



UNIVERSIDAD NACIONAL DEL SUR

TESIS DE DOCTOR EN CIENCIAS DE LA COMPUTACIÓN

Interacciones en Visualización

María Luján Ganuza

BAHÍA BLANCA

ARGENTINA

2018

Esta página ha sido intencionalmente dejada en blanco.

Prefacio

Esta Tesis se presenta como parte de los requisitos para optar al grado Académico de Doctor en Ciencias de la Computación, de la Universidad Nacional del Sur y no ha sido presentada previamente para la obtención de otro título en esta Universidad u otra. La misma contiene los resultados obtenidos en investigaciones llevadas a cabo en el ámbito del Departamento de Ciencias e Ingeniería de la Computación durante el período comprendido entre el 20/12/2007 y el 16/03/2018, bajo la dirección de la Dra. Silvia Mabel Castro.

María Luján Ganuza



UNIVERSIDAD NACIONAL DEL SUR
Secretaría General de Posgrado y Educación Continua

La presente tesis ha sido aprobada el / / , mereciendo la calificación de(.....)

Esta página ha sido intencionalmente dejada en blanco.

Agradecimientos

Muchas gracias a todos aquellos que contribuyeron directa o indirectamente al desarrollo y escritura de esta tesis.

En primer lugar, quisiera agradecer a la Dra. Silvia Mabel Castro por su apoyo incondicional durante toda mi carrera académica y por su invaluable amistad. Estaré siempre agradecida por el conocimiento que me trasmite día a día, por haberme dado la oportunidad de formar parte de su grupo de investigación y por haberme guiado con tanto entusiasmo y compromiso durante todo este proceso. Su dedicación inquebrantable me ha inspirado no solamente a superarme profesionalmente, sino también personalmente.

Muchas gracias a Sergio Martig, amigo y maestro. Una persona excepcional que me inició en el camino de la docencia y la investigación, y que veló por el desarrollo y crecimiento del VyGLab.

Quisiera agradecer también a mis compañeros del VyGLab. A los miembros presentes y pasados, con quienes he compartido incontables experiencias tanto en el ámbito personal como laboral que indudablemente me han enseñado a trabajar en equipo y a desenvolverme en el ambiente académico y docente. Muchas gracias a Sebastián Escarza y Martín Larrea por leer parte de esta tesis. Sus correcciones y comentarios han sido un gran aporte a este trabajo.

Quisiera agradecer también al comité evaluador de esta tesis, cuyos aportes y sugerencias indudablemente enriquecen este trabajo de investigación.

Parte de este trabajo de investigación se realizó en colaboración con un equipo interdisciplinario integrado por miembros del VyGLab (DCIC, UNS), del INGEOSUR (Instituto Geológico del Sur, UNS-CONICET), la TUWien (Universidad Técnica de Viena) y el VRVis center de Viena. Muchas gracias al Dr. Ernesto Bjerg, a la Dra. Gabriela Ferracutti, a la Dra. Florencia Gargiulo, al Dr. Eduard Gröller y al Dr. Krešimir Matković por su confianza, apoyo, constante dedicación y compromiso con este proyecto.

También quisiera agradecer a las instituciones que han posibilitado el desarrollo de este trabajo de tesis: al Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), a la Comisión de Investigaciones Científicas (CIC), a la Universidad Nacional del Sur, y en especial a todos los miembros del Departamento de Ciencias e Ingeniería de la Computación (DCIC, UNS).

Muchas gracias a los profesores y ayudantes las materias en las que me desempeñé y he desempeñado como asistente (Silvia Castro, Sonia Rueda, Sergio Martig y Sergio Gómez), que me enseñan todos los días a enseñar. Y muchas gracias a mis alumnos, que me enseñan todos los días a aprender.

Finalmente, muchas gracias a mi familia y amigos, por acompañarme siempre y apoyarme en todos mis proyectos profesionales y personales. Nada hubiera sido posible sin ustedes.

Dedico esta tesis a mi gran amiga Pety, cuyos consejos, amistad y compañía han sido fundamentales para alcanzar este logro.

Resumen

En la actualidad, el crecimiento vertiginoso de la cantidad de información genera volúmenes de datos cada vez más grandes y difíciles de comprender y analizar. El aporte de la visualización a la exploración y entendimiento de estos grandes conjuntos de datos resulta altamente significativo.

Es frecuente que distintos dominios de aplicación requieran representaciones visuales diferentes; sin embargo, varios de ellos comparten estados intermedios de los datos, transformaciones, y/o requieren manipulaciones similares a nivel de vistas. Al analizar estos denominadores comunes se plantea la necesidad de contar con un modelo de visualización consistente para todas las áreas de visualización que sea válido para distintos dominios de aplicación. En este contexto se define el Modelo Unificado de Visualización (MUV), un modelo de estados representado como un flujo entre los distintos estados que asumen los datos a lo largo del proceso.

Las características del proceso de visualización determinan que el usuario deba poder interactuar con los datos y sus representaciones intermedias, controlar las transformaciones y manipular las visualizaciones. En este contexto, la definición de una taxonomía de las interacciones en el área de visualización es sumamente necesaria para lograr un mejor entendimiento del espacio de diseño de las interacciones.

El objetivo general de esta tesis consiste en establecer tanto las interacciones como una clasificación de las mismas en el área de visualización que sea válida en los distintos dominios de aplicación. Las interacciones definidas deberán poder aplicarse sobre las distintas transformaciones y estados del proceso de visualización. En este contexto, surge la necesidad de definir una representación para los conjuntos de datos lo suficientemente flexible y orientada al área de visualización, que permita soportar las distintas clasificaciones de datos, atributos, conjuntos de datos y mapeos visuales presentes en la literatura de visualización.

Finalmente, con el objetivo de estudiar y validar los conceptos introducidos en esta tesis, se diseñó e implementó el *SpinelViz* y el *Spinel Explorer*, dos prototipos de visualización de datos geológicos. Para cada prototipo se diseñó un conjunto de interacciones dedicadas que contribuyeron directamente a un avance significativo en el flujo de trabajo de los geólogos expertos. Además, se mostró cómo la clasificación de las interacciones y las operaciones definidas permiten ordenar y facilitar el desarrollo de un sistema de visualización en un determinado campo de aplicación.

Abstract

Nowadays, the vertiginous growth of information generates volumes of data that are increasingly larger and difficult to understand and analyze. The contribution of visualization to the exploration and understanding of these large data sets is highly significant.

Usually, different application domains require different visual representations, however, several of them share intermediate states of data, transformations, and/or require similar manipulations. These common denominators suggest the need for a visualization model that is consistent for all visualization areas and valid for different application domains. In this context, the Unified Visualization Model (MUV) is defined. The MUV is a model of states represented as a flow among the different states assumed by the data throughout the process.

The properties of the visualization process determine that the user should be able to interact with the data and its intermediate representations, control the transformations and manipulate the visualizations. In this context, the definition of a taxonomy of the interactions in the visualization area is extremely necessary to achieve a better understanding of the design space of the interactions.

The overall goal of this thesis is to define the interactions and a classification of interactions in visualization, that is valid in different application domains. The defined interactions will be applied to the states and transformations of the visualization process. In this context, it is necessary to define a representation for the data sets involved in this process. This representation must be sufficiently flexible to support the different classifications of data, attributes, datasets and visual mappings present in the visualization literature.

Finally, with the aim of studying and validating the concepts introduced in this thesis, we designed and implemented the *SpinelViz* and the *Spinel Explorer*, two prototypes for geological data visualization. For each prototype, a set of dedicated interactions that

significantly improved the traditional workflow was designed. In addition, it was exposed how the presented classification and the defined operations allow to order and facilitate the development of a visualization system in a specific application field.

Certifico que fueron incluidos los cambios y correcciones sugeridos por los jurados.

Firma del Director

Esta página ha sido intencionalmente dejada en blanco.

Abreviaturas

BVI	Interacciones Básicas de Visualización
DA	Datos Abstractos
DaV	Datos a Visualizar
DC	Datos Crudos
DMV	Datos Mapeados Visualmente
DV	Datos Visualizados
GIS	Sistema de Información Geográfica
HCI	Human-Computer Interaction, Interacción Humano-Computadora
MUV	Modelo Unificado de Visualización
TD	Transformación de Datos
TF	Transformación de Filtrado
TMV	Transformación de Mapeo Visual
TV	Transformación de Visualización
VA	Visual Analytics
VTK	Visualization Toolkit
WIMP	Windows, Icons, Mouse, Pointer

Esta página ha sido intencionalmente dejada en blanco.

Índice general

Índice de figuras	XIX
Índice de tablas	XXIII
1. Introducción	1
1.1. Contexto	2
1.2. Objetivo General y Aportes	3
1.3. Estructura de la Tesis	4
2. El Proceso de Visualización	7
2.1. Introducción	7
2.2. Trabajo Relacionado	10
2.3. Modelo Unificado de Visualización	15
2.3.1. Los Estados de los Datos	17
2.3.1.1. Datos Crudos (DC)	17
2.3.1.2. Datos Abstractos (DA)	18
2.3.1.3. Datos a Visualizar (DaV)	26
2.3.1.4. Datos Mapeados Visualmente (DMV)	27
2.3.1.5. Datos Visualizados (DV)	39
2.3.2. Las Transformaciones	40
2.4. Múltiples Vistas Coordinadas en el MUV	43
2.5. Conclusión	45
3. Interacciones en Visualización	47
3.1. Definición	48
3.2. Taxonomías de las Interacciones	49

3.2.1.	Taxonomías de las Técnicas de Interacción	52
3.2.2.	Taxonomías de las Dimensiones de las Técnicas de Interacción . . .	67
3.2.3.	Taxonomía de las Tareas	70
3.3.	Interacciones en <i>Visual Analytics</i>	78
3.4.	Conclusión	82
4.	Representación de los Datos	85
4.1.	Introducción	85
4.2.	Trabajo Relacionado	86
4.3.	Clasificación de Los Datos	89
4.4.	Clasificación de los Atributos	92
4.5.	Clasificación de los Conjuntos de Datos	95
4.6.	Representación de los Datos Mapeados Visualmente	102
4.7.	Conclusión	108
5.	Clasificación de Interacciones en el MUV	109
5.1.	Interactuando en el MUV	110
5.2.	Clasificación de Interacciones	112
5.2.1.	Características de los operadores	113
5.3.	Interacciones a Nivel del Programador	114
5.3.1.	Interacciones de Bajo Nivel del Programador	114
5.3.1.1.	Interacciones de Bajo Nivel del Programador Aplicadas a los <i>Datasets</i>	115
5.3.1.2.	Interacciones de Bajo Nivel del Programador Aplicadas al <i>Mapeo Visual</i>	116
5.3.2.	Interacciones de Alto Nivel del Programador	118
5.3.2.1.	Interacciones de Alto Nivel del Programador en los DC .	119
5.3.2.2.	Interacciones de Alto Nivel del Programador en la TD .	119
5.3.2.3.	Interacciones de Alto Nivel del Programador en los DA .	123
5.3.2.4.	Interacciones de Alto Nivel del Programador en la TF .	125
5.3.2.5.	Interacciones de Alto Nivel del Programador en la TMV .	129
5.3.2.6.	Interacciones de Alto Nivel del Programador en la TV .	133
5.3.2.7.	Interacciones de Alto Nivel del Programador en los DV .	134

5.4. Interacciones a Nivel del Usuario	135
5.4.1. Interacciones a Nivel del Usuario Sobre los Conjuntos de Datos . .	137
5.4.2. Interacciones a Nivel del Usuario Sobre el Mapeo Visual	140
5.4.3. Interacciones a Nivel del Usuario Sobre la Vista	142
5.4.4. Interacciones Compuestas a Nivel del Usuario	145
5.5. Conclusión	150
6. Casos de Estudio	153
6.1. Introducción	154
6.2. Campo de Aplicación	156
6.2.1. Flujo de Trabajo Tradicional de Análisis de Minerales del Grupo de los Espinelos	159
6.3. Trabajo Relacionado	161
6.4. <i>SpinelViz</i>	163
6.4.1. <i>SpinelViz</i> en el MUV	165
6.4.2. Interacciones	174
6.5. <i>Spinel Explorer</i>	179
6.5.1. <i>Spinel Explorer</i> en el MUV	184
6.5.2. Interacciones	190
6.6. Conclusión	195
7. Conclusiones y Trabajo a Futuro	199
7.1. Publicaciones	202
7.2. Direcciones Futuras de Investigación	204
Bibliografía	207

Esta página ha sido intencionalmente dejada en blanco.

Índice de figuras

2.1. Distintos Tipos de Visualización	8
2.2. El Proceso de Visualización	9
2.3. Modelo de Referencia de Card <i>et al.</i> [CMS99]	13
2.4. Modelo de Referencia de Chi y Riedl [CR98]	14
2.5. Modelo Anidado de Visualización de Munzner [Mun09]	15
2.6. Modelo Unificado de Visualización [MCFE03]	16
2.7. Clasificación de los Datos de Munzner [Mun14]	23
2.8. Tipos de Celdas en el <i>Visualization Toolkit</i> (VTK) [SML ⁺ 06]	24
2.9. Conjuntos de Datos en el <i>Visualization Toolkit</i> (VTK) [SML ⁺ 06]	26
2.10. Tipos de Ejes	29
2.11. Técnicas de Aumento de Codificación Espacial	31
2.12. Alternativas de Organización de los Datos de Munzner [Mun14]	32
2.13. Alternativas de Organización Espacial	35
2.14. Clasificación de las Marcas Visuales de Munzner [Mun14]	36
2.15. Variables Visuales de Bertin [Ber83]	37
2.16. Canales Visuales de Munzner [Mun14]	38
2.17. Clasificación de Efectividad de los Canales Visuales de Munzner [Mun14]	39
2.18. Estados y Transformaciones en el Modelo Unificado de Visualización	42
2.19. Ramificaciones en el MUV: Múltiples Vistas	44
3.1. Clasificación Jerárquica de las BVIs [CR96]	54
3.2. Taxonomía de Keim [Kei02]	58
3.3. Espacios de Interacción de Ward y Yang [WY04]	65
3.4. <i>Zoom</i> en Distintos Espacios de Interacción [WY04]	66
3.5. Clasificación de Técnicas de Filtrado de Wilkinson [Wil06]	66

3.6. Clasificación de Técnicas de Navegación de Wilkinson [Wil06]	67
3.7. Ciclo de Acción de Norman [Nor88]	67
3.8. Clasificación de Tareas de Brehmer y Munzner [BM13, Mun14]	76
4.1. Clasificación de los Datos en el MUV	90
4.2. Clasificación de los Atributos en el MUV	95
4.3. Clasificación de los Conjuntos de Datos en el MUV	103
4.4. Representación del Sustrato Espacial en el MUV	105
4.5. Representación de las Marcas en el MUV	106
4.6. Representación del Mapeo Visual en el MUV	108
5.1. Ejemplo del MUV en Ejecución	111
5.2. Conjuntos de Datos en el MUV	115
5.3. Interacciones de Bajo Nivel del Programador	117
5.4. Interacciones de Alto Nivel del Programador	120
5.5. Interacciones a Nivel del Usuario	136
5.6. Clasificación de Interacciones en el Contexto del MUV	151
6.1. Prisma de los Espinelos	157
6.2. Prisma de los Espinelos: Patrón de Referencia y Muestra	158
6.3. El Prisma de los Espinelos y sus Proyecciones	158
6.4. Contornos de Barnes y Roeder [BR01]	159
6.5. Flujo de Trabajo Tradicional de Análisis de Espinelos	160
6.6. <i>SpinelViz</i> . Vistas Correlacionadas	164
6.7. <i>SpinelViz</i> : Mapeo Visual.	165
6.8. <i>SpinelViz</i> : Patrones de Referencia	165
6.9. <i>SpinelViz</i> en el MUV: Ramificaciones.	166
6.10. Transformación de Datos en el <i>SpinelViz</i>	168
6.11. Datos Abstractos en el <i>SpinelViz</i>	169
6.12. Transformación de Filtrado en el <i>SpinelViz</i>	169
6.13. Datos a Visualizar en el <i>SpinelViz</i>	170
6.14. Datos Mapeados Visualmente en el <i>SpinelViz</i>	171
6.15. <i>SpinelViz</i> en el MUV	175
6.16. <i>SpinelViz</i> : Zoom Semántico y Proyecciones Integradas.	176

6.17. <i>SpinelViz</i> en el Modelo Unificado de Visualización (MUV): Interacciones.	178
6.18. <i>Spinel Explorer</i>	180
6.19. Categorización Semi-Automática en el <i>Spinel Explorer</i>	181
6.20. Creación de Contornos en el <i>Spinel Explorer</i>	182
6.21. <i>Brushing</i> en el <i>Spinel Explorer</i>	184
6.22. <i>Spinel Explorer</i> en el MUV	185
6.23. <i>Spinel Explorer</i> en el MUV. Mapeo Visual para la Vista de <i>Triplot</i> . . .	187
6.24. <i>Spinel Explorer</i> en el MUV. Mapeo Visual para la Vista de <i>Scatterplot</i> .	188
6.25. <i>Spinel Explorer</i> en el MUV: Mapeo Visual para la Vista de Coordenadas Paralelas	188
6.26. <i>Spinel Explorer</i> en el MUV. Mapeo Visual para la Vista 3D	189
6.27. <i>Spinel Explorer</i> en el MUV. Datos Visualizados	190
6.28. <i>Spinel Explorer</i> en el MUV: Interacciones Generales	195
6.29. <i>Spinel Explorer</i> en el MUV: Interacciones de Categorización	196

Esta página ha sido intencionalmente dejada en blanco.

Índice de tablas

3.1. Síntesis de las Taxonomías de Interacciones en Visualización	52
3.2. Interacciones Básicas de Visualización (BVIs) [CR96]	53
3.3. Taxonomía de Interacciones de Dix y Ellis [DE98]	55
3.4. Taxonomía de Interacciones de Keim [Kei02]	57
3.5. <i>Framework</i> de Ward <i>et al.</i> [WY04, WGK10]	62
3.6. Clasificación de Controles de Wilkinson [Wil06]	63
3.7. Clasificación de Yi <i>et al</i> para las Técnicas de Interacción [YaKSJ07]	64
3.8. Caracterización de Tweedie [Twe97]	68
3.9. <i>Framework</i> de Interacciones de Spence [Spe07]	69
3.10. Taxonomía de Tareas Visuales de Zhou y Feiner [ZF98]	70
3.11. Taxonomía de Tareas Visuales de Amar <i>et al.</i> [AES05]	72
3.12. Espacio de Diseño de Tareas de Schulz.	74
3.13. Clasificación de Tareas de Visualización de Brehmer y Munzner.	78
3.14. Niveles de Interacción en VA de Thomas y Cook [TC05]	80
3.15. Taxonomía de las Interacciones para VA de Kerren y Schreiber [KS12] . . .	82
4.1. Nuestra Propuesta de Clasificación de Datos y su Relación con la Bibliografía	93
4.2. Nuestra Propuesta de Clasificación de los Conjuntos de Datos y su relación con la Bibliografía	104
5.1. Interacciones de Bajo Nivel del Programador Aplicadas a los <i>Datasets</i> . . .	116
5.2. Interacciones de Bajo Nivel del Programador Aplicadas al <i>Mapeo Visual</i> . . .	118
5.3. Interacciones de Alto Nivel del Programador en los Datos Crudos	119
5.4. Interacciones de Alto Nivel del Programador en la Transformación de Datos	122
5.5. Interacciones de Alto Nivel del Programador en los Datos Abstractos. . .	125
5.6. Interacciones de Alto Nivel del Programador en la Transformación de Filtrado	128

5.7. Interacciones de Alto Nivel del Programador en la Transformación de Mapeo Visual	133
5.8. Interacciones de Alto Nivel del Programador en la Transformación de Visualización	134
5.9. Interacciones de Alto Nivel del Programador en los Datos Visualizados .	135
5.10. Interacciones a Nivel del Usuario Sobre los Conjuntos de Datos	140
5.11. Interacciones a Nivel del Usuario Sobre el Mapeo Visual	142
5.12. Interacciones a Nivel del Usuario sobre la Vista	144
5.13. Interacciones Compuestas a Nivel del Usuario	150

Capítulo 1

Introducción

“The greatest value of a picture
is when it forces us to notice
what we never expected to see”

— John Tukey

La visualización es una herramienta muy valiosa que nos permite detectar, inferir y sacar conclusiones sobre las posibles relaciones existentes entre los datos. Esta herramienta es muy utilizada en diversas áreas de las ciencias, la ingeniería y la medicina, entre otras, en las cuales el crecimiento vertiginoso de la cantidad de información genera volúmenes de datos cada vez más grandes y difíciles de comprender y analizar sin un soporte visual adecuado [Cle93]. Es en estos casos donde el aporte de la visualización a la exploración y entendimiento de grandes conjuntos de datos resulta altamente significativo. El procesamiento, la consulta, la exploración y la visualización de distintos conjuntos de datos presenta una serie de interesantes desafíos computacionales, visuales y de interacción.

La visualización es el proceso por el cual se crea una representación visual interactiva a partir de un conjunto de datos. Se trata de un proceso cognitivo mediante el cual, iterativa e interactivamente, el usuario explora el espacio de sus datos [CMS99, War04, Spe07, WGK10]. Aunque resulta frecuente que distintos dominios de aplicación requieran representaciones visuales diferentes, varios de ellos comparten estados intermedios de los datos, transformaciones, y/o requieren manipulaciones similares a nivel de vistas. El análisis de estos denominadores comunes en los distintos procesos de visualización plantea la necesidad de contar con un modelo de visualización consistente para todas las áreas de visualización que sea válido para distintos dominios de aplicación.

Dado que el proceso de visualización puede considerarse una transformación de los datos hacia la vista, es importante clasificar y definir la representación que asumirán los datos en cada etapa y transformación del proceso de visualización. Para esto, es necesario recopilar las definiciones y clasificaciones de los datos disponibles en el contexto de visualización para los distintos campos de aplicación con el objetivo de asegurarnos lograr una representación lo suficientemente abarcativa.

Las características del proceso de visualización determinan que el usuario deba poder interactuar con los datos y sus representaciones intermedias, controlar las transformaciones y manipular las visualizaciones. En este contexto, la definición de una taxonomía de las interacciones en el área de visualización es sumamente necesaria para lograr un mejor entendimiento del espacio de diseño de las interacciones.

1.1. Contexto

En el Laboratorio de Investigación y Desarrollo en Visualización y Computación Gráfica (VyGLab) se ha definido el Modelo Unificado de Visualización (MUV) que brinda tanto al diseñador como al usuario una guía en el proceso de visualización [MCFE03]. Un modelo tal constituye un marco conceptual en el que se pueden definir las interacciones necesarias, pudiéndose determinar sobre qué operandos se opera, qué resultados se obtienen y cómo impacta sobre el proceso general.

La definición de taxonomías de las interacciones en el contexto de visualización es una tarea muy útil para lograr un mejor entendimiento del espacio de diseño de las interacciones. Hasta el momento existe una gran cantidad de trabajos que proponen taxonomías que difieren en diferentes aspectos [BCS96, CR96, DE98, Kei02, WY04, YaKSJ07, Mun14, TC05]. Sin embargo las taxonomías definidas presentan niveles de granularidad significativamente distintos y, en algunos casos, las características de las mismas en relación con su funcionalidad, los objetos de la interacción y sus alcances son parcialmente disjuntas. Estas divergencias plantean la necesidad de definir una taxonomía abarcativa con el objetivo de ofrecer un esquema de interacción consistente y válido independientemente del dominio de aplicación.

1.2. Objetivo General y Aportes

El objetivo general de esta tesis consiste en establecer tanto las interacciones como una clasificación de las mismas en el área de visualización que sea válida en los distintos dominios de aplicación. Las interacciones definidas deberán poder aplicarse sobre las distintas transformaciones y estados del proceso de visualización. De esto surge la necesidad de definir una representación para los conjuntos de datos lo suficientemente flexible y orientada al área de visualización, que permita soportar las distintas clasificaciones de datos, atributos, conjuntos de datos y mapeos visuales presentes en la literatura.

