado dinámico de trayectoria se pueden apreciar en la captura de interacción mostrada en la figura B.8.

B.5 Conclusiones

Este apéndice muestra un estudio de diferentes métodos diseñados para estimar la aceleración en un entorno virtual con respuesta háptica.

Mediante el análisis y comparación de los resultados se llega a la conclusión de que la estimación de la aceleración por *suavizado dinámico de trayectoria*, la cual realiza una simplificación de la trayectoria trazada por

la herramienta háptica, obtiene una estimación continua, suave y sin efecto rebote para la fuerza de inercia.

El método de suavizado dinámico de trayectoria ofrece una forma robusta y eficiente para la estimación de la aceleración de la herramienta háptica. Esto puede ser especialmente interesante en aplicaciones de simulación de fluidos (médicas o artísticas) o en entornos virtuales que quieran añadir inmersión realista al usar un objeto con peso o inercia.

Apéndice C

Notas de Copyright

Las marcas *AutoCAD*[®] y *Adobe*[®] *Illustrator*[®] que aparecen en el [Capítulo 1](#), así como PHANTOM[®] Desktop[™], mencionada varias veces en el texto, son marcas registradas y todos los derechos pertenecen a sus respectivos dueños, habiendo sido citadas en este trabajo únicamente con fines educativos.

Las Figuras [4.2](#) y [4.4](#) son trabajos derivados de obras con licencia Creative Commons Genérica Attribution-ShareAlike 3.0, habiendo sido obtenidas las obras originales del sitio web Wikimedia Commons bajo los nombres *File:Stress vector.svg* y *File:Stress in a continuum.svg*, respectivamente.

Todas las Figuras de autores distintos del de este trabajo hacen referencia a su autor en el texto al pie de la imagen.

El resto de las Figuras son de creación propia, y quedan a disposición del público con la licencia Creative Commons Genérica Attribution-ShareAlike 3.0.

Índice de símbolos

C	Constante de proporcionalidad elástica, página 86
\mathbf{C}	Tensor de propiedades elásticas, página 58
C_{ep}	Constante de proporcionalidad plástica, página 104
\mathbf{C}_{ep}	Tensor constitutivo elastoplástico, página 68
E	Módulo elástico, página 58
E_{ep}	Módulo de deformación elastoplástico, página 66
\mathbb{E}_σ	Dominio elástico unidimensional, página 65
\mathbb{E}_σ	Dominio elástico tridimensional, página 67
$\partial\mathbb{E}_\sigma$	Superficie de fluencia unidimensional, página 65
$\partial\mathbb{E}_\sigma$	Superficie de fluencia tridimensional, página 67
$F(\sigma, \alpha)$	Función de fluencia plástica unidimensional, página 65
$F(\boldsymbol{\sigma}, \alpha)$	Función de fluencia plástica tridimensional, página 67
$F^\sim(\sigma^\sim, \alpha^\sim)$	Función de fluencia plástica estimada, página 103
$G(\boldsymbol{\sigma}, \alpha)$	Función de potencial plástico, página 67
H'	Parámetro de endurecimiento, página 65
$\mathbf{J}(\mathbf{x}, t)$	Campo gradiente de los desplazamientos, página 51
L_v	Longitud del lado de un vóxel cúbico, página 104

M	Conjunto de muestras de un modelo, página 41
$N(v_{ijk})$	Función vecindad de un vóxel, página 43
$T(\vec{c}, r)$	Función vóxeles incluidos en la herramienta, página 45
V	Conjunto de vóxeles no vacíos de un modelo, página 44
V_t	Volumen del medio continuo en el instante t , página 53
∂V_t	Superficie del medio continuo en el instante t , página 54
X_{ijk}	Punto del recubrimiento, página 47
Ω_0	Configuración del medio continuo en el instante 0, página 50
Ω_t	Configuración del medio continuo en el instante t , página 50
α	Variable de endurecimiento, página 65
α^{\sim}	Variable de endurecimiento estimada, página 104
$\chi(v_i, \vec{u}_i)$	Función chi de distribución del material, página 99
δ_x	Distancia entre muestras en el eje X, página 41
δ_y	Distancia entre muestras en el eje Y, página 41
δ_z	Distancia entre muestras en el eje Z, página 41
δ_{ij}	Delta de Kronecker, página 59
λ	Multiplicador plástico, página 67
λ	Primera constante de Lamé, página 59
λ	Segunda constante de Lamé, página 59
ω	Información de entorno del puntero háptico, página 75
$\phi(\boldsymbol{\sigma})$	Tensión uniaxial equivalente, página 67
$\psi(P)$	Funcion de transferencia espacial, página 139
ρ	Densidad del medio continuo, página 53

σ	Tensión unidimensional, página 62
$\boldsymbol{\sigma}(\mathbf{x}, t)$	Campo tensorial de tensiones de Cauchy, página 57
σ^{\sim}	Tensión estimada, página 103
σ_e	Límite elástico, página 63
$\sigma_f(\alpha)$	Tensión de fluencia, página 65
ε	Deformación unidimensional, página 62
$\boldsymbol{\varepsilon}(\mathbf{x}, t)$	Campo tensorial de deformaciones infinitesimales, página 51
ε^{\sim}	Deformación estimada, página 103
$\boldsymbol{\varepsilon}_e$	Deformación elástica tridimensional, página 66
ε_e	Deformación elástica unidimensional, página 64
$\boldsymbol{\varepsilon}_p$	Deformación plástica tridimensional, página 66
ε_p	Deformación plástica unidimensional, página 64
$\mathbf{a}(\mathbf{x}, t)$	Campo vectorial de aceleraciones del medio continuo, página 54
$\mathbf{b}(\mathbf{x}, t)$	Campo vectorial de fuerzas másicas por unidad de masa, página 53
\vec{c}	Vector posición de la herramienta háptica, página 45
$d(m_{ijk})$	Función valor de propiedad de la muestra, página 41
$e(\mathbf{X}, \vec{c}_0, \vec{u}^{\sim})$	Trasformación espacial elástica, página 141
\vec{f}^{\sim}	Fuerza estimada que genera la deformación, página 87
\vec{f}_h	Fuerza de salida del dispositivo háptico, página 75
$h(\vec{c}, \hat{o}, \omega)$	Función de cálculo o <i>callback</i> háptico, página 75
m_{ijk}	Muestra, página 41

\vec{o}	Vector orientación de la herramienta háptica, página 45
$\mathbf{t}(\mathbf{P}, \mathbf{n})$	Campo vectorial de tensiones internas, página 54
$\mathbf{t}(\mathbf{x}, t)$	Campo vectorial de fuerzas superficiales por unidad de superficie, página 53
t_k	Instante discreto k , página 75
$\mathbf{u}(\mathbf{x}, t)$	Campo de desplazamientos de las partículas del medio continuo, página 51
\vec{u}^{\sim}	Vector de desplazamiento estimado para el cálculo de \vec{f}^{\sim} , página 87
$\mathbf{u}_S^{\sim}(\mathbf{x}, t)$	Campo de desplazamientos estimado y parcial de la superficie, página 80
$\mathbf{u}_e^{\sim}(\mathbf{x}, t)$	Campo de desplazamientos elásticos estimados, página 143
$v(\vec{x})$	Función identidad punto-vóxel, página 43
v_{ijk}	Vóxel, página 43
\vec{w}	Vector de dispersión principal, página 113
\vec{w}_d	Vector de dispersión abajo, página 113
\vec{w}_l	Vector de dispersión izquierdo, página 113
\vec{w}_r	Vector de dispersión derecho, página 113
\vec{w}_u	Vector de dispersión arriba, página 113
\mathbf{x}	Posición de una partícula del medio continuo, página 50
$x(m_{ijk})$	Función posición de la muestra, página 41

Índice alfabético

- 2D, 4
- 3D, 4
- adyacencia, 16
- alargamiento unitario, 58, 59, 68, 107
- bidimensional, 4
- bilineal, *véase* interpolación
- CAD, 4
- campo
 - escalar, 7, 20
 - vectorial, 7
- deformación, 2, 12
- dimensión, 3
- FEM, *véase* modelo de elementos finitos
- filtro, 17
- Free-Form Deformation, 33
- gráfico, 3, 13, *véase también* representación
- interpolación, 18
 - bilineal, 19, 20
 - lineal, 18
- isocurva, 20
- isosuperficie, 20, 21, 24, 31, 32
- Ley de Hook, 29
- Manhattan (distancia), 17
- Marching Cubes, 24
- modelado virtual, 1
- modelo, 3, 13
 - de elementos finitos, 10, 11
 - volumétrico, 13, *véase también* representación volumétrica discreta, 30, 34, 36, 47
- muestra, 7, 10, 12, 13, 18
- píxel, 7
- partición, 18, 49
- proyección, 23
- ráster, 7
- ray tracing, 25
- rejilla, 16, 20
 - anisotrópica, 14, 15
 - estructurada, 13, 14
 - irregular, 14
 - isotrópica, 14, 15, 19
 - no estructurada, 13–15
 - rectilínea, 14, 17, 19
 - regular, 14, 16, 18, 19