En resumen, los aportes de este trabajo de investigación se detallan a continuación:

1. **Relevamiento exhaustivo y análisis de las distintas taxonomías de las interacciones en el contexto de visualización presentes en la literatura.** Se realizó un relevamiento exhaustivo y un análisis profundo de las distintas clasificaciones de las interacciones y se ordenó el trabajo previo para un mejor entendimiento del estado del arte.
2. **Especificación de una clasificación para los datos y atributos en el contexto de visualización.** La especificación y clasificación de los datos y atributos en el contexto de visualización es un tema muy amplio y complejo y que ha sido tratado por diversos autores. En esta tesis se presenta una clasificación generalizada de los datos y de los atributos que considera las taxonomías más relevantes presentadas en el área de visualización.
3. **Especificación de una clasificación para los conjuntos de datos en el contexto de visualización.** Se presenta una clasificación para los conjuntos de datos en el contexto de visualización. Nuestro objetivo fue el de lograr una clasificación que integrase todos los conjuntos de datos presentes en la literatura para las distintas áreas de visualización.
4. **Especificación de una representación para el mapeo visual.** Se introduce una especificación para el mapeo visual. La estructura de datos definida para representar el mapeo visual debe contemplar tanto el almacenamiento del conjunto de datos a representar, como el de la estructura visual que será utilizada para representar esos datos en pantalla.

5. **Definición de una clasificación multi-nivel de interacciones para visualización en el contexto del MUV.** Encontramos en el Modelo Unificado de Visualización (MUV) un cuerpo troncal de interacciones operacionalmente similares entre los distintos dominios de aplicación analizando las características distintivas de los distintos dominios y/o técnicas particulares, e integrándolas en un modelo conceptual de interacción. La clasificación presentada se compone de dos clases principales, en función del usuario al que están destinadas: las interacciones a nivel del programador y las interacciones a nivel del usuario. Las interacciones a nivel de programador, a su vez, se clasifican en dos grupos: las interacciones de bajo nivel del programador y las interacciones de alto nivel del programador. Las interacciones de bajo nivel del programador se definieron en función de las representaciones para los conjuntos de datos y *mapeo visual* presentadas en el capítulo 4. Las interacciones de alto nivel del programador se definieron en función de las interacciones a bajo nivel del programador. Las interacciones a nivel del usuario se clasificaron en 4 clases principales, en función de la etapa del proceso de visualización que afectan: las interacciones a nivel del usuario sobre los conjuntos de datos, sobre el mapeo visual, sobre la vista y composiciones de estas tres. Las interacciones a nivel del usuario se definieron en base a las interacciones a nivel del programador, especificando para cada una en qué estado o transformación del proceso de visualización se resuelven.
6. **Diseño e implementación de dos prototipos de visualización de datos geológicos y de las interacciones asociadas.** Se diseñó e implementó el *SpinelViz* y el *Spinel Explorer*, dos prototipos de visualización de datos geológicos que sirvieron de base para el estudio y exploración de los conceptos introducidos en esta tesis. Para cada prototipo se diseñó un conjunto de interacciones dedicadas que contribuyeron directamente en un avance significativo en el flujo de trabajo de los geólogos expertos.

1.3. Estructura de la Tesis

En esta tesis se incluyeron los conceptos relevantes, de modo tal que la lectura de la misma sea auto contenida. A continuación se describe su estructura:

- **Capítulo 1 Introducción:** En este capítulo se introducen los objetivos y el mar-

co en el que se desarrolla esta investigación. También se enumeran cuáles son las contribuciones de la misma.

- **Capítulo 2 El Proceso de Visualización:** En este capítulo se introduce el área de visualización en general y las subáreas correspondientes. Se detallan los diferentes modelos presentados hasta el momento con el objetivo de formalizar el proceso de visualización. Finalmente, se describe el Modelo Unificado de Visualización (MUV), que servirá de marco para los conceptos introducidos en este trabajo de investigación.
- **Capítulo 3 Interacciones en Visualización:** En este capítulo se provee una definición del concepto de interacción en el contexto de visualización y se detalla el trabajo previo realizado en este contexto. Se presenta un relevamiento exhaustivo y un análisis profundo de las distintas taxonomías de las interacciones presentes en la bibliografía.
- **Capítulo 4 Representación de los Datos:** En este capítulo se define una clasificación y representación de los datos, atributos, conjuntos de datos y mapeo visual en el contexto del Modelo Unificado de Visualización (MUV).
- **Capítulo 5 Interacciones en el contexto del MUV:** En este capítulo se define una clasificación multi-nivel de interacciones para el Modelo Unificado de Visualización (MUV). Esta clasificación constituye un modelo conceptual de interacción, distinguiéndose las interacciones operacionalmente similares entre los distintos dominios y/o técnicas particulares.
- **Capítulo 6 Casos de Estudio:** En este capítulo se presentan dos prototipos de visualización de datos geológicos que sirvieron de base para el estudio y exploración de los conceptos introducidos en esta tesis. Para ambos prototipos se describe la representación de los datos utilizados en cada etapa y en cada transformación del proceso, como así también el conjunto de interacciones específicas incorporadas, las cuales se describen en función del proceso y de la clasificación de interacciones presentada en el capítulo 5.
- **Capítulo 7 Conclusiones y Trabajo Futuro:** En este capítulo se presentan las conclusiones de nuestro trabajo, como así también el trabajo futuro a realizar.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 2

El Proceso de Visualización

“The purpose of visualization is insight, not pictures.”

— Ben Shneiderman

2.1. Introducción

El objetivo común de los distintos procesos de visualización es la obtención de representaciones visuales interactivas con el propósito de ampliar la adquisición y el uso del conocimiento. Según la naturaleza y las características de la información a visualizar podemos hablar de dos grandes áreas de Visualización. Estas son:

Visualización Científica: Tiene como característica la visualización de datos científicos, típicamente datos físicos. El sustrato espacial se encuentra presente en los datos a visualizar. Dentro de este campo a su vez se distinguen:

- Visualización de Volúmenes: Representación, manipulación y *rendering* de datos volumétricos.
- Visualización de Flujos: Representación, manipulación y *rendering* de datos vectoriales y tensoriales.
- Geovisualización: Representación, manipulación y *rendering* de datos espaciales, en los que el sustrato espacial lo constituye un mapa. El mapa se refiere a representaciones de datos geográficos, abarcando tanto a toda la representación de la Tierra, como a partes de ésta.

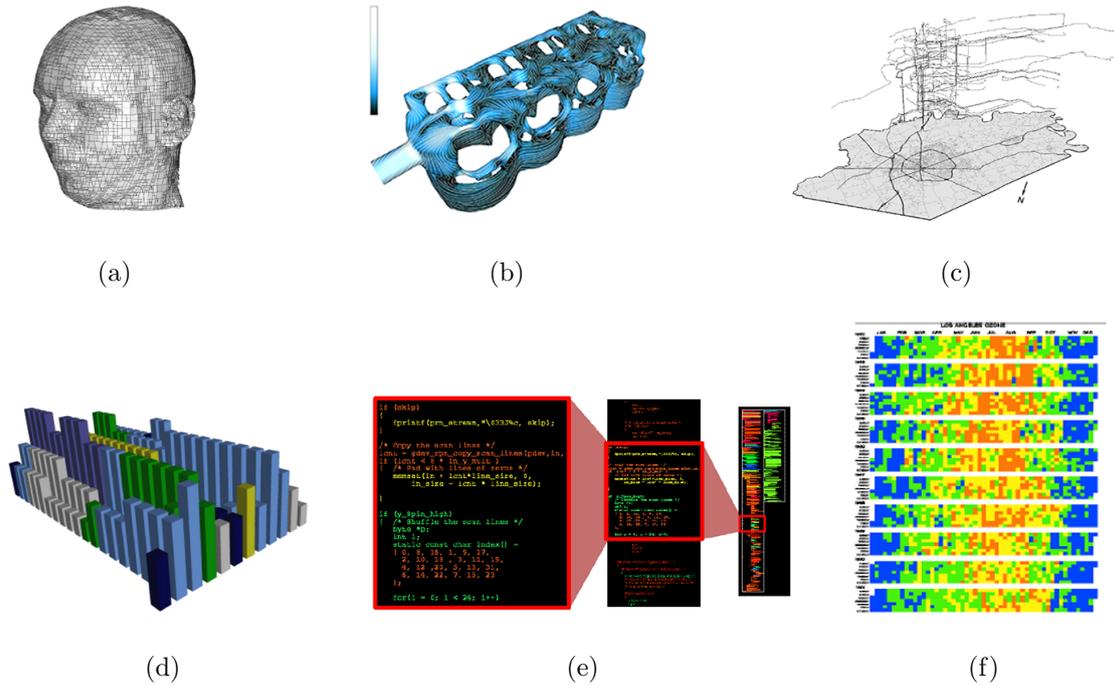


Figura 2.1: Ejemplos de Visualización Científica: (a) visualización de volúmenes aplicado a datos médicos [SMW94], (b) visualización de flujos [SLCZ09] y (c) geovisualización [KLM⁺04]. Ejemplos de Visualización de Información: (d) visualización de algoritmos, donde cada barra representa una línea de texto, el color está mapeado al tipo de estructura de control que la contiene y la altura al nivel de anidamiento [MFM03], (e) visualización de programas [BE96] y (f) representación del nivel de concentración de ozono sobre Los Ángeles en un período de diez años [Spe07].

Visualización de Información: Visualización de datos abstractos, no basados en lo físico. Los datos no poseen un mapeo espacial inherente.

La distinción entre Visualización Científica y Visualización de Información se basa en la naturaleza de los datos a visualizar. En Visualización Científica se trabaja con datos físicos, datos que tienen una geometría inherente. En contraste, en Visualización de Información se manejan datos abstractos en los cuales el mapeo espacial puede estar directamente ausente. En la figura 2.1 se dan ejemplos de las distintas áreas de visualización mencionadas.

Intuitivamente, cualquier proceso de visualización puede pensarse como una transformación de los datos en una representación visual; es un proceso cognitivo en el cual el usuario tiene el poder de interactuar para lograr el objetivo buscado (ver figura 2.2). Aunque resulta frecuente que distintos dominios de aplicación requieran representaciones

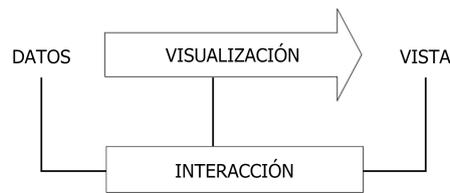


Figura 2.2: El Proceso de Visualización.

visuales diferentes, varios de ellos pueden compartir estados intermedios de los datos, o requerir manipulaciones similares a nivel de vistas o incluso necesitar de las mismas transformaciones de los datos. Es claro que los requerimientos impuestos por cada dominio pueden llegar a ser dramáticamente diferentes; sin embargo, como expresáramos previamente, de un análisis cuidadoso de los mismos podemos encontrar denominadores comunes:

- En uno de los extremos de todo proceso de visualización se encuentran los datos propios del dominio de aplicación y, en el otro, una representación abstracta de esos datos en pantalla. Las propiedades distintivas de los datos manejados en los extremos del proceso permite hablar de separación entre datos y vista. La separación entre datos y vista se hace más evidente en la Visualización de Información dado que los datos a ser mostrados carecen usualmente de un mapeo espacial inherente.
- Es evidente que entre los dos extremos mencionados los datos sufren sucesivas transformaciones.
- Para obtener un buen *insight*¹ de los datos, el proceso de visualización debe ser **netamente interactivo**.
- Para que los usuarios puedan interactuar de manera efectiva debe ser claro y natural conocer **sobre qué se está trabajando y cuáles son los efectos de las posibles interacciones** propuestas.

En función de lo expuesto planteamos la necesidad de contar con un modelo de visualización consistente para todas las áreas de visualización que sea válido para los diferentes dominios de aplicación. Esta consistencia permitirá a los usuarios lograr interacciones

¹*Insight* es un término utilizado en Psicología proveniente del inglés que se puede traducir al español como “visión interna” o más genéricamente “percepción” o “entendimiento”. Mediante un *insight* el sujeto capta, internaliza o comprende, una verdad revelada.

efectivas en el proceso de visualizar datos provenientes de distintos dominios de aplicación, basándose en un único modelo mental. De esta manera se facilitará la interacción con el sistema disminuyendo así el *gulf de ejecución*² - la diferencia entre las intenciones del usuario y las acciones permitidas.

Un sistema del tal porte no sólo beneficia a los usuarios del sistema sino también a los diseñadores. Al momento de extender un sistema existente para incluir nuevos dominios de aplicación, los diseñadores poseen un marco de referencia que define cuáles son las transformaciones que deben sufrir los datos y cuál es el conjunto básico de operaciones que deberán proveer. El modelo debe asistir tanto a los usuarios como a los diseñadores en la evaluación y ejecución de las tareas adecuadas para alcanzar las metas propuestas, es decir, se debe disminuir el *gulf de evaluación*³, logrando así que la realimentación provista por el sistema sea directamente interpretable en términos de las intenciones y expectativas del usuario.

2.2. Trabajo Relacionado

El campo de la Visualización Científica, tal como lo conocemos hoy en día, nació en 1987 [McC88] y algunos de los primeros marcos propuestos planteaban el uso de un *pipeline*⁴ de visualización para el manejo de la entrada, transformación, visualización y registro de los datos [Hae88, LAC⁺92, Nor98].

A lo largo de los años, el *pipeline* de visualización ha provisto una estructura clave en el desarrollo de muchos sistemas de visualización [Mor13], tales como el *Application Visualization System* (AVS) [UFK⁺89], Datavis [Hil91], apE [Dye90], Iris Explorer [Fou95], VISAGE [SLMV92], OpenDx [AT95], SCIRun [PJ95], y el *Visualization Toolkit* (VTK) [SLM04].

Estructuras de *pipelines* análogas se han utilizado extensivamente en campos relacio-

²El *gulf de ejecución*, término introducido por Norman [Nor88], es el grado en que las posibilidades de interacción de un sistema informático se corresponden a las intenciones del usuario y con lo que el usuario percibe qué es posible hacer con el sistema.

³El *gulf de evaluación*, también introducido por Norman [Nor88], es el grado en que el sistema proporciona representaciones que pueden ser percibidas e interpretadas directamente en función de las expectativas e intenciones del usuario.

⁴La arquitectura en *pipeline* consiste en transformar un flujo de datos en un proceso que comprende varias fases secuenciales, siendo la entrada de cada una la salida de la anterior.

nados tales como computación gráfica [Hae88, Kas92], *shaders* para renderizado [AW90, Coo84, Per85], y procesamiento de imágenes [KS94, WR90, JMIntISC].

Herramientas de visualización como Paraview [HA04], VisTrails [BCC⁺05], y Mayavi [RV11] permiten a los usuarios finales construir *pipelines* de visualización mediante interfaces gráficas. Por otra parte, el *pipeline* de visualización es utilizado internamente en una gran cantidad de aplicaciones incluyendo VisIt [vis05], VolView (Kitware Inc.), Osirix [RSR04] y 3D Slicer [PHK04].

En 1996 Schroeder *et al.* presentaron su *Visualization Toolkit (VTK)* [SML96] [SLM04] [SML⁺06] que utilizó con éxito un Modelo de Flujo de Datos para generar visualizaciones en el contexto de Visualización Científica. El modelo utilizado en el VTK consiste en una red de procesos que puede contener varias fuentes y sumideros. Cada paso intermedio de esa red es un filtro que tiene entradas y/o salidas de determinado tipo.

Conceptos similares han llevado a la construcción de modelos para solucionar este tipo de problemas. Lee y Grinstein [LG95] presentaron un modelo conceptual para la exploración visual de bases de datos que describe el proceso de análisis como un conjunto de transformaciones *valor a valor*, *valor a vista* y *vista a vista*.

Foley *et. al* [F⁺90] presentan un *framework* de interacción de usuario, describiendo tres niveles en el diseño de interfaces: léxico, sintáctico y semántico. En el nivel léxico hacen referencia a cómo las entradas y salidas son derivadas de las funciones básicas del *hardware*. El nivel sintáctico consiste en un conjunto de reglas para combinar las primitivas de entrada o las unidades de salida, descritas en el nivel léxico, formando secuencias ordenadas de entradas o salidas. El nivel semántico se enfoca sobre las tareas, es decir sobre las secuencias de acciones con un significado particular. La realización de una tarea puede ser concretada de varias formas distintas, cada una con su propia sintaxis.

Chuah y Roth [CR96] extendieron el *framework* de interacción de usuario presentado por Foley *et.al.* [F⁺90] incorporando las Interacciones Básicas de Visualización (BVI). Estas BVI son presentadas como una caracterización más detallada de los filtros de datos en el contexto de Visualización de Información. Adicionalmente, Chuah y Roth [CR96] presentaron una taxonomía básica de clasificación de BVIs que, a pesar de no ser suficiente para un proceso de visualización general, fue la base de futuros diseños de espacios de interacción. Esta taxonomía se retoma con más detalle en la sección 3.2.1 del capítulo 3.

Un problema similar presenta el *framework* presentado por Tweedie [Twe97], que

consiste en un modelo de transformación de datos similar al introducido por Lee y Grinstein [LG95]. Éste consiste en un modelo simple de interacciones que básicamente clasifica las interacciones basadas en la cantidad de control que el usuario tiene sobre el proceso. El modelo de interacciones de Tweedie se retoma con más detalle en la sección 3.2.2 del capítulo 3 de esta tesis.

En 1997, Card y Mackinlay presentaron una *Taxonomía orientada a Flujo de Datos* [CM97]. Este modelo fue secuencialmente expandido en [CMS99] y divide el campo de visualización en múltiples categorías, tales como Visualización Científica, GIS, gráficos y tablas multi-dimensionales, nodos y conexiones, árboles, textos, etc. En este modelo se enfatiza en los procesos que constituyen un proceso de visualización. El modelo consiste en una sucesión de transformaciones; éstas se representan mediante nodos y las aristas entre los mismos indican la dirección del flujo de datos entre las transformaciones (ver figura 2.3). Se distinguen tres tipos de transformaciones básicas, a nivel de las cuales se plantean las interacciones de los usuarios:

- Transformación de Datos: Esta transformación es la que permite el ingreso al *pipeline* de los datos provenientes de distintos dominios de aplicación y los lleva a una representación tabular interna.
- Mapeo Visual: Esta transformación convierte las tablas de datos en estructuras visuales. Estas estructuras visuales combinan el sustrato espacial, las marcas y las propiedades gráficas o canales visuales.
- Transformación de Vistas: Esta transformación crea una vista a partir de las estructuras visuales, especificando parámetros gráficos tales como posición, escalado, y *clipping*, combinándolos con las características expresadas en la transformación anterior.

Este modelo introduce las interacciones como parámetros de control de las transformaciones descritas arriba; sin embargo, si bien constituye un espacio de diseño muy rico, su modelo de interacción es incompleto.

En 1999, Chi y Riedl presentan un *Modelo de Estados* [CR98] que representa explícitamente los sucesivos estados que asumen los datos a lo largo del proceso y las transformaciones que los generan. En este último, los estados están representados por nodos y las

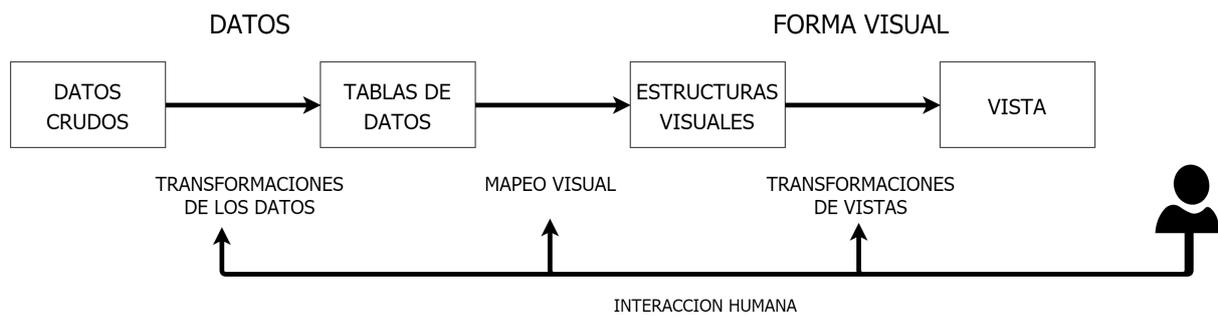


Figura 2.3: Modelo de Referencia de Card *et al.* [CMS99].

transformaciones por las aristas que conectan un estado con el próximo en el *pipeline* (ver figura 2.4). En su modelo, Chi y Riedl distinguen cuatro estados y tres transformaciones:

Estados :

- Datos: Son los datos crudos que constituyen los datos de entrada al proceso de visualización. Proviene de los distintos dominios de aplicación.
- Abstracción Analítica: Son los datos procesados que aún no son mapeables visualmente, pero incluyen toda la información presente en los datos crudos que el usuario está interesado en visualizar. Chi y Riedl se refieren a este estado como *datos sobre datos*, o metadatos.
- Abstracción Visual: Se trata de la información que ya es mapeable y visualizable en pantalla mediante la aplicación de una técnica de visualización.
- Vista: Es el producto final del proceso, donde el usuario ve e interpreta las imágenes presentadas.

Transformaciones :

- Transformación de Datos: Proceso de transformación que genera algún tipo de abstracción a partir de los datos crudos.
- Transformación de Visualización: Proceso que, tomando como entrada una abstracción analítica, la lleva a alguna forma de abstracción visual conteniendo información visual.
- Transformación de Mapeo Visual: Proceso que toma datos que se encuentran en un formato visualizable y los presenta al usuario en una vista gráfica.

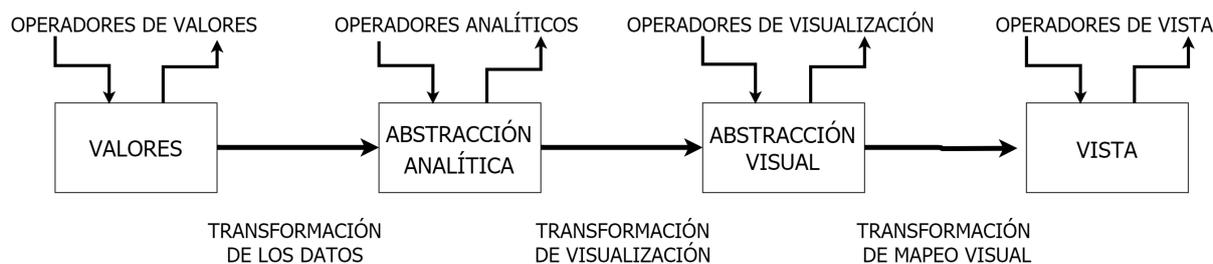


Figura 2.4: Modelo de Estados de Chi y Riedl [CR98].

Adicionalmente, en cada etapa se definen operadores que no modifican las estructuras de datos subyacentes; éstos son llamados *operadores aplicados a las etapas*. Hay cuatro tipos de operadores aplicados a las etapas, uno por cada etapa: operadores de valores, operadores analíticos, operadores de visualización y operadores de vista. En 2001, Duke [Duk01] plantea la necesidad de trabajar con un modelo de referencia general que no se base en las diferencias pragmáticas entre la naturaleza de los datos correspondientes a los campos de la Visualización Científica y de la Visualización de Información. En su trabajo, describe un modelo de datos conceptual que provee cierto nivel de integración entre conjuntos de datos científicos y datos de origen más abstracto. Utiliza el *framework* VTK de Schroeder *et al.* [SML96], hasta el momento aplicado únicamente en el contexto de Visualización Científica, para implementar técnicas de visualización de grafos.

En 2009, Munzner [Mun09] presentó un Modelo Anidado para el Diseño y Validación de Visualizaciones (ver figura 2.5). Este modelo consta de cuatro niveles. En el primer nivel se caracteriza el problema y los datos de un dominio en particular. En el siguiente nivel se mapea el problema y los datos caracterizados en operaciones y tipos de datos abstractos. En el tercer nivel se diseña la codificación visual y las interacciones. Finalmente, en el cuarto nivel, se crea el algoritmo para ejecutar las técnicas de forma automática y eficiente. La salida de un nivel superior es la entrada del nivel inmediatamente inferior. El reto de esta anidación es que un error en un nivel superior inevitablemente se propaga en cascada a todos los niveles descendentes, tal como indican las flechas en la figura 2.5. Si se toma una mala decisión en la etapa de abstracción, entonces incluso una codificación visual y un diseño de algoritmo perfectos no crearán una visualización que resuelva el problema deseado. Munzner presenta un conjunto de metodologías de evaluación sugeridas para cada nivel, seleccionadas en función de los desafíos que se presentan en cada uno de ellos en cuanto a la validación. Resulta claro que tanto la Visualización Científica

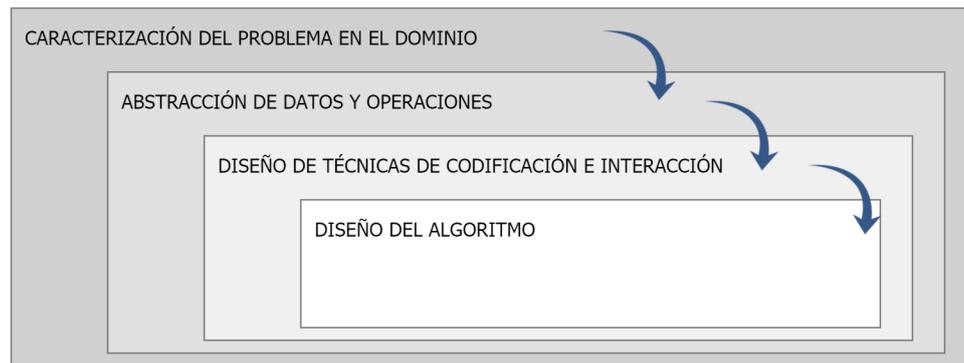


Figura 2.5: Modelo Anidado de Visualización de Munzner [Mun09].

como la de Información comparten el mismo objetivo: la comunicación visual. Desde el punto de vista conceptual se parte de un conjunto de datos y se pretende obtener una representación visual de los mismos, proveyendo mecanismos que favorezcan su exploración. El hecho de que en los datos exista un mapeo espacial inherente o que la asignación de las posiciones en el espacio forme parte del proceso no parece justificar un tratamiento separado, por lo menos desde el punto de vista del modelo de referencia. De hecho, nada garantiza que en el caso de visualizaciones de datos científicos se respete exactamente la información espacial existente en los datos para su representación visual, ya que puede ser útil mostrar los datos con otra organización espacial que contribuya o facilite su análisis.

Todo lo anterior nos ha motivado en la búsqueda de un modelo unificado de visualización, que brinde un único marco de referencia independientemente de la naturaleza de los datos del campo de aplicación.

Se definió entonces un modelo en el que es posible describir las transformaciones y modelar los estados intermedios de los datos, para poder enfocarnos sobre los operadores o sobre los operandos favoreciendo así el entendimiento de las posibles interacciones.

2.3. Modelo Unificado de Visualización

La propuesta consiste en único modelo unificado de visualización [MCFE03] que se pueda enfocar tanto en los procesos como en los estados de los datos y que sea aplicable a cualquier tipo de visualización independientemente del campo particular de origen de los datos. En este modelo quedan explícitamente representadas las operaciones provistas y los operandos sobre los que se pueden aplicar, conjuntamente con la secuencia de transformaciones propias del proceso. Un modelo tal constituye un marco conceptual en

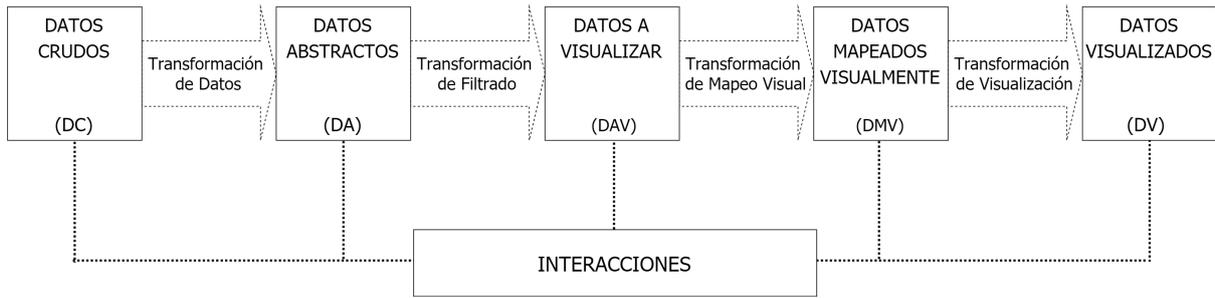


Figura 2.6: Modelo Unificado de Visualización (MUV) [MCFE03].

el que se pueden definir las interacciones necesarias, pudiéndose determinar sobre qué operandos se opera, qué resultados se obtienen y cómo impacta sobre el proceso general. El MUV favorece que el usuario pueda abstraerse de las técnicas particulares, logrando ubicarse en el proceso general, habilitándolo a determinar sobre qué etapa necesita actuar para obtener los resultados deseados.

Para lograr una arquitectura que cumpla con las propiedades mencionadas es necesaria la consolidación de un modelo de referencia para los procesos de visualización. El Modelo Unificado de Visualización (MUV) es un modelo de estados representado como un flujo entre los distintos estados que van asumiendo los datos a lo largo del proceso. En el esquema planteado, los nodos representan los estados de los datos y las aristas, las transformaciones necesarias para pasar de un estado al próximo (ver figura 2.6).

El hecho de que cada nodo represente un estado de los datos permite definir cuáles son las operaciones disponibles sobre ese nodo, así como el tipo de los resultados de su aplicación. Esto mismo sucede con las aristas, que representan las transformaciones entre estados. De este modo, el usuario puede interactuar a nivel de las transformaciones siendo consciente del resultado de tales interacciones. En resumen, el modelo permite modelar el proceso, operar sobre los nodos y operar sobre las aristas. Además, los resultados de operar sobre un elemento del modelo se propagará en las aristas y nodos en el camino a la vista.

Como puede observarse en la figura 2.6 la interacción del usuario está presente a nivel de todas las etapas y de todas las transformaciones. La interacción del usuario es la que controla los parámetros de las transformaciones y manipula los datos en los distintos estados. A continuación se describirán los sucesivos estados que van asumiendo los datos y las transformaciones que los llevan a los mismos. En el capítulo 5 se describirá y analizará la interacción del usuario a lo largo del Modelo Unificado de Visualización, identificando

las interacciones disponibles en cada etapa y en cada transformación, y sus repercusiones en las restantes.

2.3.1. Los Estados de los Datos

La distinción entre los estados que van atravesando los datos a lo largo del proceso no siempre obedece a diferencias estructurales, sino también a diferentes roles que cumplen los datos dentro de la secuenciación de todo el proceso. En algunos casos, la distinción sólo tiene como justificativo la representación conceptual de un determinado estado, siempre teniendo como objetivo la consistencia con el modelo conceptual del usuario.

2.3.1.1. Datos Crudos (DC)

Se corresponde con el estado inicial del proceso de visualización. Son los datos que constituyen la entrada al proceso de visualización, provenientes de distintos dominios de aplicación en un determinado formato. Este conjunto de datos constituye el espacio de información que el usuario quiere explorar.

Los DC son muy diversos y la representaciones que se utilizan son muy variadas dependiendo, en muchos casos, del dominio del problema. Algunos formatos son propios de los distintos campos de aplicación como DICOM⁵ (formato de archivo usado a menudo en ciencias médicas), DEM⁶ (formato de archivo de modelos de elevación digital derivado del U.S. *Geological Survey*), PDB⁷ (formato de archivo que almacena estructuras 3D de macro-moléculas biológicas) o CML⁸ (formato de archivo que describe información química usando XML⁹). También existen varios formatos más generales como los que almacenan geometría y/o atributos de una escena como VRML¹⁰, PLY¹¹, XDF¹², 3DS¹³,

⁵DICOM: *Digital Imaging and Communication in Medicine*.

⁶DEM: *Digital Elevation Models*.

⁷PDB: *Protein Data Bank*.

⁸CML: *Chemical Markup Language*.

⁹XML: *Extensible Markup Language*.

¹⁰VRML: *Virtual Reality Modeling Language*.

¹¹PLY: *Polygon File Format*.

¹²XDF: *eXtensible Data Format*.

¹³3DS: *3D Studio*.

OBJ¹⁴, etc. o los que almacenan imágenes como JPEG¹⁵, BMP¹⁶, PNG¹⁷, etc. Estos son sólo algunos de los cientos de formatos existentes.

Un aspecto importante a tener en cuenta es que los datos que se manejan son discretos, ya sea por características propias de adquisición o por cuestiones relacionadas a la representación de la información.

2.3.1.2. Datos Abstractos (DA)

Se trata de un estado intermedio de los datos en el que éstos están en un formato manejable en el proceso de visualización pero aún no visualizable. Los DC han sido transformados en una representación unificada facilitando así el manejo de los conjuntos de datos en el resto del proceso de visualización, especialmente en el mapeo de las formas visuales. En este estado están presentes todos los datos que el usuario preseleccionó dentro de los DC como potencialmente visualizables. Los DA pueden definirse como la unión de un subconjunto de los DC y de los datos derivados o mejorados. En este estado, además de disponer de los datos de interés, también se dispone de los metadatos generados en la transformación de datos.

Se ha recorrido un largo camino en la búsqueda de una representación unificada para tratar de que la misma sea abarcativa y puedan representarse adecuadamente los datos en el contexto de las distintas áreas de visualización.

Evolución de las Clasificaciones de los Datos Utilizadas en Visualización

La clasificación de los datos que se utilizó desde los inicios de la visualización computacional es un tema muy amplio y complejo y se abordará enfatizando las características que puedan resultar relevantes para su posterior visualización.

En 1946 el estadístico S. Stevens [Ste46] clasificó los datos en función de su uso en Estadística en cuatro tipos:

- Nominal: Es básicamente una función de etiquetado. Sólo se puede decir si un valor es igual o distinto de otro.
- Ordinal: Los valores obedecen a la relación de orden $<$.

¹⁴OBJ: *Wavefront 3D Object File*.

¹⁵JPEG: *Joint Photographic Experts Group*.

¹⁶BMP: *Bitmap Image File*.

¹⁷PNG: *Portable Network Graphics*.

- Intervalo: Los valores de los datos se clasifican en intervalos.
- Razón: Tienen todo el poder expresivo de los números reales. Incluyen un valor 0 de referencia. Podemos expresar por ejemplo que el “Objeto A es el doble de grande que el objeto B.”

Bertin [Ber77] sugiere que los datos se representan bajo dos formas: los *valores de los datos* y la *estructura de los datos*. Los valores de los datos constituyen los objetos de interés. La estructura de los datos está dada por las relaciones existentes entre los valores de los datos. En otras palabras, clasifica los datos en *entidades* y *relaciones*. Al hablar de entidades y relaciones surge la necesidad de nombrar a los atributos, que pueden considerarse como las propiedades de las entidades y de las relaciones en el dominio de aplicación o en el mundo real. Uno de los problemas que se debe enfrentar es precisamente la decisión sobre cuáles son las entidades y cuáles los atributos. Esta distinción no siempre es clara y depende de la intención con la que el usuario manipula los datos y su contexto. Desde el punto de vista de la visualización los atributos y sus características impactan fuertemente en la manera en que los mismos pueden ser visualizados.

En 1999, Card *et al.* [CMS99], en el que se considera el primer libro de Visualización de Información, introducen la noción de *Tabla de Datos*, que combina relaciones entre datos con metadatos que describen esas relaciones. En su trabajo, Card *et al.* clasifican los datos provenientes del dominio de aplicación en tres tipos:

- Nominales: Se trata de conjuntos no ordenados, tales como títulos de películas.
- Ordinales: Corresponde a conjuntos ordenados, tales como la clasificación cinematográfica (ATP,+13,+16,+18), por ejemplo.
- Cuantitativos: Se trata de un valor numérico, como la duración de una película.

En 2004, Ware [War04] clasifica los datos en entidades y relaciones, siguiendo el trabajo de Bertin [Ber77]. Ware define las entidades como los objetos que se desea visualizar, y las relaciones como las estructuras y patrones que relacionan las entidades entre sí. En algunos casos estas relaciones pueden proveerse de forma explícita, pero en otros, el descubrimiento de estas relaciones es el propósito de la visualización. Tanto las entidades como las relaciones tienen atributos. Ware define los

atributos como una propiedad de una entidad que no puede pensarse de forma independiente a la misma.

Posteriormente Munzner [Mun14] presenta una clasificación de conjuntos de datos y atributos más detallada y en el contexto de visualización. De las distintas clasificaciones de datos existentes hasta el momento en la literatura, ésta es sin duda la más completa. En su trabajo, Munzner resalta la importancia de conocer tanto el tipo de los datos a visualizar como su semántica. Ocasionalmente, la semántica de los datos pueden ser inferida correctamente mediante simple observación de la sintaxis de los datos, pero generalmente debe ser provista junto con el conjunto de datos para ser interpretada correctamente. Este tipo de información adicional puede considerarse *metadato*. Munzner define el tipo de un dato como su interpretación estructural o matemática, y presenta una clasificación de datos en tres niveles: el de los datos, el de los conjuntos de datos y el de los atributos. A nivel de los datos distingue cinco tipos de datos:

- Atributos: Un atributo se define como una propiedad específica que puede ser medida, observada o registrada.
- Ítems: Un ítem es una entidad individual discreta.
- Conexiones: Una conexión es una relación entre ítems.
- Grillas: Una grilla especifica la estrategia de muestreo de datos continuos tanto en términos geométricos como en términos topológicos.
- Posiciones: Una posición provee información sobre una locación en un espacio de dos dimensiones o en un espacio de tres dimensiones.

A nivel de los conjuntos de datos, Munzner [Mun14] distingue cuatro tipos (ver figura 2.7(a)):

- Tablas: Las tablas están compuestas por filas y columnas. Cada fila representa un ítem de datos y cada columna un atributo de esos ítems. Cada celda en la tabla se determina por la combinación de una fila y una columna, y contiene el valor asociado a ese par.
- Redes y Árboles: Las redes representan relaciones entre dos o más ítems. Un ítem en una red se representa con un nodo, y una conexión entre dos ítems con

una arista. Un árbol es un tipo especial de red que mantiene una estructura jerárquica.

- Campos: Un campo es conjunto de datos que contiene valores de atributos asociados a celdas. Cada celda contiene medidas o resultados provenientes de un dominio continuo. Conceptualmente es posible tomar infinita cantidad de medidas, de modo que siempre se puede tomar una nueva medida entre dos existentes. Ejemplos de fenómenos continuos que pueden ser medidos en el mundo físico o simulados en un *software* incluyen, por ejemplo, temperatura, presión, velocidad, fuerza, densidad, etc. Consideremos, por ejemplo, un campo que represente un escaneo médico del cuerpo humano que contiene mediciones de densidad de tejido en puntos de muestra ubicados regularmente en un volumen en el espacio 3D. Un escaneo a baja resolución podría contener 262.144 celdas, cada una de las cuales está asociada a una región en el espacio 3D. Las medidas de densidad pueden tomarse a menor distancia (aumentando la cantidad de celdas) para lograr una resolución más alta o a una distancia mayor (reduciendo la cantidad de celdas) para obtener una resolución más baja.
- Conjuntos de Datos Geométricos: Este tipo de conjuntos de datos contienen información sobre la forma de los ítems con información explícita sobre posiciones espaciales. Los conjuntos de datos geométricos son intrínsecamente espaciales y se utilizan en el contexto de tareas que requieran entendimiento de formas. Estos conjuntos de datos no necesariamente tienen atributos. Los conjuntos de datos puramente geométricos son interesantes en el contexto de visualización cuando se derivan o se transforman de modo que requiera considerar una alternativa de diseño. Ejemplos de este tipo de conjuntos de datos son los contornos que se derivan de un campo espacial, la curva que marca un camino, etc.
- Conjuntos: Un conjunto es una colección de ítems sobre los que no hay establecida una relación de orden y que no contiene ítems repetidos.

A nivel de los atributos, Munzner [Mun14] distingue dos tipos fundamentales (ver figura 2.7(b)):

- **Categoricos:** Las constantes de un tipo categórico no tienen un orden implícito, pero a veces pueden presentar una estructura jerárquica. Solo se puede distinguir entonces si dos constantes son iguales o diferentes. Podría imponerse un orden arbitrario externo; sin embargo, este orden no es implícito al atributo en sí mismo; esto sucede por ejemplo, en el ordenamiento de las luces de un semáforo considerando los colores.
- **Ordenados:** A diferencia de los datos categóricos, en las constantes de un tipo de dato ordenado sí hay un orden implícito. Los datos ordenados, a su vez, pueden subdividirse en datos **ordinales** y datos **cuantitativos**.
 - **Ordinales:** Los datos ordinales son aquéllos sobre cuyas constantes existe un orden bien definido; sin embargo, sobre éstas no están definidas las operaciones aritméticas. Un ejemplo de dato ordinal es el talle de una prenda (*small, medium, large, etc.*)
 - **Cuantitativos:** Los datos cuantitativos son aquéllos sobre cuyas constantes hay una relación de orden y sobre éstas se puede aplicar operaciones aritméticas. Por ejemplo peso, temperatura, etc.

A su vez, los datos ordenados pueden ser **secuenciales** o **divergentes**. Los datos ordenados son secuenciales cuando se definen en un rango homogéneo que abarca desde un valor mínimo hasta un valor máximo. Los datos ordinales son divergentes cuando su rango de valores está caracterizado por dos subsecuencias que parten de un punto en común considerado el origen y una de ellas es ascendente y la otra descendente. Por ejemplo, la altura de una montaña es un dato ordenado secuencial, cuyo punto de origen es el nivel del mar; las elevaciones y depresiones en la Tierra están definidas en base a la cota cero que es el nivel del mar: las elevaciones están por encima y las depresiones por debajo de la cota cero.

Adicionalmente, los datos ordenados pueden ser **cíclicos** cuando los valores que toman parten de un valor inicial y retornan al mismo valor en vez de crecer infinitamente de manera secuencial. Muchos tipos de medidas de tiempo son cíclicas, tales como la hora del día, los días de la semana y los meses del año.

En el contexto de Visualización Científica el dominio natural de los datos suele

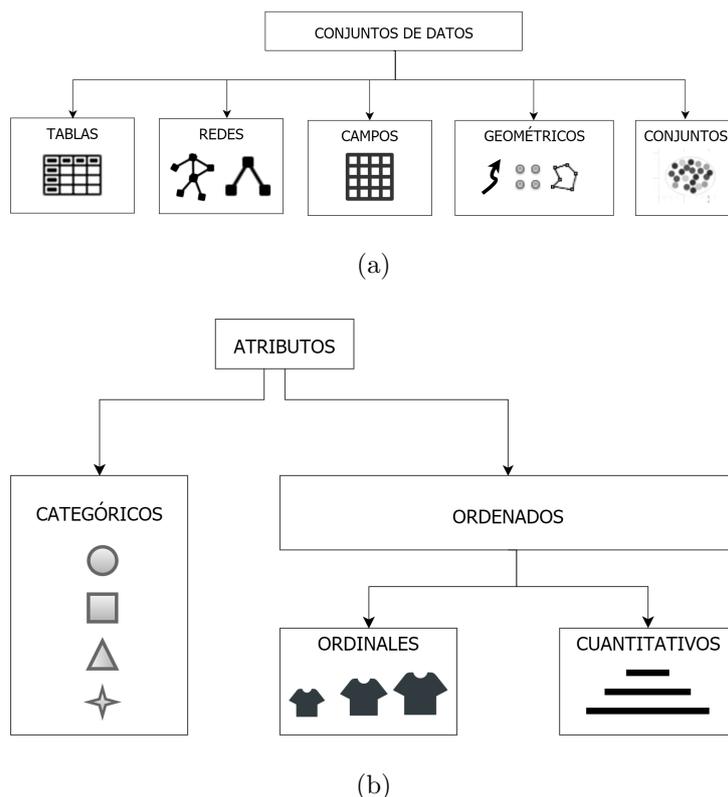


Figura 2.7: Clasificación de los Datos de Munzner [Mun14]. (a) Clasificación de tipos a nivel de los conjuntos de datos. (b) Clasificación de tipos a nivel de los atributos.

ser continuo y debe ser discretizado para poder representarse. El resultado de esta discretización es un conjunto de puntos muestreados que mantienen una relación topológica entre ellos.

En este contexto, Schroeder *et al.* [SML⁺06] presentan una caracterización de datos para el *Visualization Toolkit* (VTK). En ésta, caracterizan los datos en regulares e irregulares. Los datos regulares presentan una relación inherente entre los puntos muestreados. Por ejemplo, si el muestreo se realiza en puntos separados regularmente, entonces no es necesario almacenar todas las coordenadas de los puntos, sino que sólo con almacenar la posición del punto de origen y la distancia entre los puntos es suficiente. Los datos irregulares pueden representar la información de forma más densa cuando ésta cambia rápidamente y menos densa cuando no hay cambios o presenta cambios muy pequeños. Schroeder *et al.* definen los conjuntos de datos como una composición de celdas y puntos. Las celdas especifican la topología del conjunto de datos, mientras que los puntos especifican la geometría del mismo.

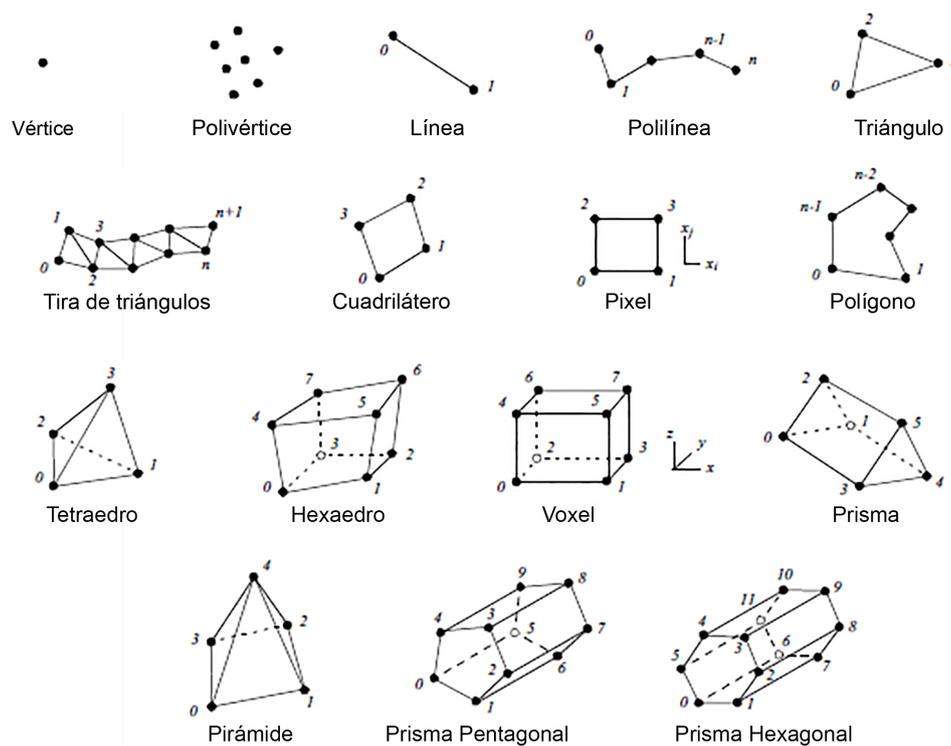


Figura 2.8: Tipos de Celdas en el *Visualization Toolkit* (VTK) [SML⁺06].