representación, 13
 de frontera, 6
 de superficie, 8–10, 25
 volumétrica, 8
 volumétrica continua, 8, 9, 11
 volumétrica discreta, 9–12, 23

representacion
 de superficie, 28

solido deformable, 2

superficie (modelo de), 28

TAC, 4

teselación, 18, 19, 48

tomografía axial computerizada, 4

topología, 46

tridimensional, 4

vóxel, 10, 34, 48, 49

valor de propiedad, 47, 50

vecindad, 16, 49, 131
 18-vecindad, 16
 26-vecindad, 16, 49, 131
 6-vecindad, 16

vectores de dispersión, 119

vectorial, 6, 7

visualización, 23

Voronoi (teselación de), 18, 19, 48

Bibliografía

- [AKO95] Y. Adachi, T. Kumano, and K. Ogino. Intermediate representation for stiff virtual objects. *Virtual Reality Annual International Symposium, 1995. Proceedings.*, pages 203–210, 1995. [32](#), [208](#)
- [APH11] Garrett Aldrich, Dmitriy Pinskiy, and Bernd Hamann. Collision-driven volumetric deformation on the gpu. In N. Avis and S. Lefebvre, editors, *EG 2011: 32nd annual conference of the European Association for Computer Graphics - Short Papers*, pages 9–12, Llandudno, UK, 4 2011. Eurographics Association. [165](#)
- [AS96] Ricardo S. Avila and Lisa M. Sobierajski. A haptic interaction method for volume visualization. In *VIS '96: Proceedings of the 7th conference on Visualization '96*, pages 197–ff., Los Alamitos, CA, USA, 1996. IEEE Computer Society Press. [34](#), [83](#), [209](#)
- [ATTY99] Hideki Arata, Yoshiaki Takai, Nami K. Takai, and Tsuyoshi Yamamoto. Free-form shape modeling by 3d cellular automata. In *Proceedings of the International Conference on Shape Modeling and Applications*, SMI '99, pages 242–, Washington, DC, USA, 1999. IEEE Computer Society. [38](#), [50](#), [110](#), [189](#)
- [Bar84] Alan H. Barr. Global and local deformations of solid primitives. *SIGGRAPH Comput. Graph.*, 18(3):21–30, January 1984. [36](#), [167](#)

- [BS02] Remis Balaniuk and Kenneth Salisbury. Dynamic simulation of deformable objects using the long elements method. In *Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, HAPTICS '02*, pages 58–, Washington, DC, USA, 2002. IEEE Computer Society. [45](#)
- [BW01] Ken Brodlie and Jason Wood. Recent advances in volume visualization. *Comput. Graph. Forum*, 20(2):125–148, 2001. [23](#)
- [CB01] I. F. Costa and R. Balaniuk. Lem-an approach for real time physically based soft tissue simulation. *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, 3:2337–2343 vol.3, 2001. [45](#)
- [CCF94] Brian Cabral, Nancy Cam, and Jim Foran. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In *Proceedings of the 1994 symposium on Volume visualization, VVS '94*, pages 91–98, New York, NY, USA, 1994. ACM. [25](#), [200](#)
- [CD98] S. Cotin and H. Delingette. Real-time surgery simulation with haptic feedback using finite elements. *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, 4:3739–3744 vol.4, 1998. [46](#)
- [Che95] E. Chernyaev. Marching cubes 33: Construction of topologically correct isosurfaces, 1995. [22](#)
- [CHS00] Kwong-Wai Chen, Pheng-Ann Heng, and Hanqiu Sun. Direct haptic rendering of isosurface by intermediate representation. In *VRST '00: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 188–194, New York, NY, USA, 2000. ACM Press. [35](#)
- [CMS88] Michael Chen, S. Joy Mountford, and Abigail Sellen. A study in interactive 3-d rotation using 2-d control devices. In *Proceedings of the 15th annual conference on Computer graphics and*

- interactive techniques*, SIGGRAPH '88, pages 121–129, New York, NY, USA, 1988. ACM. [27](#)
- [CSH03] Kup-Sze Choi, Hanqiu Sun, and Pheng-Ann Heng. Interactive deformation of soft tissues with haptic feedback for medical learning. *Information Technology in Biomedicine, IEEE Transactions on*, 7(4):358–363, Dec. 2003. [44](#), [209](#)
- [CSW⁺03] M. Chen, D. Silver, A. S. Winter, V. Singh, and N. Cornea. Spatial transfer functions: a unified approach to specifying deformation in volume modeling and animation. In *Proceedings of the 2003 Eurographics/IEEE TVCG Workshop on Volume graphics*, VG '03, pages 35–44, New York, NY, USA, 2003. ACM. [167](#)
- [DA04] H. Delingette and N. Ayache. Soft tissue modeling for surgery simulation. In N. Ayache, editor, *Computational Models for the Human Body*, Handbook of Numerical Analysis (Ed : Ph. Ciarlet), pages 453–550. Elsevier, 2004. [46](#)
- [DC04a] Guillaume Dewaele and Marie-Paule Cani. Interactive global and local deformations for virtual clay. *Graph. Models*, 66(6):352–369, November 2004. [39](#), [123](#), [189](#)
- [DC04b] Guillaume Dewaele and Marie-Paule Cani. Virtual clay for direct hand manipulation. In *Eurographics (short papers)*, 2004. [39](#), [109](#), [189](#), [190](#)
- [Del98] H. Delingette. Towards realistic soft tissue modeling in medical simulation. *Proceedings of the IEEE : Special Issue on Surgery Simulation*, pages 512–523, April 1998. [12](#), [37](#), [38](#), [43](#), [50](#)
- [FG99] Sarah F. Frisken-Gibson. Using linked volumes to model object collisions, deformation, cutting, carving, and joining. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):333–348, 1999. [45](#)
- [GSL04] C. González, J. Segurado, and J. LLorca. Numerical simulation of elasto-plastic deformation of composites: evolution of

- stress microfields and implications for homogenization models. *Journal of the Mechanics and Physics of Solids*, 52(7):1573 – 1593, 2004. [106](#)
- [Han97] Chris Hand. A survey of 3d interaction techniques. *Computer Graphics Forum*, 16(5):269–281, 1997. [27](#)
- [HL02] K.C. Hui and H.C. Leung. Virtual sculpting and deformable volume modelling. *Information Visualisation, 2002. Proceedings. Sixth International Conference on*, pages 664–669, 2002. [40](#)
- [HSS⁺05] Markus Hadwiger, Christian Sigg, Henning Scharsach, Katja Bühler, and Markus H. Gross. Real-time ray-casting and advanced shading of discrete isosurfaces. *Comput. Graph. Forum*, 24(3):303–312, 2005. [22](#)
- [JsHsJC00] Farrokh Janabi-sharifi, Vincent Hayward, and Chung shin J. Chen. Discrete-time adaptive windowing for velocity estimation. *IEEE Transactions on Control Systems Technology*, 8(6):1003–1009, Nov 2000. [210](#)
- [KGC⁺96] Rolf M. Koch, Markus H. Gross, Friedrich R. Carls, Daniel F. von Büren, George Fankhauser, and Yoav I. H. Parish. Simulating facial surgery using finite element models. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 421–428, New York, NY, USA, 1996. ACM. [38](#)
- [KLB93] K. Kato, N. S. Lee, and K. J. Bathe. Adaptive finite element analysis of large strain elastic response. *Computers & Structures*, 47(4/5):829–855, 1993. [166](#)
- [KP04] Laehyun Kim and Se Hyung Park. A haptic sculpting technique based on volumetric representation. *Articulated Motion and Deformable Objects*, pages 14–25, 2004. [209](#)
- [KW03] J. Kruger and R. Westermann. Acceleration techniques for gpu-based volume rendering. In *Proceedings of the 14th IEEE*