Los conjuntos de datos consisten en una o más celdas, que puede ser de distinto tipo y estarán asociadas a una lista ordenada de puntos. Adicionalmente, las celdas tendrán asociada una dimensión. En general, se define una cantidad determinada de variedades de posibles tipos de celdas. Schroeder *et al.* distinguen entre las celdas lineales y las celdas no lineales. En la figura 2.8 se ilustran las celdas lineales disponibles en VTK. Además, VTK soporta un conjunto de celdas no lineales y provee un mecanismo para transformar las celdas no lineales en celdas lineales mediante un proceso de descomposición.

Con respecto a los conjuntos de datos, Schroeder *et al.* distinguen los conjuntos de datos regulares y los irregulares. Consideran que un conjunto de datos es regular si la relación entre los puntos que lo componen se rige por una única función matemática. Este tipo de conjuntos de datos puede ser representado de forma implícita ahorrando espacio de almacenamiento y cómputo. Los conjuntos de datos irregulares, en cambio, deben ser representados de forma explícita ya que no existe un patrón inherente que los describa de forma compacta. Los tipos de conjuntos de datos soportados por VTK se ilustran en la figura 2.9 y son los siguientes:

- **Imagen:** Se trata de una colección de puntos y celdas organizadas en una grilla rectangular y regular. Tanto la topología como la geometría es regular. Este tipo de conjunto de datos también puede ser llamado Grilla Uniforme y abarca los arreglos de puntos (en 1 dimensión), las imágenes (en dos dimensiones) y los volúmenes (en tres dimensiones).
- **Grillas Rectilíneas:** Se trata de una colección de puntos y celdas organizadas en una grilla regular. En este caso, si bien la topología del conjunto de datos es regular, la geometría es parcialmente regular. Es decir, los puntos están alineados sobre un eje de coordenadas, pero el espaciado entre los mismos varía. Estos conjuntos de datos incluyen los *pixels* (en dos dimensiones) y los *voxels* (en tres dimensiones). Para representar la topología basta con especificar las dimensiones de la grilla. Para representar la geometría se especifican las coordenadas geométricas del origen y la separación a lo largo de los ejes. Puede haber una única separación para cada uno de los ejes (Δ_x , Δ_y , Δ_z) o un conjunto de separaciones para cada uno de éstos.
- **Grillas Estructuradas:** Se trata de un conjunto de datos con topología regular y geometría irregular. Las celdas que suelen componer estos conjuntos de datos son los cuadriláteros (en dos dimensiones) y los hexaedros (en tres dimensiones).
- **Puntos No Estructurados:** Se trata de puntos ubicados de forma irregular en el espacio. No tienen una topología asociada y la geometría es completamente no estructurada. Las celdas de tipo vértice y polivértice son usualmente utilizadas para representar este tipo de conjunto de datos.
- **Conjuntos de Datos Poligonales:** Estos conjuntos de datos incluyen los conjuntos de vértices, polivértices, líneas, polilíneas, polígonos y tiras de triángulos.
- **Grillas No Estructuradas:** Es el conjunto de datos más general. Tanto la topología como la geometría son completamente no estructuradas.

También en el contexto de Visualización Científica, Telea [Tel14] define un conjunto de celdas y grillas muy similar al de Schroeder *et al.* [SML⁺06]. Entre los conjuntos de datos identifica la Grilla Uniforme (se corresponde con el tipo de dato *Imagen* de Schroeder *et al.*), la Grilla Rectilínea, la Grilla Estructurada y la no Estructurada.

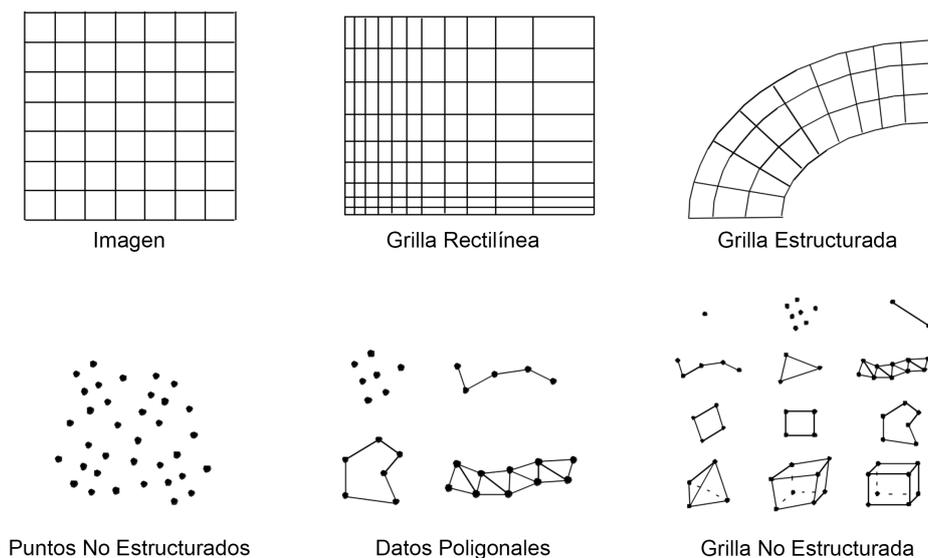


Figura 2.9: Conjuntos de Datos en el *Visualization Toolkit* (VTK) [SML⁺06].

Conceptualmente, podemos considerar que en este estado, los datos están organizados en cualquiera de los conjuntos de datos descriptos, ya sean las tablas, redes, campos, conjuntos o conjuntos geométricos de Munzner [Mun14] o las grillas para datos continuos de Schroeder *et al.* [SML⁺06] o Telea [Tel14]. Con el objetivo de definir una representación unificada y abarcativa definimos una clasificación de datos y conjuntos de datos que incluye los tipos de conjuntos de datos más relevantes en el contexto de visualización. En este contexto, definimos un conjunto de datos genérico llamado *dataset*, que podrá instanciarse con cualquiera de los conjuntos de datos definidos por los autores mencionados. La clasificación propuesta se detalla en la sección 4.5 del capítulo 4 de esta tesis. De un mismo conjunto de DC se pueden generar varios conjuntos de DA distintos, ya sea representados estructuralmente de forma diferente (con *datasets* distintos), o como resultado del procesamiento de un subconjunto distinto de los DC.

2.3.1.3. Datos a Visualizar (DaV)

Los Datos a Visualizar se componen de todos los DA o de un subconjunto de ellos. Son los datos que van a estar presentes en la visualización. En lo que se refiere a la representación de los datos se mantienen las correspondientes al estado anterior (DA).

Todos los datos presentes en este estado intervienen en las visualizaciones que se generen en las etapas posteriores. La forma en que los datos se vayan a representar visualmente dependerá de las etapas posteriores, pero la información presente en este

estado tiene que estar presente en las vistas que se generen.

La distinción entre DaV como un estado independiente de los DA formaliza en el modelo la posibilidad de tener más de un conjunto de DaV simultáneamente. De un mismo conjunto de DA se pueden generar varios conjuntos de DaV. Esta característica permite la exploración de distintas regiones del espacio de información y su comparación. Las interacciones aplicadas sobre este estado, o cualquiera de los siguientes, no influyen sobre los otros conjuntos de DaV existentes.

2.3.1.4. Datos Mapeados Visualmente (DMV)

Son los DaV enriquecidos con la información necesaria para su representación en pantalla. Los datos en este estado son directamente visualizables aplicando una técnica que los soporte. En este estado se hace presente un sustrato espacial que directamente pudo haber estado ausente en el estado anterior.

$$\text{DMV} = \text{DaV} + \textit{Estructura Visual}$$

La estructura visual constituye la información de soporte necesaria para la representación visual de los datos. Este estado representa formalmente cómo el usuario desea visualizar los datos y provee todos los elementos necesarios para que el mostrado se haga de la manera por él seleccionada.

$$\textit{Estructura Visual} = \textit{Sustrato Espacial} + \textit{Sustrato Gráfico}$$

$$\textit{Sustrato Gráfico} = \textit{Elementos Visuales} + \textit{Atributos Gráficos de los Elementos Visuales}$$

SUSTRATO ESPACIAL

Los atributos del sustrato espacial reflejan la organización del espacio que utilizará la técnica de visualización. Esta información representa una decisión clave de diseño, pues la ubicación espacial de los elementos es perceptualmente dominante. En este estado se debe tener información sobre las características generales de la representación buscada, y como resultado de la transformación anterior, se debe tener información sobre las características geométricas y topológicas de la representación a generar. La geometría propiamente dicha se instanciará en la próxima transformación por la aplicación puntual de una técnica que soporte las restricciones o características expresadas por el usuario.

La información presente en este estado debe ser suficiente para poder determinar la organización espacial de la vista, que constituye una de las primeras decisiones de diseño de la visualización y establece cómo se organizará la presentación en la pantalla. El espacio vacío en sí mismo, como contenedor, tiene su propia métrica, la cual puede ser descrita en términos de los ejes y sus propiedades.

Según Card *et al.* [CMS99], los ejes pueden ser lineales o radiales, y se pueden clasificar en cuatro tipos, según sus propias características y las de los datos que representan:

- Ejes no estructurados: La organización del espacio no presenta ejes. La figura 2.10(a) es un ejemplo de una visualización de un espacio con ejes ausentes en el cual se visualiza un conjunto de datos correspondiente a un sistema de 127729 archivos, 10814 carpetas y 13 niveles de profundidad con la técnica *Girolayout* [UCM13b].
- Ejes nominales o categóricos: Permiten dividir una región en subregiones al solo efecto de categorizar el contenido del espacio. En la figura 2.10(b) se muestran datos sobre quiénes sobrevivieron o no al hundimiento del *Titanic* tanto divididos por sexo como por tipo de pasajero (tripulación y cada una de las tres clases) [BKH05]. En el ejemplo se utiliza la técnica de coordenadas paralelas.
- Ejes ordinales: Permiten dividir una región del espacio en subregiones. Cada región del espacio tiene una posición respecto de las restantes, respetando una relación de orden. El orden relativo de las subregiones es significativo. En la figura 2.10(c) se muestra una visualización del historial delictivo de una persona utilizando *LifeLines* [PMS⁺98], donde cada eje tiene un ordenamiento temporal.
- Ejes cuantitativos: Representan regiones que tienen una métrica. Usualmente están asociados con atributos de este tipo. En la figura 2.10(d) se muestra una visualización de una función matemática con ejes cuantitativos.

Una de las limitaciones que deben manejarse en visualización es la que impone el *display* para representaciones de las vistas. Independientemente de la dimensionalidad de los datos a visualizar, la pantalla nos ofrece la posibilidad de lograr repre-

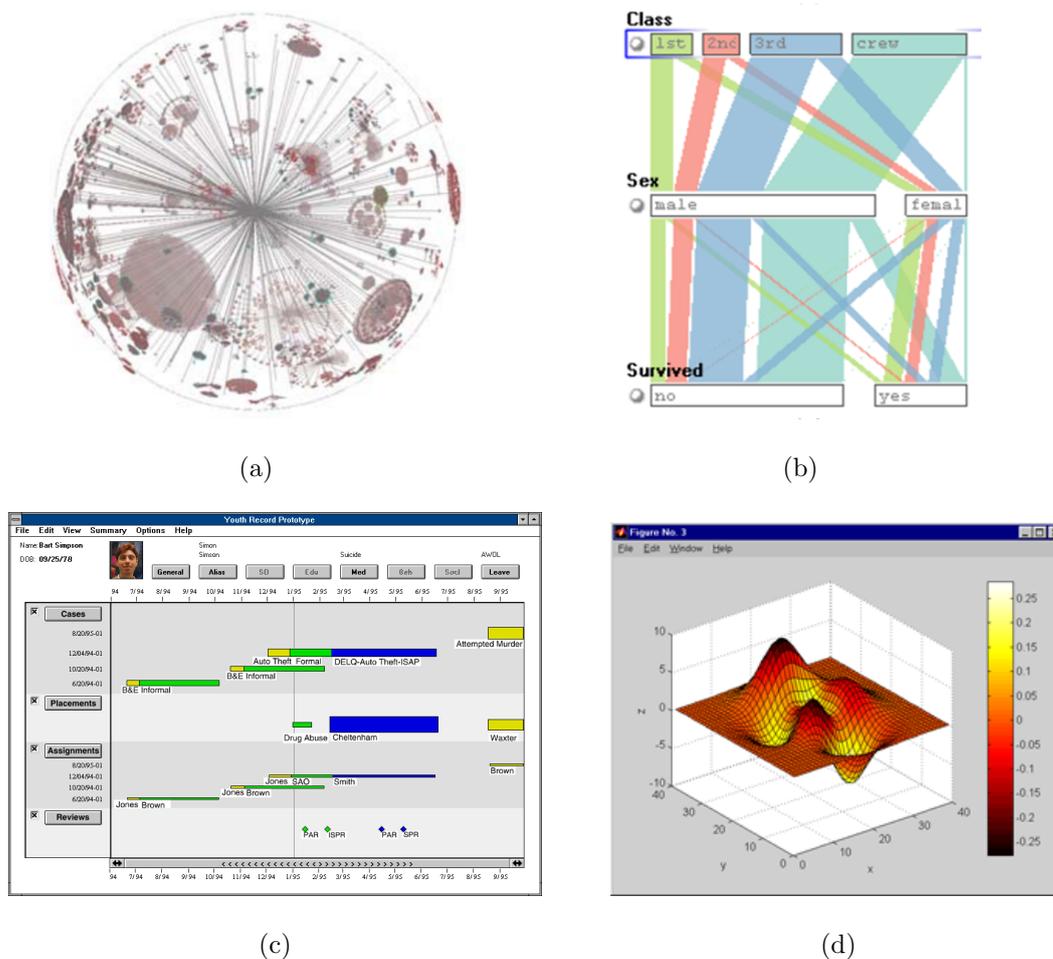


Figura 2.10: (a) Ejemplo de ejes no estructurados: visualización de un sistema de archivos con Girolayout [UCM13b]. (b) Ejemplo de ejes nominales: visualización de la tripulación y los pasajeros que sobrevivieron o no al hundimiento del *Titanic*, tanto divididos por sexo como por tipo de pasajeros (tripulación y cada una de las tres clases) [BKH05]. (c) Ejemplo de ejes ordinales: en LifeLines [PMS⁺98] cada eje tiene un ordenamiento temporal. (d) Ejemplo de ejes cuantitativos: en este caso en particular, los ejes representan coordenadas físicas.

representaciones visuales en una, dos o tres dimensiones espaciales. Para aumentar las posibilidades de codificación espacial Card *et al.* [CMS99] proponen las siguientes técnicas:

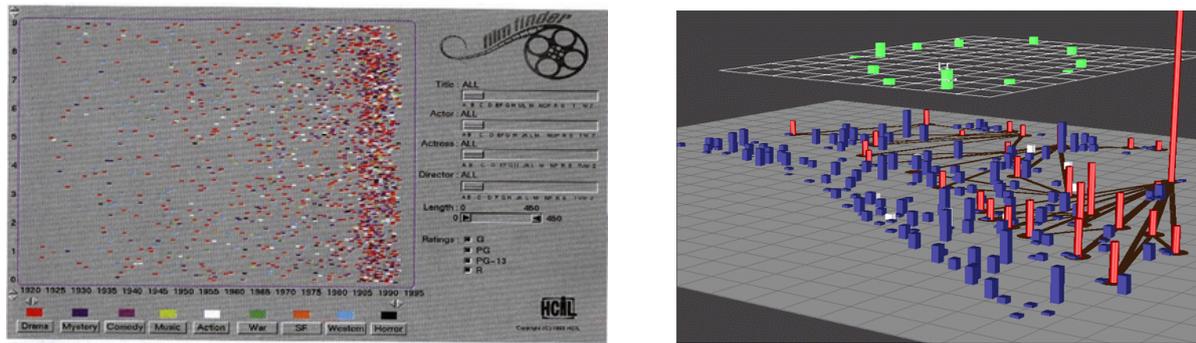
- **Composición:** Ubicación ortogonal de los ejes, creándose un espacio métrico entre cada par de ejes. En la figura 2.11(a) se muestra una ubicación ortogonal de los ejes en una visualización de un conjunto de datos de películas utilizando

la herramienta FilmFinder [AS94b].

- Alineación: Repetición de un eje en distintas posiciones en el espacio. En la figura 2.11(b) se muestran dos planos, uno de los cuales (inferior) muestra todos los datos y el otro (superior) muestra solo determinados elementos de interés [CRMK95].
- Plegado: Continuación de un eje en una dimensión ortogonal. En la figura 2.11(c) se muestra una visualización de los personajes de *The Jungle Book* [KKDF97] mediante el sistema *SeeSoft*. Cada capítulo se representa en un eje donde cada línea horizontal corresponde a un personaje y éstos se muestran en orden de aparición [ESS92].
- Recursión: Subdivisión repetida del espacio. Se logra plegando y alineando los ejes repetidamente. En la figura 2.11(d), se muestra un árbol utilizando la técnica de *Tree-map*¹⁸ [JS91], en la cual el espacio se particiona en rectángulos para representar a los nodos del árbol. Recursivamente, el espacio destinado a cada nodo se particiona en nuevos rectángulos que representarán a los hijos del nodo.
- Sobrecarga: Reuso del mismo espacio, por el mismo conjunto de datos. En la figura 2.11(e) se muestra una visualización utilizando superficies 3D, aplicando la técnica *Mundo dentro de Mundos* [FB90]. En esta técnica, el significado de un sistema de coordenadas está determinado por su disposición dentro de otro. Esta técnica se basa fundamentalmente en que los datos ocupan sólo una porción del espacio comprometido, permitiendo que el espacio sea reutilizado para un segundo uso. Debido a que esta sobrecarga es controlada dinámicamente por el usuario, éste debe estar dispuesto a aceptar casos de oclusión.

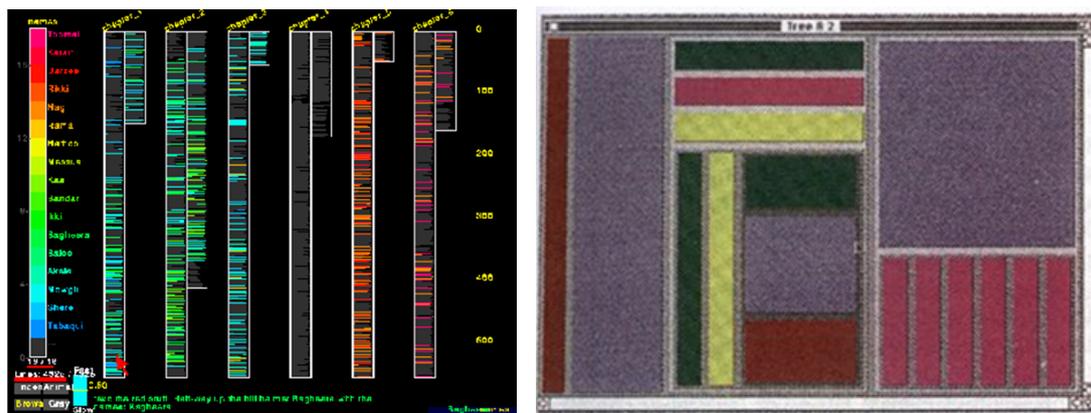
En 2014, Munzner [Mun14] también aborda la organización de los datos en pantalla presentando distintas alternativas en función del tipo de los datos a representar. En la figura 2.12 se especifican las distintas alternativas de organización presentadas por Munzner:

¹⁸ *Tree-map*: Técnica de Visualización de datos jerárquicos que utiliza rectángulos anidados para representar los nodos de un árbol. Cada rectángulo tiene un área proporcional a la cantidad de datos que representa.



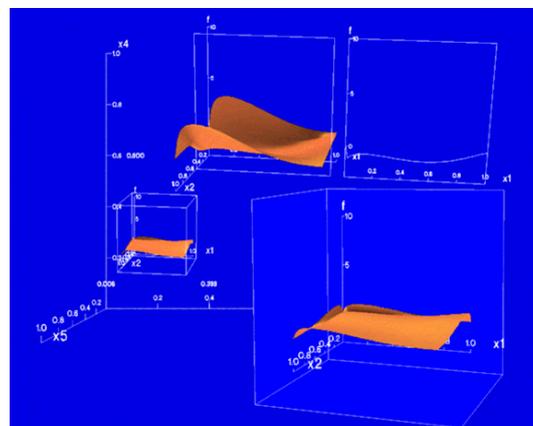
(a)

(b)



(c)

(d)



(e)

Figura 2.11: Técnicas de Aumento de Codificación Espacial. Ejemplos de: (a) composición de ejes [AS94b], (b) alineación de ejes [CRMK95], (c) plegado de ejes [ESS92], (d) recursión [JS91] y (e) sobrecarga espacial [FB90].

- Valores Cuantitativos: Cuando los datos a representar son cuantitativos una alternativa consiste en utilizar la posición espacial para codificar visualmente los datos. El atributo cuantitativo es mapeado a una posición espacial en un

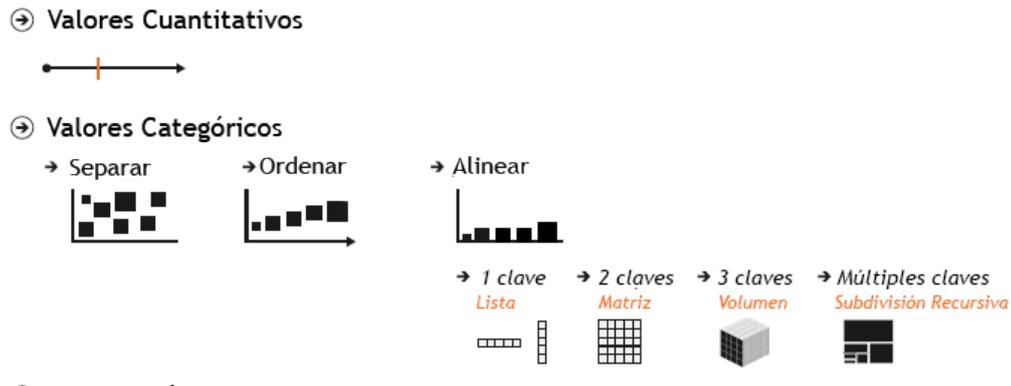


Figura 2.12: Alternativas de Organización de los Datos de Munzner [Mun14].

eje. En el caso más simple de codificación de un único atributo simplemente se codifica cada ítem de dato con una marca en una posición en un eje.

En la figura 2.13(a) se muestra un ejemplo de la utilización del espacio para codificar valores cuantitativos mediante la utilización de un *scatterplot*. En un *scatterplot* se codifican dos variables cuantitativas a los ejes vertical y horizontal. En esta figura, cada punto representa un país, donde el eje horizontal codifica la expectativa de vida y el eje vertical codifica el índice de mortalidad infantil. El canal de color es utilizado para representar el continente de cada país (atributo categórico) y el tamaño de la marca para representar la población [RFF⁺08].

- Valores Categóricos (Separar, Ordenar y Alinear): El uso del espacio para codificar atributos categóricos es más complejo que para codificar atributos cuantitativos, y su valor puede ser expresado como una posición espacial. Los atributos categóricos, sin embargo, se pueden codificar satisfactoriamente como regiones en el espacio. Al agrupar todos los ítems que tengan el mismo valor dentro de la misma región se utiliza la proximidad espacial para codificar su similitud. La distribución en regiones de ítems en función de un atributo categórico se desglosa en tres operaciones: separar en regiones, alinear las regiones y ordenar las regiones. La separación en regiones y el ordenamiento de las mismas debe llevarse a cabo siempre, pero la alineación es opcional. La separación siempre debe hacerse de acuerdo a un atributo categórico, mientras que la alineación y el ordenamiento deben hacerse de acuerdo a un atributo or-

dinal. Dependiendo de la cantidad de claves¹⁹ del conjunto de datos utilizado, la alineación puede ser en forma de:

- Lista (una clave): Al separar en regiones utilizando una sola clave se obtiene una región por ítem de dato. Estas regiones se organizan frecuentemente en forma de lista.

En la figura 2.13(b) se muestra un gráfico de barras, un ejemplo simple de alineación de regiones en forma de lista. En este ejemplo en particular se está visualizando un conjunto de datos de animales. La especie de los animales corresponde a un atributo categórico codificado, en este caso, en el eje horizontal. A cada especie animal le corresponde una región, y se utiliza una marca de área para representar el peso promedio de la especie (atributo cuantitativo). En este caso en particular, las regiones correspondientes a la especie animal están ordenadas por orden alfabético de acuerdo a su nombre.

- Matriz (dos claves): Aquellos conjuntos de datos que tienen dos claves suelen organizarse en matrices de dos dimensiones, donde una clave se distribuye a lo largo de las filas y la otra clave se distribuye a lo largo de las columnas. Uno de los usos más simples de alineación en forma de matriz es el *Heatmap*²⁰; en éste, cada celda codifica valores cuantitativos mediante el uso de color. La figura 2.13(c) permite visualizar un conjunto de datos bio-informáticos representados con un *Heatmap*. Las claves son genes y condiciones experimentales y el valor cuantitativo representado es el nivel de actividad de un gen en particular en una dada condición experimental. En este *Heatmap* en particular se aplica una escala de color del rojo al verde, utilizada comúnmente en el dominio de la Genómica²¹.
- Grillas Volumétricas (tres claves): De la misma forma que los datos pueden alinearse en listas de una dimensión o matrices de dos dimensiones, es posible también alinear los datos en una grilla volumétrica de tres dimen-

¹⁹Clave: dimensión que permite identificar unívocamente cada ítem del conjunto de datos.

²⁰*Heatmap*: Es una representación bidimensional de datos en la que los valores están representados por colores.

²¹Genómica: Conjunto de disciplinas relacionadas con el estudio de los genomas y su aplicaciones en terapia génica, biotecnología, etc.

siones. Sin embargo, esta decisión de diseño típicamente no se recomienda para datos no espaciales ya que introduce problemas perceptuales (como oclusión, por ejemplo).

- Subdivisión recursiva (múltiples claves): Cuando los datos tienen múltiples claves es posible extender las propuestas anteriores subdividiendo recursivamente los elementos de una matriz o una lista.

Con respecto a la orientación espacial de los ejes, Munzner plantea tres posibilidades:

- **Orientación Rectilínea:** En una organización rectilínea, las regiones o ítems se distribuyen a través de ejes ortogonales. Todas las visualizaciones de la figura 2.13 utilizan organización rectilínea.
- **Orientación Paralela:** En este tipo de organización, los ejes se ubican paralelos entre sí. Un ítem se representa como una poligonal que pasa por todos los ejes una única vez atravesándolos exactamente en el punto correspondiente al valor del ítem en la dimensión asociada al eje. En la figura 2.10(b) se muestra un ejemplo de una visualización que utiliza la técnica de coordenadas paralelas con ejes nominales.
- **Orientación Radial:** En una organización radial los ítems se distribuyen alrededor de un círculo. El sistema de coordenadas natural para este tipo de organización es el sistema de coordenadas polar. Uno de los gráficos estadísticos radiales más utilizados es el diagrama de torta.

SUSTRATO GRÁFICO

Los atributos del sustrato gráfico representan los elementos (y sus propiedades gráficas asociadas) que se utilizarán para representar los datos en la vista a generar.

La información presente en este estado se puede descomponer en:

- **Elementos Visuales (Marcas)**

Un elemento visual (o marca) es un elemento gráfico en una imagen. Se trata de primitivas geométricas que pueden clasificarse según el número de dimensiones que requieren. Se debe tener información que represente las decisiones tomadas en la transformación anterior sobre los elementos visuales que se utilizarán para

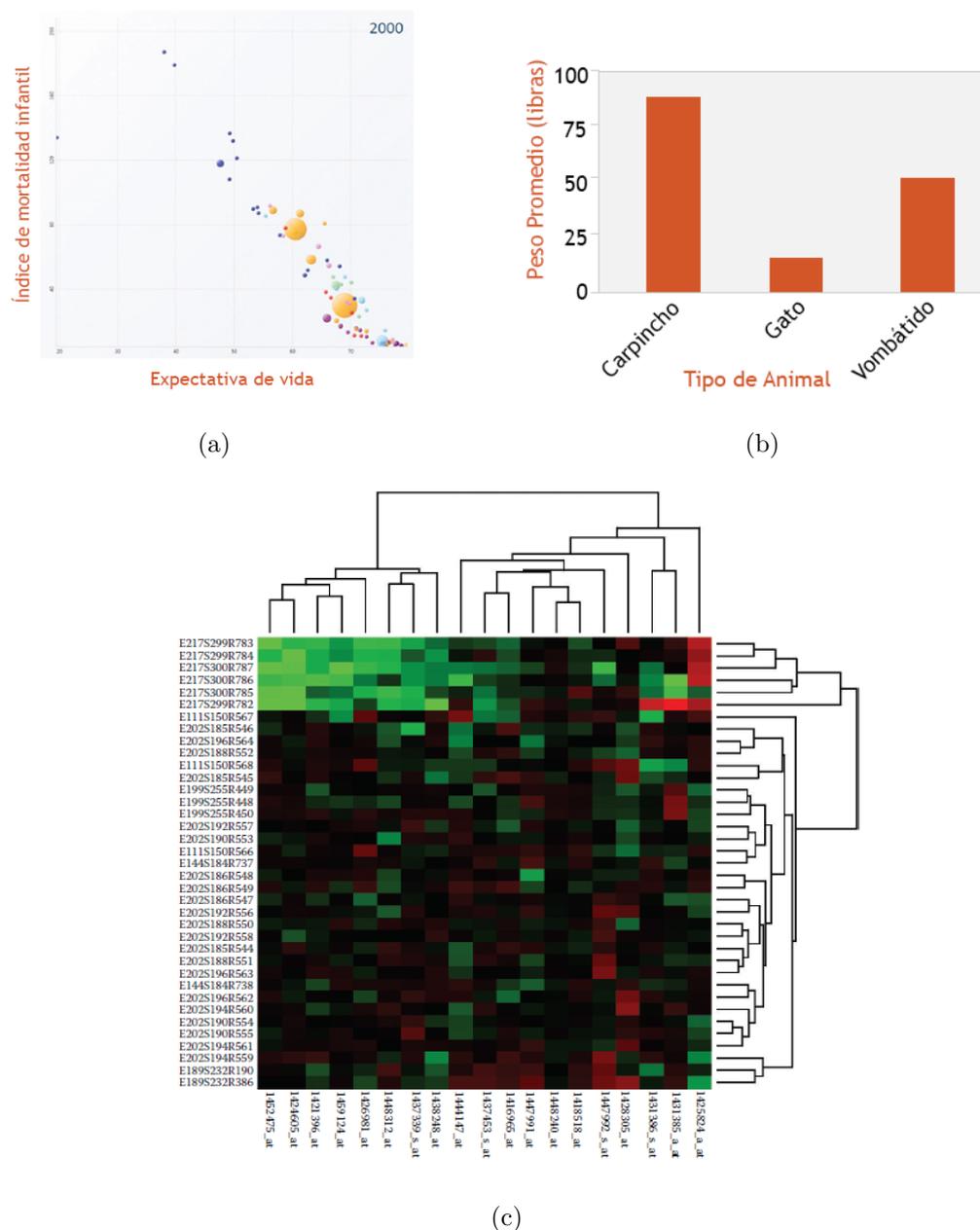


Figura 2.13: Alternativas de Organización Espacial. (a) *Scatterplot*: ejemplo de la utilización del espacio para codificar valores cuantitativos [RFF⁺08]. (b) Gráfico de barras: ejemplo de la separación en regiones para codificar valores categóricos alineando las regiones en una lista [Mun14]. (c) *Heatmap*: ejemplo de alineación en forma de matriz [Mun14].

representar los datos y atributos de interés para el usuario. En realidad, las operaciones del usuario están limitadas al rango de elementos visuales que se pueden mostrar en la pantalla.

En 1967, Bertin [Jac67, Ber83] publicó el libro *Semiology of graphics: diagrams, networks, maps* reconocido como un trabajo fundacional en los campos de diseño y cartografía. En éste sintetiza los principios de comunicación gráfica, tarea nunca realizada con anterioridad. En ese contexto define tres tipos de marcas: los puntos, las líneas y las zonas.

Card *et al.* [CMS99] establecen cuatro tipos elementales de marcas: puntos (0D), líneas (1D), áreas (2D) y volúmenes (3D).

Marcas tales como los puntos y las líneas pueden utilizarse para representar estructuras topológicas como grafos y árboles [CMS99]. En este contexto, Munzner [Mun14] clasifica las marcas en dos tipos: aquéllas que representan un ítem y aquéllas que representan un enlace. Entre las marcas de enlace identifica dos tipos: las marcas de conexión y las marcas de contenido. Una marca de conexión muestra la relación entre dos ítems utilizando, por ejemplo, una línea. Una marca de contenido representa relaciones jerárquicas utilizando áreas anidadas que representan múltiples niveles. En la figura 2.14 se ilustran los distintos tipos de marcas presentados por Munzner.

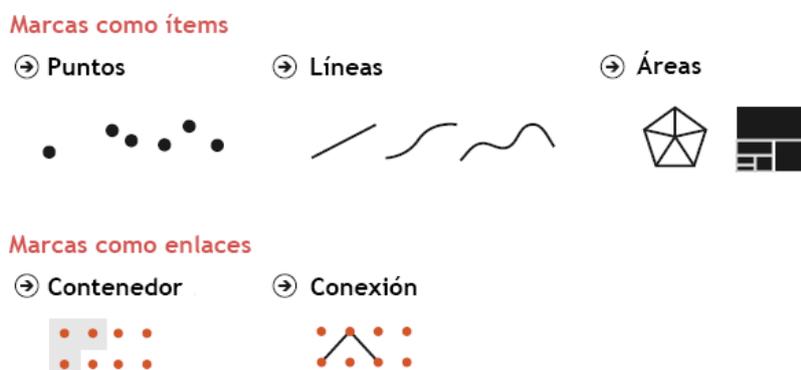


Figura 2.14: Clasificación de las Marcas Visuales de Munzner [Mun14].

- Atributos de los Elementos Visuales (Canales)

Son las propiedades de los elementos visuales que se pueden utilizar para codificar información adicional. Se trata de los atributos que controlan la apariencia de las marcas visuales, independientemente de la dimensionalidad de la geometría.

La retina del ojo es sensitiva a algunas propiedades gráficas independientemente de su posición. A estas propiedades, Bertin [Ber83] las llamó propiedades

retinianas. En este contexto, Bertin definió siete variables visuales, una de posición y seis retinianas. En la figura 2.15 se ilustran las 7 variables visuales de Bertin clasificadas en función de su capacidad para expresar alcance (posición, tamaño y escala de grises) o su capacidad de expresar diferencias (textura, color, orientación y forma).

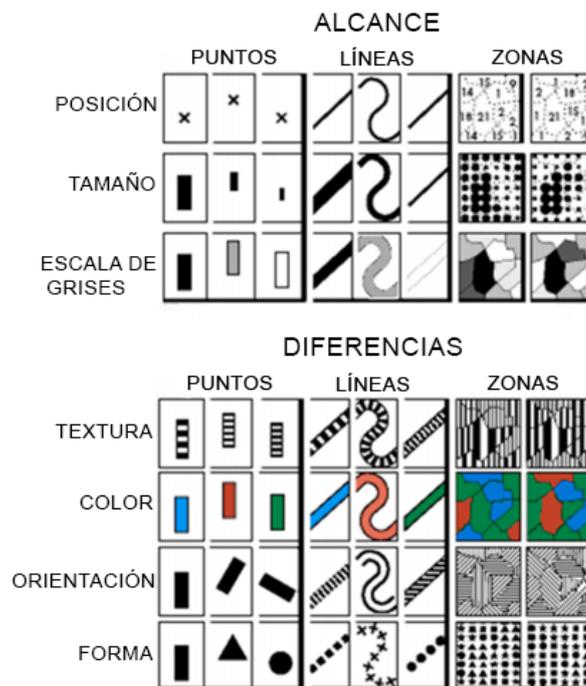


Figura 2.15: Variables Visuales de Bertin [Ber83].

Otras propiedades gráficas para codificar información han sido propuestas. Ward *et al.* [WY04] agrega el movimiento a las variables visuales de Bertin; éste puede asociarse a cualquiera de las otras siete variables ya que la forma en que las variables varían en función del tiempo también puede transmitir información relevante.

La figura 2.16 muestra algunos atributos visuales presentados por Munzner [Mun14] que pueden ser utilizados para codificar información y propiedades de las marcas. Algunos canales corresponden a la posición espacial. Otros corresponden al color, que tiene tres aspectos distintos: cromaticidad, saturación y luminancia (ya identificada inicialmente por Bertin [Ber83]). Hay tres canales de tamaño, uno para cada dimensión: longitud (1D), área (2D) y volumen (3D). Los canales orientados al movimiento incluyen el patrón, la dirección de movimiento, y la velocidad. Otros canales visuales son la orientación, la

curvatura y la forma.

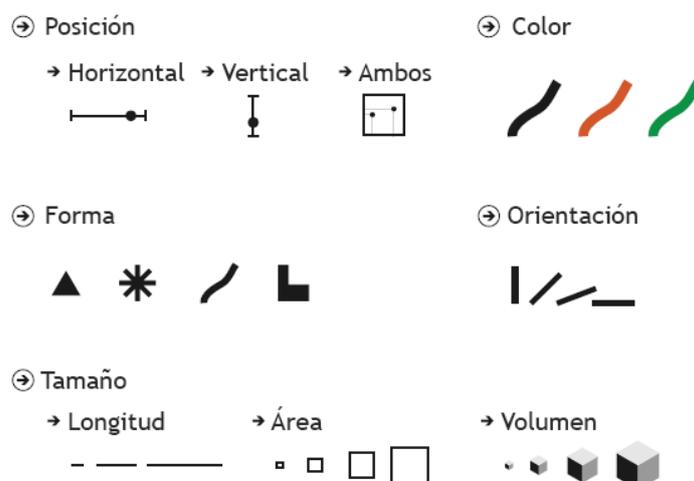


Figura 2.16: Canales Visuales de Munzner [Mun14].

Munzner [Mun14] clasifica los canales visuales en *canales de identidad* y *canales de magnitud*. Los canales de identidad dan información sobre qué es algo o dónde está (forma, cromaticidad, patrón de movimiento, etc.). Los canales de magnitud dan información sobre cantidad (longitud, tamaño, luminancia).

Hay dos principios fundamentales que guían el uso de canales visuales en la codificación visual: la expresividad y la efectividad. El principio de expresividad sostiene que la codificación visual debe expresar todo acerca de la información presente en los atributos del conjunto de datos pero no más de lo que está presente. Por ejemplo, si determinados datos están ordenados deben mostrarse de manera que nuestro sistema perceptual los perciba intrínsecamente como ordenados. Por el contrario, los datos no ordenados deberían mostrarse de forma tal que no induzcan a percibir un ordenamiento que es inexistente. El principio de la efectividad dicta que la importancia del atributo debe corresponderse con la saliencia²² del canal; es decir, que los atributos más importantes deben codificarse con los canales más efectivos. Del mismo modo, los atributos de menor importancia pueden ser mapeados a canales menos efectivos.

Si bien existen diversas clasificaciones de efectividad para los canales visuales, éstas no difieren en gran medida entre sí. En la figura 2.17 se presenta la clasificación de efectividad de Munzner [Mun14] para los canales visuales que

²²Saliencia: Capacidad de sobresalir del estímulo. Capacidad para “llamar la atención” del sujeto.

considera los canales de identidad y de magnitud para atributos categóricos y ordinales.

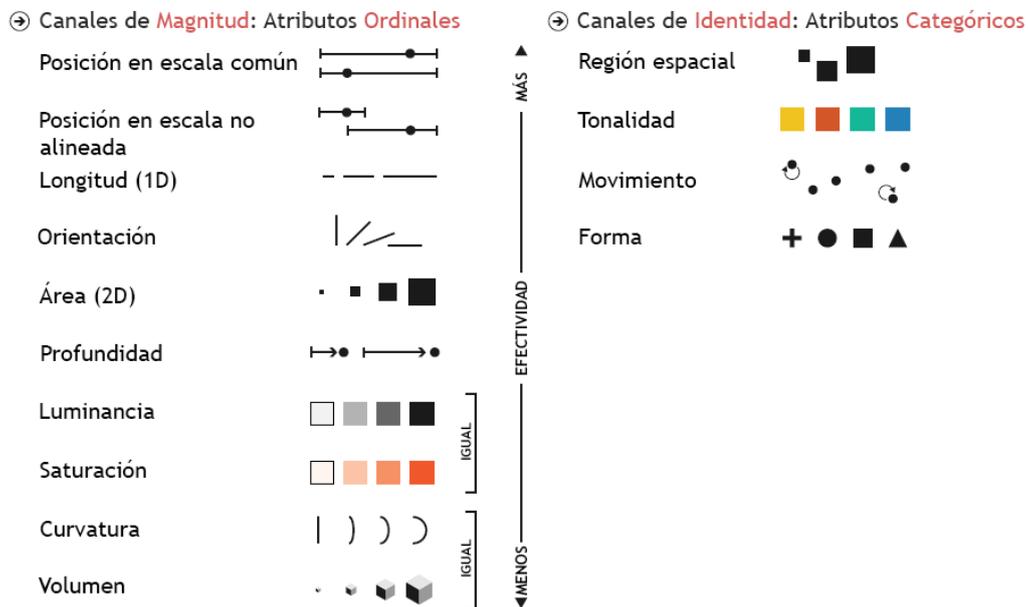


Figura 2.17: Clasificación de Efectividad de los Canales Visuales para atributos categóricos y ordinales de Munzner [Mun14].

Al analizar la efectividad de un canal visual se tienen en cuenta diversos criterios, tales como precisión, discriminabilidad, separabilidad, la habilidad para proveer *popout*²³ visual y la capacidad de proveer agrupamiento perceptual.

En este punto resulta evidente que para un determinado conjunto de Datos a Visualizar (DaV) pueden existir distintos conjuntos de Datos Mapeados Visualmente (DMV). Esto es relevante en el marco de un proceso de exploración donde el usuario pueda o necesite comparar y analizar distintas maneras de visualizar un mismo conjunto de datos.

2.3.1.5. Datos Visualizados (DV)

Es el último estado del proceso, pero no el final, ya que constituye el espacio de exploración para el usuario.

Es importante recalcar que para un mismo conjunto de DMV se puede generar más de una vista por la aplicación de las distintas técnicas que soportan las características o las restricciones especificadas en el estado anterior.

²³*Popout*: Capacidad de sobresalir.

2.3.2. Las Transformaciones

Son los procesos que permiten pasar de un estado al siguiente. El usuario puede interactuar a nivel de las transformaciones, así como a nivel de los estados de los datos.

Transformación de Datos (DC-DA)

Esta transformación genera un conjunto de *DA* a partir de las fuentes de datos externas o provenientes de otras visualizaciones. Se puede generar un nuevo conjunto de *DA* o directamente incorporarlos a un conjunto existente. Esta transformación permite capturar los *DC* provenientes del dominio de aplicación y llevarlos a un formato interno manejable para el resto del proceso. Como resultado de esta transformación se generan *datasets* como los descritos en la sección 4.5 del capítulo 4.

Transformación de Filtrado (DA-DaV)

Esta transformación permite al usuario seleccionar o filtrar los datos que desea visualizar y realizar tareas como proyección y filtrado. Se generan nuevos conjuntos de *DaV* manteniendo para todos el mismo conjunto de *DA*. El objetivo de esta transformación es permitir que el usuario defina qué es lo que quiere visualizar en una determinada instancia, sin preocuparse todavía de cómo hacerlo. En este punto el usuario decide qué tuplas o ítems quiere visualizar y cuáles de todos los atributos presentes en los *DA* participarán en la vista a generar. Esta participación puede ser su mostrado inicial o su disponibilidad ante un requerimiento por parte del usuario en una etapa posterior.

Transformación de Mapeo Visual (DaV-DMV)

Esta transformación (netamente interactiva) es clave en el proceso de visualización y le permite al usuario definir cómo quiere visualizar sus datos. Es la transformación encargada de realizar el mapeo visual, es decir, es aquí donde se establecen:

- Cuáles son las estructuras visuales adecuadas.
- Qué atributos se mapearán espacialmente y cómo.
- Qué elementos visuales (marcas) se utilizarán y con qué atributos gráficos (canales).

El objetivo final de esta transformación es lograr un mapeo expresivo y efectivo. Un mapeo es expresivo cuando son representados todos los datos pertenecientes al

conjunto de DaV pero ningún otro. Por otro lado, la efectividad del mapeo estará dada por la manera en que la representación visual sea percibida por el usuario. Esta transformación no es trivial y una gran parte de su éxito depende del conocimiento de las características perceptuales del humano que se le imprima a la misma.

Transformación de Visualización (DMV-DV)

Es la transformación encargada de generar la representación visual en pantalla según lo expresado en los DMV. Una vez seleccionada una técnica de visualización que soporte las restricciones presentes en el estado anterior, esta transformación es la que genera la vista. Probablemente se necesite llevar los datos a una estructura adecuada que permita una manipulación eficiente de los mismos. La elección de la estructura de soporte estará determinada por la técnica en particular considerando no solamente las facilidades que deberá proveer para su renderizado, sino también el soporte de las interacciones que se ofrecerán sobre la vista que ésta genera. Para un determinado conjunto de DMV pueden existir varias técnicas que lo soporten, debiendo el usuario optar por alguna de ellas para obtener de esta manera los DV.

En la figura 2.18 se ilustra el Modelo Unificado de Visualización detallando sus estados y transformaciones.



Figura 2.18: Estados y Transformaciones en el Modelo Unificado de Visualización.

2.4. Múltiples Vistas Coordinadas en el MUV

La técnica de múltiples vistas coordinadas se ha convertido en una tecnología bien establecida para el análisis visual interactivo [Rob07]. Se trata de una técnica de visualización exploratoria que permite a los usuarios explorar sus datos a través de diferentes representaciones. Para el análisis de datos complejos se requiere usualmente que el usuario considere escenarios distintos, surgiendo la necesidad de comparar múltiples visualizaciones generadas a partir del mismo conjunto de datos.

El Modelo Unificado de Visualización fue diseñado para permitir el manejo de múltiples vistas coordinadas. Como se mencionó anteriormente, a partir de un conjunto de DC se pueden obtener múltiples conjuntos de DA, ya sea que utilicen una representación distinta para los mismos datos o que representen un subconjunto diferente de los datos presentes en el conjunto de DC. Del mismo modo, a partir de un conjunto de DA se pueden obtener múltiples conjuntos de DaV, resultado de diferentes procesos de filtrado aplicados sobre los DA. Además, como resultado de la aplicación de diferentes mapeos visuales a partir de un único conjunto de DaV, pueden obtenerse múltiples conjuntos de DMV. Finalmente, dado que para un determinado conjunto de DMV pueden existir varias técnicas que lo soporten, es posible generar múltiples conjuntos de DV. En todos los casos, la generación de múltiples conjuntos de datos a partir de un único conjunto de datos fuente es el resultado de aplicar nuevas transformaciones sobre el estado correspondiente al conjunto de datos fuente en el proceso de visualización. Cada nueva transformación generará una ramificación en el proceso de visualización. El proceso de visualización tendrá la estructura de un árbol, cuya raíz será el estado DC y podrá ramificarse en cualquiera de sus transformaciones (ver figura 2.19).

Todas las ramificaciones generadas a partir de la Transformación de Visualización, generarán visualizaciones distintas del mismo conjunto de DMV, pero aplicando una técnica de visualización diferente. Todas las ramificaciones generadas a partir de la Transformación de Mapeo Visual, generarán visualizaciones del mismo conjunto de DaV, pero generadas a partir de un mapeo visual diferente y tal vez, una técnica de visualización diferente. Todas estas vistas, a pesar de ser diferentes, mostrarán el mismo conjunto de datos.