- Visualization 2003 (VIS'03)*, VIS '03, pages 38–, Washington, DC, USA, 2003. IEEE Computer Society. [26](#)
- [LB03] Adriano Lopes and Ken Brodlie. Improving the robustness and accuracy of the marching cubes algorithm for isosurfacing. *IEEE Transactions on Visualization and Computer Graphics*, 09(1):16–29, 2003. [22](#)
- [LC87] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, volume 21, pages 163–169, New York, NY, USA, July 1987. ACM Press. [21](#)
- [LD07] S.D. Laycock and A.M. Day. A survey of haptic rendering techniques. *Computer Graphics Forum*, 26(1):50–65, 2007. [85](#)
- [Liu02] G. Liu. On velocity estimation using position measurements. In *Proceedings of the American Control Conference*. IEEE, Piscataway NJ, ETATS-UNIS, 2002. [210](#)
- [LS08] Alejandro José León Salas. *IDDG-Octree: una representación híbrida para la visualización y manipulación de volúmenes*. PhD thesis, Universidad de Granada, 2008. [13](#), [15](#)
- [LTV08] Alejandro León, Juan Carlos Torres, and Francisco Velasco. Volume octree with an implicitly defined dual grid. *Computers & Graphics*, 32(3):393–401, 2008. [1](#)
- [LYG02] Karljohan Lundin, Anders Ynnerman, and Björn Gudmundsson. Proxy-based haptic feedback from volumetric density data. In *Proceedings of the Eurohaptic Conference*, pages 104–109. University of Edinburgh, United Kingdom, 2002. [35](#), [209](#)
- [M87] Martti Mäntylä. *An introduction to solid modeling*. Computer Science Press, Inc., New York, NY, USA, 1987. [6](#)

- [MLM03] Kwan-Liu Ma, Eric B. Lum, and Shigeru Muraki. Recent advances in hardware-accelerated volume rendering. *Computers & Graphics*, 27(5):725 – 734, 2003. [200](#)
- [MM07] Patricia Moore and Derek Molloy. A survey of computer-based deformable models. In *Proceedings of the International Machine Vision and Image Processing Conference*, pages 55–66, Washington, DC, USA, 2007. IEEE Computer Society. [12](#), [31](#)
- [Mol07] Daniel Molkenntin. *The Book of Qt 4: The Art of Building Qt Applications*. Open Source Press GmbH, Amalienstr. 45 Rg, 80799 München, Germany, 2007. [193](#)
- [MQW01] Kevin T. McDonnell, Hong Qin, and Robert A. Wlodarczyk. Virtual clay: a real-time sculpting system with haptic toolkits. In *I3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 179–190, New York, NY, USA, 2001. ACM Press. [44](#)
- [MRF⁺96] William R. Mark, Scott C. Randolph, Mark Finch, James M. Van Verth, and II Russell M. Taylor. Adding force feedback to graphics systems: issues and solutions. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 447–452, New York, NY, USA, 1996. ACM Press. [32](#), [209](#)
- [MTPS08] Johannes Mezger, Bernhard Thomaszewski, Simon Pabst, and Wolfgang Straßer. Interactive physically-based shape editing. In *Proceedings of the 2008 ACM symposium on Solid and physical modeling, SPM '08*, pages 79–89, New York, NY, USA, 2008. ACM. [11](#), [42](#), [106](#), [123](#), [190](#)
- [MvdMS10] R.J.E. Merry, M.J.G. van de Molengraft, and M. Steinbuch. Velocity and acceleration estimation for optical incremental encoders. *Mechatronics*, 20(1):20 – 26, 2010. Special Issue on 'Servo Control for Data Storage and Precision Systems', from 17th IFAC World Congress 2008. [211](#)

- [NC10] Suriya Natsupakpong and M. Cenk. Determination of elasticity parameters in lumped element (mass-spring) models of deformable objects. *Graphical Models*, 72(6):61 – 73, 2010. [43](#)
- [NH91] G. M. Nielson and Bernd Hamann. The asymptotic decider: Resolving the ambiguity in marching cubes. In G. M. Nielson and L. J. Rosenblum, editors, *IEEE Visualization 1991*, pages 83–91. IEEE, IEEE Computer Society Press, 1991. [22](#)
- [OBH02] James F. O’Brien, Adam W. Bargteil, and Jessica K. Hodgins. Graphical modeling and animation of ductile fracture. In *SIGGRAPH ’02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 291–294, New York, NY, USA, 2002. ACM Press. [41](#)
- [OdSB06] Xavier Oliver Olivella and Carlos Agelet de Saracibar Bosch. *Mecánica de medios continuos para ingenieros*. Universitat Politècnica de Catalunya, 2006. [60](#), [63](#), [68](#), [72](#), [86](#)
- [PLB⁺01] Hanspeter Pfister, Bill Lorensen, Chandrajit Bajaj, Gordon Kindlmann, Will Schroeder, Lisa Sobierajski Avila, Ken Martin, Raghu Machiraju, and Jinho Lee. The transfer function bake-off. *IEEE Comput. Graph. Appl.*, 21(3):16–22, May 2001. [202](#)
- [PM07] Igor Peterlik and Ludek Matyska. An algorithm of state-space precomputation allowing non-linear haptic deformation modeling using finite element method. In *WHC ’07: Proceedings of the Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 231–236, Washington, DC, USA, 2007. IEEE Computer Society. [47](#)
- [Pre84] Jean H. Prevost. Localization of deformations in elastic-plastic solids. *International Journal for Numerical and Analytical Methods in Geomechanics*, 8(2):187–196, 1984. [89](#), [103](#), [166](#), [191](#)