Todas las ramificaciones generadas a partir de la Transformación de Filtrado, generarán visualizaciones del mismo conjunto de DA, pero generadas a partir de un proceso

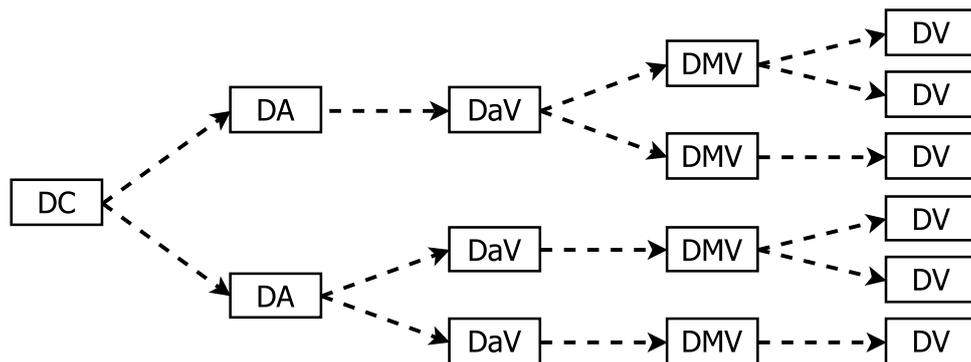


Figura 2.19: Ramificaciones en el MUV: Múltiples Vistas.

de filtrado diferente, y tal vez un mapeo visual y una técnica de visualización diferente. Estas vistas no mostrarán exactamente los mismos datos, ya que no comparten el mismo conjunto de DaV. Finalmente, todas las ramificaciones generadas a partir de la Transformación de Datos, generarán visualizaciones del mismo conjunto de DC, pero a partir de una transformación de datos diferente, y tal vez de un proceso de filtrado, un mapeo visual y una técnica de visualización diferente. Estas vistas no mostrarán necesariamente los mismos datos, ya que únicamente comparten el conjunto de DC.

En el Modelo Unificado de Visualización, todas las vistas compartirán (al menos) el conjunto de DC (ver figura 2.19); por ende, generarán representaciones visuales distintas del mismo conjunto de datos. Dado que en la Transformación de Datos los datos son transformados a un formato interno manejable para el resto del proceso, es posible asumir que a partir del estado DA todas las vistas generadas en un *pipeline* de visualización podrán estar correlacionadas. Claramente, aquellas vistas que compartan los DaV mostrarán exactamente los mismos datos, pero no necesariamente representados de la misma forma. Esto no ocurre con aquellas vistas que comparten únicamente los DA, ya que es posible que, debido a los procesos de filtrado o agregación involucrados en la Transformación de Filtrado, los datos mostrados en una vista no estén presentes en las demás. Esto último puede traer acarreados problemas de consistencia, ya que al vincular dos vistas que no muestran exactamente los mismos datos, el efecto sobre una vista puede no verse reflejado en la otra si los datos involucrados no están presentes en la segunda. Sin embargo, teniendo presente que puede ser necesario un trato especial en aquellas vistas coordinadas que no compartan los DaV, es posible coordinar todas las vistas generadas a partir de distintas ramificaciones de un *pipeline* en el MUV que compartan el estado DA.

2.5. Conclusión

En este capítulo se ha definido el concepto de visualización y se ha profundizado en el estado del arte del proceso de visualización. Además, se ha presentado el Modelo Unificado de Visualización (MUV), un marco de referencia que es independiente de la naturaleza de los datos del campo de aplicación. Este modelo se describió en base a sus estados y transformaciones. En la descripción de cada uno de los estados y transformaciones del MUV se describió el estado del arte de las clasificaciones de los datos, de los conjuntos de datos y del mapeo visual. Asimismo, se abordó una discusión sobre cómo el Modelo Unificado de Visualización soporta la técnica de múltiples vistas correlacionadas.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 3

Interacciones en Visualización

“A graphic is never an end in itself: it is a moment in the process of decision making”

— Jacques Bertin, 1981

La visualización es un proceso cognitivo mediante el cual, iterativa e interactivamente, el usuario explora el espacio de sus datos. Las características del proceso de visualización determinan que el usuario deba poder interactuar con los datos y sus representaciones intermedias, controlar las transformaciones y manipular las visualizaciones, para poder explorar así el espacio de información. Como resultado de esa exploración obtendrá, entre otras alternativas, la confirmación de hipótesis sobre los datos analizados, el descubrimiento de patrones, determinadas características presentes en los mismos, etc. Es claro que las interacciones constituyen una herramienta esencial en este proceso de descubrimiento. En este contexto, una taxonomía de las interacciones es sumamente útil para lograr un mejor entendimiento del espacio de diseño de las interacciones en el área de visualización.

En la literatura existen muchos trabajos que tienen por objetivo clasificar o definir taxonomías de interacciones en el contexto de visualización. Sin embargo, las taxonomías definidas presentan niveles de granularidad significativamente distintos y, en algunos casos, las características de las mismas en relación con su funcionalidad, los objetos de la interacción y sus alcances son parcialmente disjuntas.

En este capítulo proveemos una definición del concepto de interacciones en el contexto de visualización y nos enfocamos en el trabajo previo para obtener, a partir de éste, una primera clasificación de las distintas alternativas existentes que constituyera un marco

conceptual de las interacciones en visualización. Se realizó un relevamiento exhaustivo y un análisis profundo de las distintas clasificaciones de las interacciones y se ordenó el trabajo previo en cuatro clases:

1. Taxonomías de las Técnicas de Interacción.

En esta sección se detallan las clasificaciones y taxonomías enfocadas en las técnicas de interacción. Este tipo de taxonomías se utilizan para categorizar técnicas de interacción.

2. Taxonomías de las Dimensiones de las Técnicas de Interacción.

En esta sección se describen trabajos relevantes que pretenden proveer dimensiones para describir las técnicas de interacción.

3. Taxonomías Basadas en Tareas.

En esta sección se presentan aquellos trabajos que definen taxonomías o espacios de diseño orientados a las tareas del usuario.

4. Interacciones en *Visual Analytics*¹.

En esta sección se presentan las taxonomías y clasificaciones de las interacciones extendidas para cubrir el área de *Visual Analytics*.

3.1. Definición

Encontrar una definición para el término “Interacción” es considerado un reto [YaKSJ07]. En el contexto de HCI, Dix describe las interacciones como: “*La comunicación entre el usuario y el sistema*” [Dix09]. En el contexto del análisis de datos, Becker *et al.* [BCW87] definieron el término “Interacción” como la “*manipulación directa y el cambio instantáneo*”.

No obstante, el concepto de “técnica de interacción” es más fácil de definir y más tangible que el de “interacción” en sí mismo. En el contexto de HCI Foley *et al.* [F⁺90] definen el concepto de técnica de interacción como “*una forma de utilizar un dispositivo físico de entrada/salida para realizar una tarea genérica en un diálogo humano-computadora*”.

Las técnicas de interacción en el contexto de visualización pueden pensarse como las técnicas que proveen al usuario la posibilidad de manipular directa o indirectamente

¹ *Visual Analytics* es la ciencia del razonamiento analítico facilitado por el uso de interfaces visuales interactivas (Thomas y Cook [TC05]).

las representaciones visuales para poder interpretarlas. De acuerdo a esta definición, una imagen estática o una representación animada no tienen asociadas técnicas de interacción. Sin embargo, un menú en una interfaz que permite cambiar la representación de los datos en la vista de un *scatterplot*² a una representación con coordenadas paralelas³, por ejemplo, sería una técnica de interacción ya que permite al usuario manipular la representación de los datos.

3.2. Taxonomías de las Interacciones

La definición de taxonomías de las interacciones en el contexto de visualización es una tarea muy útil para lograr un mejor entendimiento del espacio de diseño de las interacciones. Hasta el momento, hay una gran cantidad de trabajos que proponen taxonomías que difieren en diferentes aspectos. Estas divergencias sugieren que la definición de una taxonomía abarcativa es, sin duda, un reto.

Durante el desarrollo de esta tesis se realizó un relevamiento exhaustivo de los artículos científicos publicados relacionados con taxonomías, caracterizaciones y clasificaciones de las interacciones en el contexto de visualización y *visual analytics*. Con el objetivo de integrar todo este trabajo previo y definir una taxonomía que constituya un marco conceptual de las interacciones en visualización se analizó y clasificó el estado del arte en cuatro clases diferentes. En lo que respecta a cada clase, se observa distinta granularidad en cada una de las subclases, lo que dificulta su comparación. Por otro lado, también se observa que, si bien las subclases presentan diversas alternativas (a las que denominamos unidades de interacción, de modo general) éstas son además parcialmente disjuntas. Esto implica que no todas son completas.

La tabla 3.1 resume los estudios sobre interacciones en visualización y *visual analytics* que consideramos relevantes. Algunos de estos trabajos categorizan las técnicas de interacción [BCS96, CR96, DE98, Kei02, WY04, Wil06, YaKSJ07, WGK10, LCWL14], otros describen las técnicas de interacción en función de sus dimensiones [Twe97, Spe07] y

²Un *Scatterplot* es un diagrama de dispersión donde los datos son graficados como puntos en una gráfica para mostrar una relación posible entre dos conjuntos de datos.

³Coordenadas Paralelas: técnica de visualización para datos multi-dimensionales. Cada una de las dimensiones de los datos se corresponde a un eje vertical y cada ítem de dato se muestra como una poligonal que conecta una serie de puntos a lo largo de los ejes.

otros se enfocan en las tareas del usuario [ZF98, AES05, SNHS13, BM13]. Finalmente, se describen las taxonomías relevantes definidas en el contexto de *Visual Analytics* [TC05, HS12, KS12].

A continuación detallaremos y analizaremos cada una de las clases planteadas.

Publicación	Descripción
Taxonomías de las Técnicas de Interacción	
Buja <i>et al.</i>	(1996) [BCS96] Foco, conexión y organización de la vista.
Chuah y Roth	(1996) [CR96] Interacciones Básicas de Visualización (BVI): operaciones gráficas (codificar datos, configurar valores gráficos, manipular objetos, etc.), operaciones sobre conjuntos (crear, eliminar, agrupar conjuntos, etc.) y operaciones sobre los datos (incorporar, eliminar, establecer datos, etc.).
Dix y Ellis	(1998) [DE98] Foco y resaltado, acceso a información extra, <i>overview</i> ⁴ y contexto, modificación de parámetros visuales, modificación de la técnica de representación y fusión temporal.
Keim	(2002) [Kei02] Proyecciones dinámicas, distorsión interactiva, <i>zooming</i> ⁵ interactivo, filtrado interactivo, <i>brushing</i> y <i>linking</i> ⁶ interactivo.
Ward y Yang	(2004) [WY04] Operadores de Interacción: navegación, selección y distorsión. Espacios de Interacción (operandos): espacio de la pantalla, espacio de la estructura de los datos, espacio del objeto, espacio de los atributos, espacio del valor de los datos, y espacio de la estructura de la visualización. Parámetros del operador de interacción: foco, alcance, transformación y <i>blender</i> ⁷ .
Wilkinson	(2006) [Wil06] Filtrado, navegación, manipulación, <i>brushing</i> y <i>linking</i> , rotación y animación.
Yi <i>et al.</i>	(2007) [YaKSJ07] Selección, exploración, reconfiguración, codificación, abstracción/elaboración, filtrado y conexión.

Sigue en la página siguiente.

⁴*Overview*: visión general, visión de conjunto.

⁵*Zooming*: dada una representación de los datos a distintas resoluciones, permite pasar de una resolución a otra mayor.

⁶*Brushing* y *Linking*: se aplica a múltiples vistas, permitiendo seleccionar un conjunto de datos en una vista y resaltarlos en todas las demás.

⁷*Blender*: mezclador.

	Publicación	Descripción
Ward <i>et al.</i>	(2010) [WGK10]	Operadores de Interacción: navegación, selección, filtrado, reconfiguración, conexión, elaboración/abstracción, codificación e híbridos. Espacios de Interacción (operandos): espacio de la pantalla, espacio de la estructura de los datos, espacio del objeto, espacio de los atributos, espacio del valor de los datos y espacio de la estructura de la visualización. Parámetros del operador de interacción: foco, alcance, transformación y <i>blender</i> .
Liu <i>et al.</i>	(2014) [LCWL14]	Interacciones WIMP (selección, filtrado, <i>brushing</i> ⁸ y resaltado) e interacciones post-WIMP (utilizadas en las interfaces táctiles).

Taxonomías de las Dimensiones de las Técnicas de Interacción

Tweedie	(1997) [Twe97]	Tipos de Interacción: manual, mecanizada, basada en instrucciones, orientable y automática. Tipos de Interactividad: manipulación directa e indirecta.
Spence	(2007) [Spe07]	Espacios de Información: continuo y discreto. Modos de Interacción: continua, por pasos y pasiva.

Taxonomías de las Tareas

Zhou y Feiner	(1998)[ZF98]	Tareas de Alto Nivel: informar (elaborar y resumir) y permitir (explorar y computar). Tareas de Bajo Nivel: enfatizar, asociar, <i>background</i> , categorizar, agrupar, comparar, correlacionar, distinguir, generalizar, identificar, ubicar, clasificar, sumar y revelar.
Amar <i>et al.</i>	(2005)[AES05]	Tareas de Bajo Nivel: recuperar valor, filtrar, calcular el valor derivado, encontrar extremos, ordenar, determinar el rango, caracterizar la distribución, encontrar anomalías, agrupar y correlacionar.
Schulz <i>et al.</i>	(2013)[SNHS13]	Dimensiones del Espacio de Diseño de las Tareas: objetivo, proceso, características, destino y cardinalidad.

Sigue en la página siguiente.

⁸La técnica de *brushing* selecciona los objetos contenidos en una región establecida.

	Publicación	Descripción
Brehmer y Munz-	(2013)[BM13]	¿Por qué?:
ner	(2014)[Mun14]	- análisis: consumir (descubrir, presentar y disfrutar) y producir (comentar, registrar y derivar). - búsqueda: buscar, localizar, navegar y explorar. - consulta: identificar, comparar y resumir.
		¿Cómo?:
		- codificación: organizar (expresar valores, separar, ordenar, alinear y utilizar) y mapear (color, forma, movimiento, tamaño, ángulo y forma). - manipulación: modificar, seleccionar y navegar. - distribución: juxtaponer, particionar y superponer. - reducción: filtrar, agregar y embeber.
		¿Qué?:
		- entradas y salidas.

Taxonomías de las Interacciones en Visual Analytics

Thomas y Cook	(2005) [TC05]	Niveles de Interacción: constantes de tiempo del humano (constantes del tiempo de procesamiento perceptual, de respuesta inmediata y de una unidad de tarea), uso de las interacciones (modificar la transformación de los datos, modificar el mapeo visual, modificar la transformación de la vista, dialogar) y naturaleza de las interacciones.
Heer y Shneiderman	(2012) [HS12]	Taxonomía de Interacciones en el contexto de VA:
Kerren y Schreiber	(2012) [KS12]	Categoría <i>A</i> (Especificación de los Datos y de la Vista): codificar/visualizar, reconfigurar, filtrar, ordenar, derivar y ajustar. Categoría <i>B</i> (Manipulación de la Vista): seleccionar, navegar/explorar, coordinar/conectar y organizar. Categoría <i>C</i> (Proceso y Precedencia): registrar, comentar, compartir y guiar.

Tabla 3.1: Síntesis de las Taxonomías de Interacciones Relevantes en el Área de Visualización.

3.2.1. Taxonomías de las Técnicas de Interacción

Dentro de los primeros intentos de definir una taxonomía de interacciones se encuentran los trabajos de Buja *et al.* [BCS96] y de Chuah y Roth [CR96].

En 1996, Buja *et al.* [BCS96] proponen una taxonomía rudimentaria basada en tres tareas aplicadas al análisis de datos: la búsqueda de Gestalt, la ejecución de consultas y la realización de comparaciones. Estas tres tareas son apoyadas por tres clases de interacciones de manipulación de vistas: foco, conexión y organización de la vista.

Ese mismo año Chuah y Roth [CR96] incorporaron las Interacciones Básicas de Visualización (BVI) extendiendo el *framework*⁹ presentado por Foley *et al.* [F⁺90] y presentaron una taxonomía básica de clasificación de BVIs que sirvió de base para futuros diseños de espacios de interacción. Las clasificaciones de las BVIs se detallan en la tabla 3.2. En la figura 3.1 se despliega la clasificación jerárquica completa de las BVIs.

Interacciones Básicas de Visualización (BVIs)	
Operaciones gráficas	Son aquellas operaciones que modifican la apariencia de la visualización. En este grupo se encuentran las operaciones para codificar datos, configurar valores gráficos y manipular objetos, entre otras.
Operaciones sobre conjuntos	Son aquellas operaciones que manipulan conjuntos de objetos. En este grupo se encuentran operaciones para crear conjuntos, eliminar conjuntos y agrupar conjuntos, entre otras.
Operaciones sobre los datos	Son aquellas operaciones que manipulan los datos codificados en la visualización. En este grupo se encuentran operaciones para incorporar datos, eliminar datos y generar datos derivados, entre otras.

Tabla 3.2: Interacciones Básicas de Visualización (BVIs) [CR96].

Las operaciones agrupadas en estas tres clases afectan diferentes etapas del proceso de visualización. Sin embargo, la ejecución de operaciones que afecten el flujo de control o de los datos puede causar efectos secundarios en la representación gráfica de los objetos. Al eliminar un dato, por ejemplo, se modifica el estado interno de la base de datos, pero esto también causa una modificación en la representación gráfica ya que la representación del elemento eliminado también deberá ser eliminada. Esto motiva que Chuah y Roth [CR96] presenten la posibilidad de combinar BVIs con el objeto de soportar tareas complejas. En este contexto, definen tres tipos de composición de BVIs:

- **Composición Independiente:** Las operaciones involucradas no están relacionadas y pueden ejecutarse en paralelo o en cualquier orden.
- **Composición en Conjunto:** Las operaciones se agrupan y la ejecución de una provoca

⁹*Framework*: marco de referencia.

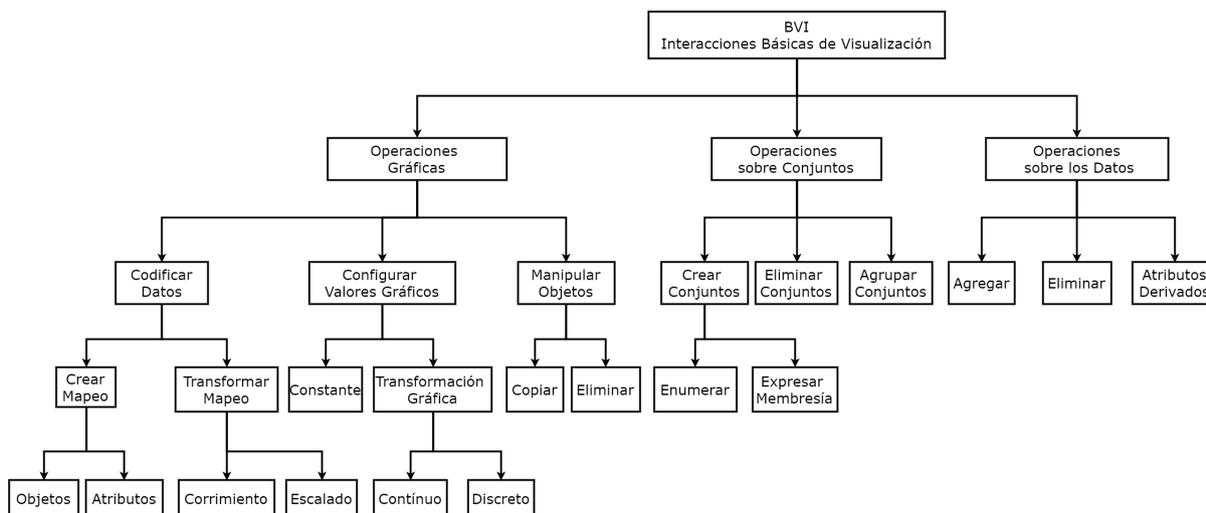


Figura 3.1: Clasificación Jerárquica de las BVI de Chuah y Roth [CR96].

la ejecución de las demás. El orden de ejecución no es relevante.

- **Composición en Secuencial:** Un grupo de operaciones se ejecuta en un orden determinado. La ejecución de la primera operación desencadena la ejecución de las demás.

Un problema importante del *framework* de BVI es que debido a su generalidad no ofrece la posibilidad de trabajar en alto grado de detalle con operaciones específicas de un dominio en particular.

Dos años más tarde, Dix y Ellis [DE98] afirman que la incorporación de elementos simples interactivos a cualquier representación visual estática la vuelve más poderosa. En este contexto, construyen una taxonomía de interacciones con el objetivo de establecer un marco de referencia que sugiera qué interacciones podría ser útil agregar a una o más representaciones estáticas. Los tipos de interacciones destacadas por los autores se describen en la tabla 3.3.

Taxonomía de Interacciones de Dix y Ellis

Foco y resaltado	Permiten resaltar un conjunto de datos de interés.
Acceso a información extra	En lugar de representar simultáneamente toda la información, suele ser útil contar con interacciones que permitan revelar información extra, ya sea a través de técnicas de <i>drill down</i> ¹⁰ o enlaces en un hipertexto.

Sigue en la página siguiente.

¹⁰*Drill down:* profundizar, examinar algo en detalle. En el contexto de visualización significa obtener detalles de los datos a partir de un resumen.

Taxonomía de Interacciones de Dix y Ellis

Overview y contexto	Técnicas como <i>zooming</i> y <i>fish-eyes</i> ¹¹ permiten al usuario tener un <i>overview</i> de todos los datos y/o detalle de un subconjunto de interés. Las vistas <i>fish-eyes</i> , tales como Table Lens [RC94], proveen detalle y contexto en todo momento, mientras que la técnica de <i>zooming</i> requiere de una interacción del usuario para hacerse efectiva. Estas técnicas difieren de los enlaces de acceso a información extra, ya que con las técnicas de <i>overview</i> y contexto se muestran las mismas características a diferentes resoluciones, mientras que con los enlaces, por ejemplo, distintas facetas de los datos son reveladas por demanda. La técnica de <i>drill down</i> , en cambio, puede ubicarse en ambas categorías, ya que puede revelar nuevas facetas de los datos o revelar las mismas características con más detalle.
Modificar parámetros visuales (sin cambiar la representación)	Se trata de aquellas interacciones que permiten modificar los parámetros visuales sin cambiar la técnica de representación utilizada. Un ejemplo clásico consiste en modificar la representación visual de un ítem de dato en función de una selección.
Modificar la técnica de representación (sin cambiar los datos)	No todas las técnicas de representación son apropiadas para todos los tipos de datos. Puede resultar útil proveer al usuario la funcionalidad de modificar la técnica de representación utilizada con el objetivo de obtener nuevos puntos de vista.
Fusión temporal (para representaciones conectadas)	La fusión temporal permite conectar múltiples representaciones haciendo uso del tiempo y la interactividad. La posibilidad de cambiar la técnica de representación o mas aún, de representar los datos con más de una técnica simultáneamente deja al usuario la tarea de determinar las relaciones entre los datos. Es por esto que resulta imprescindible la fusión entre las distintas representaciones.

Tabla 3.3: Taxonomía de Interacciones de Dix y Ellis [DE98].

En 2002, Keim [Kei02] propone una clasificación de técnicas de visualización y minería de datos basada en tres criterios: el tipo de los datos a ser visualizados, la técnica de visualización empleada y las técnicas de distorsión e interacción aplicadas. Keim hace una distinción explícita entre las técnicas de interacción y las técnicas de distorsión. Define las técnicas de interacción como aquéllas que permiten al usuario interactuar directamente

¹¹La técnica de *fish-eyes* muestra un área de interés con gran detalle, aumentando su tamaño y el resto de las áreas con un menor tamaño y una disminución del detalle en forma progresiva a medida que la distancia de los elementos al área de interés aumenta.

con las visualizaciones y modificarlas dinámicamente según sus objetivos de exploración y que además permiten conectar múltiples visualizaciones independientes y relacionarlas. Las técnicas de distorsión se distinguen como técnicas que permiten mostrar porciones de los datos con un alto nivel de detalle preservando un *overview* de los mismos. Además, Keim utiliza los términos “interactivo” y “dinámico” para distinguir los casos en los que los cambios en la visualización se realizan manualmente y los casos en los que se realizan de forma automática respectivamente. En la tabla 3.4 se describe la taxonomía de Keim para las técnicas de interacción y distorsión.

Taxonomía de Keim para las Técnicas Interacción y Distorsión

Proyecciones dinámicas	Consiste en modificar dinámicamente las proyecciones con el objetivo de explorar un conjunto de datos multi-dimensional. Un ejemplo clásico de este tipo de proyecciones es el sistema Grand Tour [Asi85] que intenta mostrar todas las proyecciones significativas en dos dimensiones de un conjunto de datos multi-dimensional (ver figura 3.2(a)). Sistemas que soportan la técnica de proyecciones dinámicas son XGobi [SCB92, BCS96] y ExplorN [CWL96].
Distorsión interactiva	Las técnicas de distorsión preservan un <i>overview</i> de los datos durante el proceso de exploración a través de operaciones <i>drill-down</i> . La idea básica es mostrar porciones de los datos a un nivel de detalle muy alto mientras que el resto del conjunto de datos se muestra a un nivel de detalle muy bajo. Han sido presentadas y desarrolladas muchas técnicas de distorsión y muy variadas. Entre las más populares se encuentran las técnicas de distorsión jerárquicas y de grafos. Ejemplos de técnicas de distorsión incluyen la Pared Perspectiva [MRC91], los <i>Displays Bifocales</i> [SA82] y Visualizaciones Hiperbólicas [LRP95, MB95, UCM13a]. En la figura 3.2(d) se muestra un ejemplo de la distorsión esférica provista por el Framework Escalable de Kreuseler <i>et al.</i> [KLS00].
<i>Zooming</i> interactivo	Permite representar el conjunto de datos a un nivel de compresión muy grande con el objetivo de brindar un <i>overview</i> del conjunto de datos, proveyendo en simultáneo una vista variable de los datos a distintas resoluciones (ver figura 3.2(c)). <i>Zooming</i> no solo significa mostrar los objetos más grandes, sino también modificar su representación para mostrar más detalles a niveles de <i>zoom</i> más altos. A un nivel de <i>zoom</i> muy bajo, los objetos podrían representarse con píxeles, a un nivel medio éstos podrían representarse con íconos y a alta resolución, con un objeto y una etiqueta.

Taxonomía de Keim para las Técnicas Interacción y Distorsión

Filtrado interactivo	Se refiere a la acción de particionar los datos en segmentos para enfocarse en subconjuntos de interés. En este contexto, Keim destaca dos formas para lograr un filtrado interactivo, la primera consiste en seleccionar directamente el subconjunto de datos deseados (navegación) y la segunda consiste en efectuar consultas que especifiquen propiedades del subconjunto deseado. Para grandes conjuntos de datos, el filtrado por navegación se torna muy complicado y el filtrado por consultas muchas veces no resulta en el subconjunto deseado. Esto motivó el desarrollo de muchas técnicas de filtrado para exploración de datos. Un ejemplo representativo de este tipo de herramientas es el Magic Lenses [BSP ⁺ 93, FS95], que utiliza un lente aumentado para aplicar filtrado directamente en la visualización (ver figura 3.2(b)). Los datos ubicados debajo del lente son procesados por un filtro y mostrados de forma diferente en la visualización resultante. Otras herramientas utilizadas para realizar filtrado interactivo son las consultas dinámicas [AS94a, Eic94, GR94] y Polaris [STH02].
<i>Brushing y linking</i> interactivos	Se aplica a múltiples vistas, permitiendo seleccionar un conjunto de datos en una vista y resaltarlos en todas las demás. Los cambios generados interactivamente en una vista se reflejan automáticamente en las vistas restantes. Conectar múltiples visualizaciones a través de esta técnica suele proveer más información que al tratar cada visualización de forma independiente. Sistemas de visualización que incluyen esta técnica son, por ejemplo, Polaris [STH02], XGobi [SCB92] y ComVis [MFGH08]. En la figura 3.2(e) se ilustra un ejemplo de este tipo de interacción en el Spinel Explorer [GGF ⁺ 15], una herramienta de visualización de datos geológicos.

Tabla 3.4: Taxonomía de Keim para las Técnicas Interacción y Distorsión [Kei02].

En 2004, Ward y Yang [WY04] sugieren que al especificar las interacciones utilizando manipulación directa surge el problema de determinar qué objeto del espacio es el foco de la interacción. Para determinar el resultado de una interacción, es necesario conocer dentro de qué espacio de interacción tiene lugar la interacción. En otras palabras, cuando un usuario hace *click* en una locación o área de la pantalla, ¿qué entidades desea referenciar?. Las posibilidades incluyen píxeles, valores de datos, componentes de la estructura de visualización, etc. En su trabajo, Ward y Yang procuran identificar un conjunto completo de espacios de interacción en el contexto de Visualización de Información y de Visualiza-

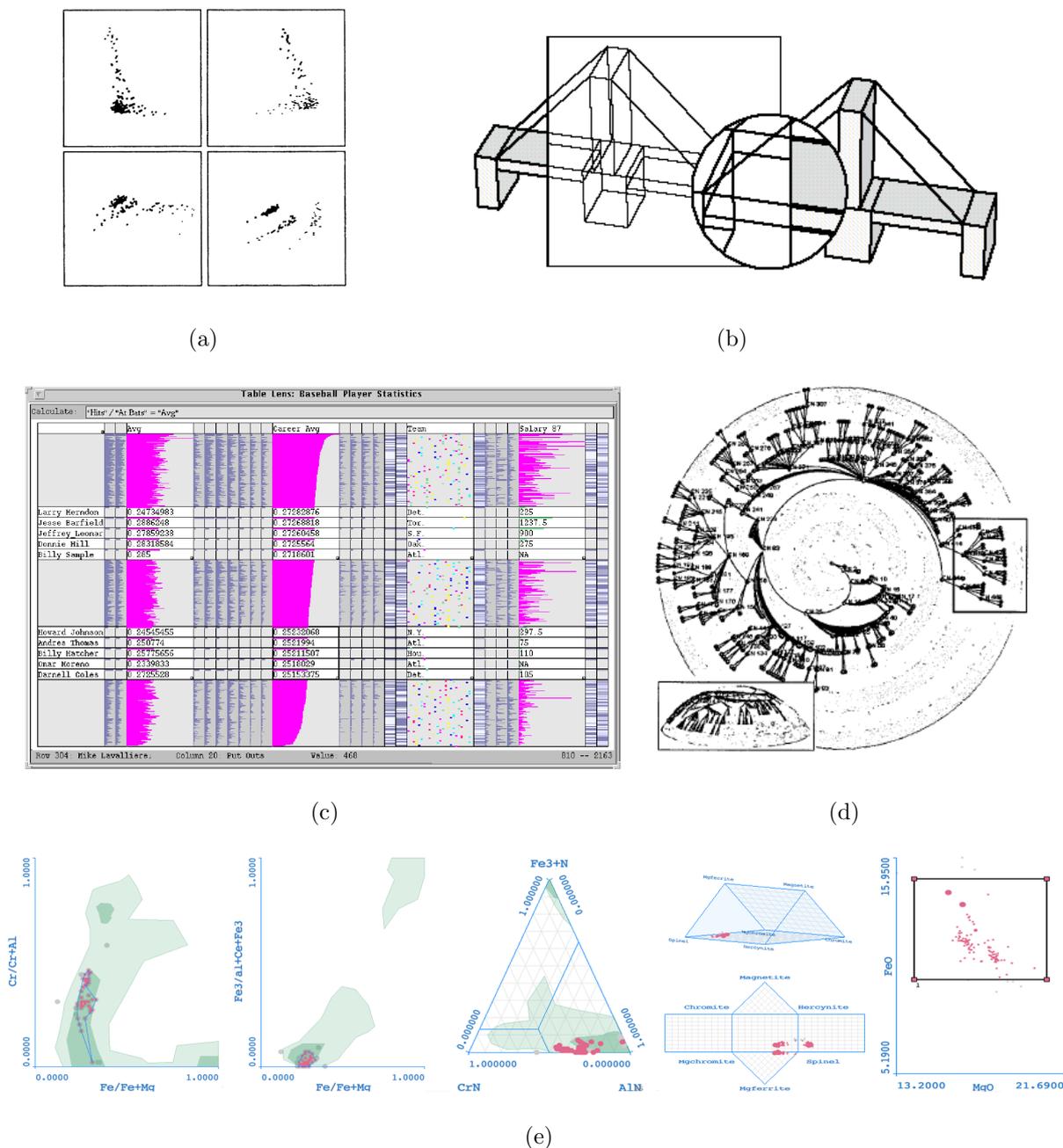


Figura 3.2: Ejemplos de la Clasificación de Keim [Kei02]. (a) Proyecciones dinámicas: 4 proyecciones distintas de un conjunto de datos de diabetes en el *Grand Tour* [Asi85]. (b) Filtrado interactivo: herramienta Magic Lenses aplicada a un puente, el filtro muestra la estructura interna [BSP⁺93]. (c) *Zooming* interactivo: herramienta Table Lens aplicada a un conjunto de datos de *beisball* con unas pocas filas seleccionadas ofreciendo más detalle [RC94]. (d) Distorsión interactiva: distorsión esférica provista por el *Framework Escalable* [KLS00]. (e) *Brushing* y *linking*: selección en el Spinel Explorer [GGF⁺15]. En esta herramienta de visualización de datos geológicos el conjunto de datos seleccionado en una vista se resalta en las otras vistas coordinadas.

ción de Datos en general. Para esto, proponen un *framework* de técnicas de interacción, en el cual identifican distintas clases de operaciones de interacción que se definen en términos de sus operadores y operandos (con operandos se refiere al espacio de interacción¹²). Entre los distintos operadores de interacción distinguen los operadores de navegación, de selección y de distorsión. Este trabajo es extendido en 2010 por Ward *et al.* [WKG10] eliminando del conjunto de operadores a los operadores de distorsión e incorporando los operadores de filtrado, reconfiguración, conexión, codificación, elaboración/abstracción e híbridos.

En su *framework*, Ward *et al.* definen un conjunto amplio de interacciones identificando el tipo del operador de interacción (navegación, selección, etc.), el espacio de interacción (pantalla, valores de datos, estructura de los datos, atributos, estructura de la visualización) y los parámetros asociados a los operadores de interacción (foco, alcance, transformación y *blender*).

En la tabla 3.5 se describe el *framework* de Ward *et al.*, especificando detalladamente la clasificación de operadores, operandos y parámetros.

<i>Framework de Ward et al.</i>	
Operadores de Interacción	
Navegación	Alteran la posición de la cámara y el escalado de la vista. Ejemplos de estos operadores son los operadores de traslación, rotación y <i>zooming</i> .
Selección	Permiten identificar un objeto, una colección de objetos o regiones de interés que estarán sujetos a operaciones adicionales tales como resaltado, eliminación o modificación.
Filtrado	Reducen el tamaño de los datos mapeados a la pantalla, ya sea eliminando registros, dimensiones o ambos.
Reconfiguración	Modifica la forma en que los datos son mapeados a entidades o atributos gráficos, tales como ordenar los datos o <i>layouts</i> ¹³ , o proponer una forma distinta de visualizar los datos.
Conexión	Permiten conectar distintas vistas u objetos con el objetivo de revelar ítems relacionados.
Elaboración/ abstracción	Permiten modificar el nivel de detalle.
Codificación	Permiten modificar los atributos gráficos, tales como el tamaño de un punto o el color de una línea, con el objetivo de revelar características potenciales.

Sigue en la página siguiente.

¹²Espacio de Interacción: sección del espacio sobre la cual se aplica un operador de interacción [WY04].

¹³*Layout*: disposición de los objetos en pantalla.

Framework de Ward et al.

Híbridos Permiten combinar varios de los operadores presentados.

Operandos: Espacios de Interacción

Espacio de la pantalla (píxeles) Al aplicar interacciones de navegación sobre el espacio de la pantalla no se agregan nuevos datos, sino que el proceso consiste en operaciones a nivel de píxeles. En una selección a este nivel cada píxel es clasificado como seleccionado o no.

La distorsión a nivel del espacio de la pantalla involucra transformaciones sobre píxeles. Ejemplos de técnicas a nivel de pantalla son los lentes *fish-eyes* [Fur86] y la metáfora de la “hoja de goma” (*rubber sheet*) que permite visualizar layouts grandes y complejos en pantallas pequeñas [SSTR93]. Las regiones de píxeles pueden ser magnificadas o reducidas con el objetivo de mostrar más en detalle un área de foco en particular (ver figuras 3.3(a) y 3.4(a)).

Espacio de la estructura de los datos Los datos pueden estructurarse de varias formas distintas: listas, tablas, grillas, jerarquías y grafos. Por cada estructura se pueden desarrollar mecanismos de interacción para indicar qué partes de la estructura se desea manipular y cómo realizar esta manipulación y aplicar *zooming* en un espacio de estructuras de los datos (que involucra el mostrado de datos más detallados, ver figura 3.4(b)).

Las técnicas de selección en este espacio involucran el mostrado de la estructura permitiendo al usuario identificar regiones de interés dentro de la misma. Un ejemplo de selección en este contexto es la provista por InterRing [YWR02], una herramienta de visualización jerárquica que permite seleccionar semi-automáticamente los nodos de acuerdo a su estructura jerárquica (ver figura 3.3(c)).

Espacio del objeto (superficies 3D) En este espacio los datos se mapean a un objeto geométrico y este objeto o sus proyecciones pueden estar sujetos a interacciones y transformaciones.

Las técnicas de navegación en este contexto consisten en moverse alrededor de los objetos observando dónde están mapeados los datos. El sistema debe soportar vistas globales y vistas cercanas.

La selección en el espacio del objeto permite seleccionar objetos mediante interacción con el *mouse*.

Ejemplos de distorsión en este espacio son las proyecciones hiperbólicas [Mun97] y la Pared Perspectiva [MRC91] (ver figura 3.3(d)).

Espacio de los atributos Componentes de las entidades gráficas. Similar al espacio de los valores de los datos, pero aplicado a los atributos.

Un paneo en este espacio involucra modificar el rango de valores de interés, mientras que aplicar *zoom* puede involucrar el escalado de los atributos o agrandar el rango de valores de interés.

Framework de Ward et al.

Espacio del valor de los datos (valores de datos multivariados)	<p>En este contexto, las interacciones de navegación involucran la utilización de los valores de los datos como especificación de la vista.</p> <p>Al hacer un paneo o <i>zoom</i> se modifican los datos mostrados. El paneo modifica el valor de comienzo del rango, y el <i>zoom</i> disminuye el tamaño del rango de los datos mostrados.</p> <p>La selección en este espacio es similar a una consulta en una base de datos donde el usuario especifica el rango de los valores de los datos para una o varias dimensiones. Esto se puede lograr vía manipulación directa a través de un <i>brushing</i> [MW95], a través de <i>sliders</i>¹⁴ [Shn94], etc.</p> <p>En este contexto, la distorsión involucra transformaciones de los valores de los datos previas a la visualización. Un ejemplo de distorsión en este espacio es el <i>zooming</i> dimensional provisto por la herramienta XmdvTool [FWR00], donde cada dimensión de un subconjunto de datos seleccionados se escala para que el subconjunto llene el área de mostrado.</p> <p>En la figura 3.3(b) se muestra un ejemplo de selección en este espacio utilizando manipulación directa mediante una técnica de <i>brushing</i> (a la izquierda) y el resultado de aplicar distorsión sobre esa selección con la herramienta XmdvTool (a la derecha).</p>
---	--

Espacio de la estructura de la visualización	<p>La visualización consiste en una estructura relativamente independiente de los valores de los atributos y de la estructura de los datos. Por ejemplo, la grilla donde se dibuja una matriz de <i>scatterplots</i> o los ejes utilizados en innumerables técnicas de visualización son componentes de la estructura de la visualización que pueden ser utilizados como foco de interacciones.</p> <p>En cuanto a la selección en el espacio de la visualización, operaciones típicas pueden elegir qué componentes mostrar, mover o acomodar.</p> <p>En coordenadas paralelas, por ejemplo, podría ser posible seleccionar un eje y arrastrarlo a una nueva locación con el objetivo de descubrir una nueva relación entre los datos.</p> <p>Un buen ejemplo de distorsión en el espacio de la visualización es la técnica Table Lens [RC94], que permite al usuario transformar filas y/o columnas de una hoja de cálculo para proveer múltiples niveles de detalle (ver figura 3.3(e)). En la literatura suele denominarse sustrato espacial a este espacio.</p>
--	--

Parámetros del Operador de Interacción

Foco	Ubicación, dentro del espacio, del centro de la zona de interés del usuario.
------	--

Sigue en la página siguiente.

¹⁴*Slider*: elemento en una interfaz gráfica que permite seleccionar un valor desplazando un indicador.

<i>Framework de Ward et al.</i>	
Alcance	Rango, dentro del espacio, que define los límites de la interacción. La métrica utilizada para especificar este rango depende del espacio de interacción (píxeles para el espacio de la pantalla, por ejemplo).
Transformación	Función aplicada a las entidades ubicadas dentro del alcance de la interacción.
<i>Blender</i>	Cómo manejar aquellas partes del espacio afectadas por más de una interacción.

Tabla 3.5: *Framework* de Ward *et al.* [WY04, WGK10].

En 2006, Wilkinson [Wil06] presenta un conjunto de controles para interactuar con gráficos estadísticos. La clasificación de Wilkinson se detalla en la tabla 3.6

Clasificación de controles de Wilkinson	
Filtrado	Wilkinson clasifica las técnicas de filtrado en función del conjunto de restricciones que definen el filtro. Estas restricciones pueden ser categorías o rangos de valores continuos. Dentro de las técnicas de filtrado distingue el filtrado categórico (para restricciones categóricas), el filtrado continuo (para restricciones continuas), el filtrado múltiple (se requiere la aplicación de múltiples filtros) y el filtrado rápido (para aplicaciones que requieren que la respuesta del filtrado se refleje rápidamente). Ver figura 3.5.
Navegación	Dentro de las técnicas de navegación, distingue la técnica de <i>zooming</i> , <i>panning</i> ¹⁵ y <i>lensing</i> ¹⁶ . Ver figura 3.6.
Manipulación	Define las técnicas de manipulación como aquellas que provocan una modificación directa completa de la visualización. Entre estas técnicas distingue el arrastrado de nodos (para visualizaciones que representan conexión entre elementos) y el ordenamiento categórico.
<i>Brushing y Linking</i>	La técnica de <i>brushing</i> resalta los objetos contenidos en una región seleccionada de un gráfico y la técnica de <i>linking</i> conecta los elementos resaltados entre dos o más gráficos de forma tal que en todos se resalten los mismos casos u objetos. Dentro de estas técnicas se distingue entre la forma del <i>brush</i> , la lógica del <i>brush</i> y el <i>brushing</i> rápido.
Rotación	En escenas 3D, es usual requerir la rotación de los objetos para comprender la estructura o poder ver qué hay detrás.

Sigue en la página siguiente.

¹⁵*Panning*: mover la cámara de forma horizontal y/o vertical para obtener una vista panorámica de la visualización.

¹⁶*Lensing*: técnica de *zoom* que permite establecer el foco en una región proveyendo a la vez contexto.

Clasificación de controles de Wilkinson

Animación	La animación puede proveer <i>insight</i> en distintos tipos de gráficos. En general, los sistemas que proveen control sobre la animación son más útiles que aquéllos que se limitan simplemente a ejecutarla repetidamente una y otra vez. Es importante que el usuario pueda seleccionar <i>frames</i> , pausar, adelantar o atrasar la animación.
-----------	--

Tabla 3.6: Clasificación de Controles de Wilkinson [Wil06].

En 2007, Yi *et al.* [YaKSJ07] publicaron un estudio exhaustivo referido al rol de las técnicas de interacción en el área de Visualización de Información. En su trabajo, Yi *et al.* clasifican las técnicas de interacción en siete categorías de alto nivel en función de la intención del usuario: selección, exploración, reconfiguración, codificación, abstracción/elaboración, filtrado y conexión. Estas interacciones se detallan en la tabla 3.7

Clasificación de Técnicas de Interacción de Yi *et al.*

Selección	Las técnicas de selección proveen al usuario la posibilidad de marcar ítems de datos como interesantes, facilitando de este modo el seguimiento de los mismos y logrando su identificación cuando la visualización cambia.
Exploración	Las técnicas de exploración permiten a los usuarios examinar diferentes subconjuntos de los datos. Usualmente, por limitaciones de <i>hardware</i> y fundamentalmente por limitaciones perceptuales y cognitivas de los usuarios, no es posible visualizar el conjunto de ítems de datos completo en un determinado instante de tiempo. Es por esto, que los usuarios suelen examinar un conjunto o un subconjunto de los datos y, mediante técnicas de exploración, visualizar otro subconjunto de los datos.
Reconfiguración	Las técnicas de reconfiguración proveen al usuario distintas perspectivas de los datos modificando la organización espacial de la representación de los mismos. Éstas permiten modificar características particulares de las técnicas de visualización aplicada, modificar los atributos asociados a los ejes, mover ítems de datos de forma libre, reducir oclusiones, etc.
Codificación	Las técnicas de codificación permiten al usuario alterar la representación visual de los datos incluyendo la apariencia visual (color, tamaño, forma, etc.) de cada elemento de dato. Técnicas de codificación incluyen modificar la técnica de visualización aplicada para mostrar los datos (cambiar de un gráfico de torta a un histograma, por ejemplo) y alterar la apariencia de los objetos gráficos involucrados en la representación.

Sigue en la página siguiente.

Clasificación de Técnicas de Interacción de Yi *et al.*

Abstracción/elaboración	Las técnicas de abstracción y elaboración permiten a los usuarios ajustar el nivel de abstracción de la representación de los datos. Este tipo de interacciones permiten alterar la representación partiendo de un <i>overview</i> hasta obtener detalles de un dato o un subconjunto de los datos en particular.
Filtrado	Las técnicas de filtrado permiten al usuario modificar el conjunto de ítems de datos representado en base a un conjunto de condiciones específicas, de modo tal que únicamente los ítems que cumplen esas condiciones son representados. Aquellos ítems que no cumplan con el criterio especificado se ocultan de la vista o se muestran de forma diferente.
Conexión	Las técnicas de conexión permiten al usuario resaltar asociaciones y relaciones entre ítems de datos y mostrar/resaltar ítems de datos relevantes a un ítem de dato especificado. Un ejemplo de este tipo de técnicas es la técnica de <i>brushing</i> , que permite relacionar ítems de datos en múltiples vistas correlacionadas.

Tabla 3.7: Clasificación de Yi *et al.* para las Técnicas de Interacción [YaKSJ07].

En 2014, Liu *et al.* [LCWL14] extendieron el trabajo de Yi. *et al.* [YaKSJ07] incorporando una actualización de las técnicas de interacción desarrolladas hasta ese año, clasificándolas en dos categorías: interacciones WIMP (*windows, icons, mouse, pointers*) e interacciones post-WIMP. Las técnicas de interacciones WIMP se focalizan principalmente en el estudio de cómo los usuarios interactúan con las herramientas de visualización a través del uso de un *mouse* o un teclado. Las técnicas de interacciones post-WIMP, en cambio, buscan explorar cómo los usuarios interactúan con las herramientas de visualización en dispositivos que van más allá del paradigma de ventanas, íconos, *mice* y punteros, tales como los dispositivos de pantalla táctil. Ejemplos típicos de interacciones WIMP incluyen interacciones básicas de selección, filtrado, *brushing* y resaltado [LZP⁺09, WWL⁺10, LZP⁺12], interacciones avanzadas de comparación visual [TFJ12] y navegación [HD12, MGT⁺03, TS08, DRRD12]. Las interacciones post-WIMP son aquellas asociadas, por ejemplo, a interfaces táctiles [WLJ⁺12, BHP⁺12, IFP⁺12], a interfaces de Realidad Virtual [TJ00, JGH⁺08, Gil16], a interfaces de Realidad Aumentada [LGH⁺96, KHG03, MSS12, LSH⁺15, SH16], etc.