- [PST09] Francisco Joaquín Rodríguez Prados, Alejandro León Salas, and Juan Carlos Torres. Elastoplastic material deformation with haptic feedback: a physically correct method. In *SIACG 2009 - IV Symposium Iberoamericano de Computación Gráfica*, 6 2009. [188](#)
- [PST10] Francisco Joaquín Rodríguez Prados, Alejandro J. León Salas, and Juan Carlos Torres. Cinemática háptica. In Miguel Chover and Miguel Ángel Otaduy, editors, *Actas del XX Congreso Español de Informática Gráfica (CEIG2010)*, pages 89–96, 2010. [188](#), [200](#)
- [PST11a] Francisco J. R. Prados, Alejandro J. León Salas, and Juan Carlos Torres. Volumetric plastic deformation. In N. Avis and S. Lefebvre, editors, *EG 2011: 32nd annual conference of the European Association for Computer Graphics - Short Papers*, pages 5–8, Llandudno, UK, 4 2011. Eurographics Association. [188](#)
- [PST11b] Francisco Joaquín Rodríguez Prados, Alejandro León Salas, and Juan Carlos Torres. Interactive elastic deformation of 3d images. In F. Silva, D. Gutiérrez, and M. Figueredo, editors, *Proceedings of SIACG2011: V Ibero-American Symposium in Computer Graphics*, pages 169–174, 06 2011. [188](#)
- [RKK97] Diego C. Ruspini, Krasimir Kolarov, and Oussama Khatib. The haptic display of complex graphical environments. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 345–352, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. [33](#), [208](#)
- [SB88] Theodore Sussman and Klaus-Jürgen Bathe. Some advances in the analysis of semideformable media. *Computers & Structures*, 30(1/2):105–112, 1988. [166](#), [169](#), [170](#), [181](#)
- [Sen08] Openhaptics toolkit programmer’s guide. Technical report, SensAble Technologies, 2008. [85](#), [198](#)

- [SMS⁺01] Scott S. Snibbe, Karon E. MacLean, Rob Shaw, Jayne Roderick, William L. Verplank, and Mark Scheeff. Haptic techniques for media control. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, UIST '01, pages 199–208, New York, NY, USA, 2001. ACM. 87
- [SP86] Thomas W. Sederberg and Scott R. Parry. Free-form deformation of solid geometric models. *SIGGRAPH Comput. Graph.*, 20:151–160, August 1986. 31, 37, 56, 167
- [STK99] Y. Sakamoto, K. Tuchiya, and M. Kato. Deformation method for surgery simulation using voxel space automata. In *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, volume 4, pages 1026–1031 vol.4, 1999. 38, 50, 110, 189
- [SZY88] Francis W. Sears, Mark W. Zemansky, and Hugh D. Young. *Física universitaria*. Addison-Wesley Iberoamericana, 1988. 86
- [VA02] Francisco Velasco Anguita. *Representación y visualización de datos volumétricos*. PhD thesis, Universidad de Granada, 2002. 13
- [Win02] Andrew Steven Winter. *VOLUME GRAPHICS Field-Based Modelling and Rendering*. Computer science, University of Wales, December 2002. 167
- [ZS95] C. B. Zilles and J. K. Salisbury. A constraint-based god-object method for haptic display. In *IROS '95: Proceedings of the International Conference on Intelligent Robots and Systems-Volume 3*, page 3146, Washington, DC, USA, 1995. IEEE Computer Society. 33
- [ZSPG08] Jilin Zhou, Xiaojun Shen, E.M. Petriu, and N.D. Georganas. Linear velocity and acceleration estimation of 3 dof haptic interfaces. In *IEEE International Workshop on Haptic Audio visual Environments and Games 2008 (HAVE 2008)*, pages 137–142, 2008. 211