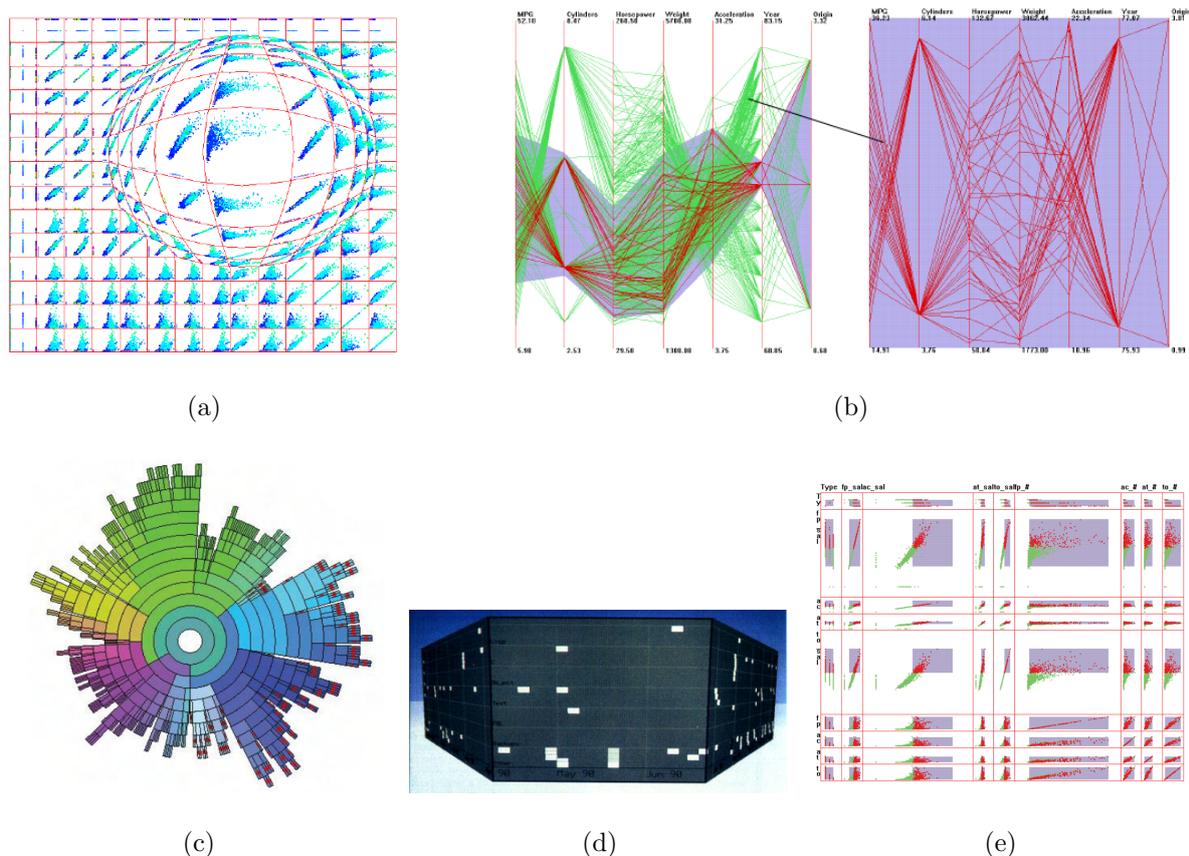


Figura 3.3: Espacios de Interacción de Ward y Yang [WY04]. (a) Espacio de la pantalla (píxeles): en la figura se muestra una matriz de dispersión en la que se ha seleccionado y magnificado a nivel de píxeles un centro de foco utilizando una técnica de lente confocal. (b) Espacio de valor de los datos: en la figura se muestra la aplicación de la técnica de distorsión utilizando la herramienta XmdvTool [FWR00]. Se efectúa una selección n-dimensional sobre un eje (imagen de la izquierda) y se escalan todas las dimensiones seleccionadas para llenar el espacio de mostrado (imagen de la derecha). (c) Espacio de la estructura de los datos: selección mediante InterRing [YWR02]; en vez de efectuar una selección de un nodo a la vez, los nodos marcados en rojo han sido seleccionados mediante una consulta de usuario. (d) Espacio de los objetos: la Pared Perspectiva [MRC91], en este caso se visualiza globalmente un documento sobre una superficie que simula estar sobre una pared. Esta superficie se puede mover a izquierda y a derecha de lo que se considera el foco y que constituye la vista cercana de una parte del documento. (e) Espacio de la estructura de la visualización: la técnica de distorsión modifica los elementos subyacentes de la visualización. El ejemplo de la figura está inspirado en Table Lens [RC94] y muestra una matriz de *scatterplots* con dos celdas (y sus correspondientes filas y columnas) magnificadas.

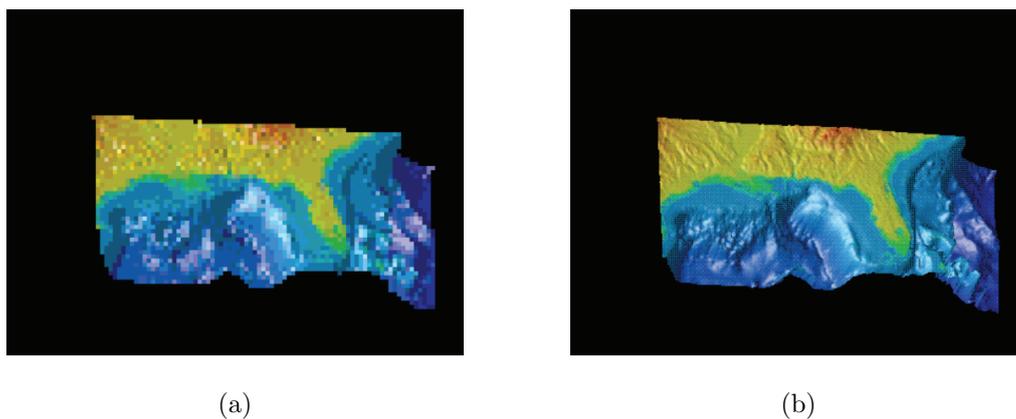
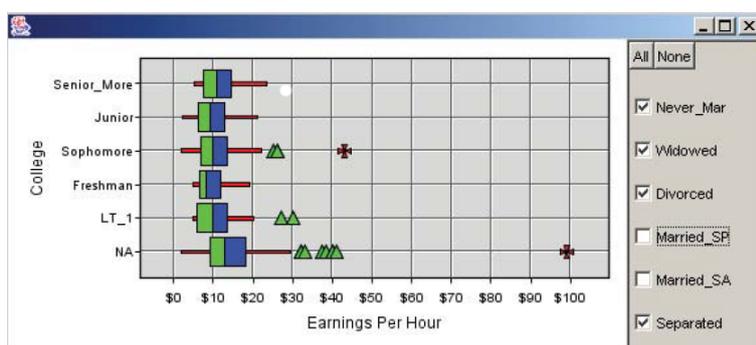
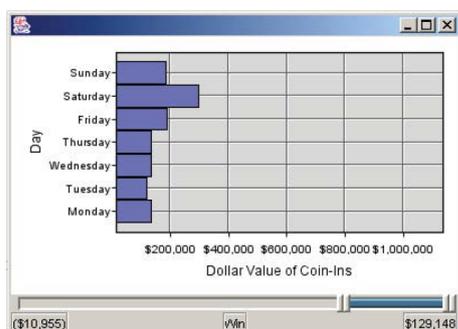


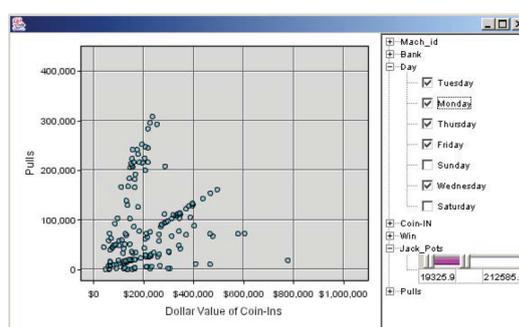
Figura 3.4: Técnica de *Zoom* Aplicada al Espacio de la Pantalla y al Espacio de la Estructura de los Datos [WY04]. (a) En el espacio de la pantalla los píxeles son replicados para alcanzar el tamaño seleccionado. (b) En el espacio de la estructura de los datos, se puede efectuar un re-muestreo de los datos para ajustar la resolución.



(a)



(b)



(c)

Figura 3.5: Clasificación de Técnicas de Filtrado de Wilkinson [Wil06]. (a) Filtrado categórico a través *checkboxes*. (b) Filtrado continuo a través de un *slider*. (c) Filtrado múltiple.

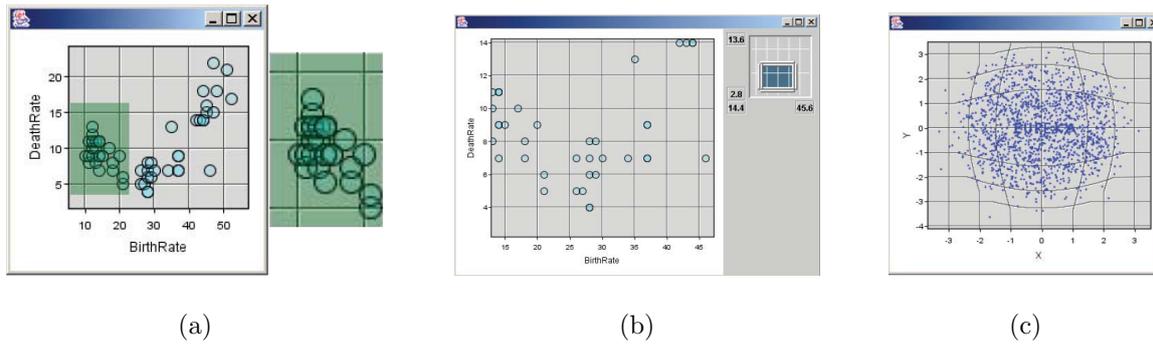


Figura 3.6: Clasificación de Técnicas de Navegación de Wilkinson [Wil06]. (a) *Zooming*. (b) *Panning*. (c) *Lensing*.

3.2.2. Taxonomías de las Dimensiones de las Técnicas de Interacción

Muchos diseños de interfaces que soportan interacción Humano-Computadora han sido creados en base al “Ciclo de Acción” de Norman [Nor88]. Norman identifica dos *gulfs* existentes entre el humano, que tiene un objetivo que cumplir, y el mundo, típica pero no necesariamente una computadora, que tiene que ser modificado para que el objetivo sea alcanzado. Este concepto tiene relevancia directa con el área de visualización, en la que el objetivo es un modelo mental y el mundo es una representación visual de los datos. En la figura 3.7 se ilustra el ciclo de acción de Norman, donde el *gulf* de ejecución refleja el problema de saber qué hacer para lograr un cambio en el mundo (en el contexto de visualización, un cambio en la representación de los datos) y el *gulf* de evaluación refleja el problema de cómo interpretar y evaluar los cambios en el mundo.

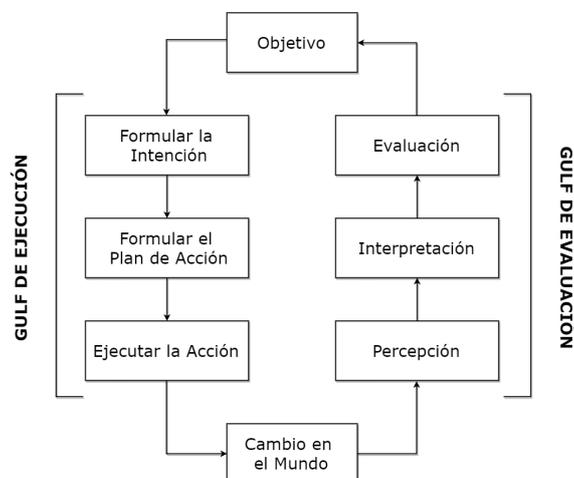


Figura 3.7: Ciclo de Acción de Norman [Nor88].

Muchos términos diferentes han sido utilizados para caracterizar las distintas formas en que un usuario puede interactuar con un *display* visual. Términos como “manual” o “manipulación directa” son virtualmente sinónimos para describir aquellas interacciones en las cuales, por ejemplo, un *mouse* es presionado o arrastrado sobre una pantalla para mover un objeto. En este contexto, Norman [Nor88] utilizó los términos “primera persona–hágalo usted mismo” y “tercera persona–modo comando”.

En 1997, Tweedie [Twe97] buscó caracterizar el espacio de técnicas de visualización interactiva existentes hasta el momento. Clasificó una selección de técnicas de visualización con respecto a los datos representados, la forma de representar los datos visualmente y las formas de interacción utilizadas. Con respecto a las interacciones, Tweedie utiliza la clasificación de Lunzer [Lun96] definiendo cinco tipos de interacción: manual, mecanizada, basada en instrucciones, orientable y automática. Además, Tweedie describe la interactividad en las herramientas de visualización como de manipulación directa y de manipulación indirecta. La caracterización de Tweedie se describe en la tabla 3.8.

Caracterización de Tweedie	
Tipos de Interacción	
Manual	Un ejemplo de este tipo de interacción consiste en mover un objeto en pantalla utilizando un dispositivo físico, como por ejemplo, un <i>mouse</i> .
Mecanizada	Un ejemplo de interacción mecanizada consiste en seleccionar un conjunto de objetos haciendo uso de un <i>slider</i> o herramienta para tal fin.
Basada en instrucciones	Este tipo de interacciones utilizan, por ejemplo, una fórmula en una hoja de cálculo.
Orientable	Este tipo de interacciones son realizadas mediante un algoritmo para llevar a cabo una tarea de una determinada forma.
Automática	Instrucciones que, utilizando un algoritmo, se ejecutan automáticamente.
Tipos de Interactividad	
Manipulación directa	Se trata de una réplica literal del comportamiento físico del mundo real. La manipulación puede ser manual (el usuario mismo mueve el objeto) o mecanizada (utilizando una herramienta para mover el objeto).
Manipulación indirecta	Este tipo de interactividad no se basa en la metáfora física. Un ejemplo de manipulación indirecta puede involucrar herramientas que deformen un objeto en un entorno de realidad virtual.

Tabla 3.8: Caracterización de Tweedie [Twe97].

En 2007, Spence [Spe07] aborda dos aspectos de la interacción en visualización: el espacio de información y el modo de interacción. Con respecto al espacio de información, observa dos tipos: espacio de información discreta y espacio de información continua. Además de esta distinción en la naturaleza del espacio de información, Spence se enfoca en la forma en la que el usuario puede interactuar con esa información. Basándose en el trabajo de Norman [Nor88] y de Tweedie [Twe97] identifica tres modos de interacción: interacción continua, interacción por pasos e interacción pasiva. Además, utiliza el término “interacción compositiva”, que se refiere a la combinación de estos tres modos. El *framework* de interacciones de Spence se detalla en la tabla 3.9.

<i>Framework</i> de Interacciones de Spence	
Espacios de Información	
Continuo	El rango de la información corresponde a un espacio continuo.
Discreto	La información se organiza como una colección de entidades discretas.
Modos de Interacción	
Interacción continua	El usuario interactúa con la visualización a través de movimientos continuos, independientemente de si el espacio de información explorado es continuo o discreto. Aunque la respuesta del sistema pueda introducir cierto retardo, el usuario percibe la exploración como continua.
Interacción por pasos	Una acción mínima (un <i>click</i> con el <i>mouse</i> , por ejemplo) provoca un cambio en el espacio de información discreta.
Interacción pasiva	Se refiere al proceso cognitivo al que se somete el usuario y no necesariamente puede tener lugar una acción física. Este proceso puede ser altamente complejo cognitiva y visualmente. Hay dos aspectos de la interacción pasiva que no deben ser pasados por alto. Primero, que durante el uso de una herramienta de visualización, los usuarios utilizan la mayor parte del tiempo para observar, involucrando en este proceso un gran número de movimientos oculares y procesamiento cognitivo. En segundo lugar, la interacción pasiva no implica necesariamente una representación estática. La representación de los datos puede cambiar por muchas razones, mientras el usuario observa alimentando su proceso cognitivo.

Tabla 3.9: *Framework* de Interacciones de Spence [Spe07].

3.2.3. Taxonomía de las Tareas

Además de la representación visual, en visualización también deben tenerse en cuenta las tareas que el usuario desea realizar mediante la representación visual. En la literatura existen varias clasificaciones, taxonomías y *frameworks* para describir estas tareas. En el contexto de Visualización de Información, las tareas pueden definirse como las actividades que se llevan a cabo interactivamente sobre una representación visual con un objetivo en particular.

Uno de los primeros trabajos desarrollados en el tema es el de Zhou y Feiner [ZF98], quienes presentaron una taxonomía de tareas visuales. Zhou y Feiner clasificaron las tareas visuales en dos categorías de alto nivel: las tareas visuales que tienen el propósito de *informar* y las que tienen el propósito de *permitir* al usuario la realización de un *cómputo* o *exploración* visual. Las tareas visuales cuyo objetivo es informar, pueden a su vez *resumir* o *elaborar* información. De la misma forma, aquellas tareas que permiten al usuario *explorar* o *computar* pueden a su vez ser clasificadas con respecto a si permiten *buscar* un objeto en particular, *verificar* un hecho o *diferenciar* o *sumar* diferentes valores. Además, Zhou y Feiner definieron un conjunto de tareas visuales de bajo nivel. En la tabla 3.10 se presenta la clasificación de tareas de alto nivel detallando, para cada una de éstas, las tareas visuales de bajo nivel que ayudan a completarlas.

Taxonomía de Tareas Visuales de Zhou y Feiner	
Informar	Elaborar: enfatizar y revelar. Resumir: asociar, <i>background</i> , categorizar, agrupar, comparar, correlacionar, distinguir, generalizar, identificar, ubicar y clasificar.
Permitir	Explorar: - Buscar: categorizar, agrupar, comparar, correlacionar, distinguir, enfatizar, identificar ubicar, clasificar y revelar. - Verificar: categorizar, comparar, correlacionar, distinguir, identificar, ubicar, clasificar y revelar. Computar: - Sumar: correlacionar, ubicar y clasificar. - Diferenciar: correlacionar, ubicar y clasificar.

Tabla 3.10: Taxonomía de Tareas Visuales de Zhou y Feiner [ZF98].

En 2005, Amar *et al.* [AES05] presentaron un conjunto de diez tareas de bajo nivel que capturan en gran medida las actividades de las personas mientras emplean herramientas

de visualización. La taxonomía de tareas de Amar se describe en la tabla 3.11.

Taxonomía de Tareas de Amar <i>et al.</i>	
Tareas de Bajo Nivel	
Recuperar valor	Dado un conjunto de casos específicos, encontrar atributos de esos casos. Ejemplos de este tipo de tareas se desprenden de preguntas tales como ¿cuál es el kilometraje por litro del Audi TT? o ¿cuánto dura la película “Lo que el viento se llevó”?.
Filtrar	Dadas algunas condiciones concretas sobre los valores de los atributos, encontrar los datos que satisfagan esas condiciones. Ejemplos de este tipo de tareas se desprenden de preguntas tales como ¿qué cereales de Kellogg tienen alto contenido en fibra? o ¿qué comedias han ganado premios?.
Calcular el valor derivado	Dado un conjunto de datos, calcular una representación numérica agregada sobre esos datos. Ejemplos de este tipo de tareas se desprenden de preguntas tales como ¿cuál es el ingreso bruto de todos los negocios combinados? o ¿cuántos fabricantes de automóviles hay?.
Encontrar extremos	Buscar datos que poseen un valor extremo de un atributo sobre su rango dentro del conjunto de datos. Ejemplos de este tipo de tareas se desprenden de preguntas tales como ¿qué director/película ha ganado más premios? o ¿qué película de Robin Williams tiene la fecha de lanzamiento más reciente?.
Ordenar	Dado un conjunto de datos, clasificarlos de acuerdo con alguna métrica ordinal. Ejemplos de este tipo de tareas son ordenar autos por peso, o ordenar cereales por la cantidad de calorías que aportan.
Determinar el rango	Dado un conjunto de datos y un atributo de interés, encontrar el intervalo de valores de ese atributo dentro del conjunto. Ejemplos de este tipo de tareas se desprenden de preguntas tales como ¿cuál es el rango de longitud de las películas? o ¿qué actrices están en el conjunto de datos?.
Caracterizar la distribución	Dados un conjunto de datos y un atributo cuantitativo de interés, caracterizar la distribución de los valores de ese atributo sobre el conjunto. Ejemplos de este tipo de tareas se desprenden de preguntas tales como ¿cuál es la distribución de carbohidratos en los cereales? o ¿cuál es la distribución por edad de los compradores?.
Encontrar anomalías	Identificar anomalías dentro de un conjunto de datos con respecto a una relación o expectativa dada. Ejemplos de este tipo de tareas se desprenden de preguntas tales como ¿existen excepciones a la relación entre la potencia y la aceleración? o ¿existen valores atípicos en las proteínas?.

Sigue en la página siguiente.

Taxonomía de Tareas de Amar <i>et al.</i>	
Agrupar	Dado un conjunto de datos, encontrar grupos de valores de atributos similares. Ejemplos de este tipo de tareas se desprenden de preguntas tales como ¿hay grupos de cereales con grasas / calorías / azúcar similares? o ¿hay un grupo de longitudes de película típicas?.
Correlacionar	Dados un conjunto de datos y dos atributos, determinar las relaciones entre los valores de esos atributos. Ejemplos de este tipo de tareas se desprenden de preguntas tales como ¿existe una correlación entre los carbohidratos y la grasa? o ¿existe una tendencia a aumentar la longitud de las películas a lo largo de los años?.

Tabla 3.11: Taxonomía de Tareas Visuales de Amar *et al.* [AES05].

En 2013, Schulz *et al.* [SNHS13] presentaron un espacio de diseño para las tareas en el área de visualización. Este espacio de diseño tiene como propósito ser utilizado para recomendar visualizaciones adecuadas para llevar a cabo determinada tarea. Las dimensiones del espacio de diseño de tareas de Schultz *et al.* son cinco y están relacionadas con las preguntas:

- ¿**Por qué** se lleva a cabo una tarea? Especifica el **objetivo** de la tarea.
- ¿**Cómo** se lleva a cabo una tarea? Especifica el **proceso** a través del cual se lleva a cabo la tarea.
- ¿**Qué** explora una tarea? Especifica las **características** de los datos asociados a la tarea.
- ¿**Dónde** opera la tarea? Especifica tanto el **destino**, como la **cardinalidad** de las entidades de los datos afectados por la tarea.
- ¿**Cuándo** se ejecuta la tarea? Especifica el orden de las tareas.
- ¿**Quién** ejecuta la tarea? Especifica el tipo de usuario.

Los últimos dos aspectos claramente no son propiedades fundamentales de la tarea en sí misma, sino que relacionan la tarea con su contexto. El *Cuándo* establece el contexto temporal de precedencia y sucesión de tareas en una secuencia de tareas y el *Quién* pone en contexto las capacidades y responsabilidades de realizar tareas en un ambiente colaborativo.

En función de estas preguntas, las cinco dimensiones del espacio de diseño de Schulz *et al.* se definen como: objetivo, proceso, características, destino y cardinalidad y se describen en la tabla 3.12.

Espacio de Diseño de Tareas de Schulz <i>et al.</i>	
Dimensiones	
Objetivo	<p>El objetivo de una tarea de visualización define la intención con la que se persigue la tarea. Schulz <i>et al.</i> distinguen tres objetivos de alto nivel:</p> <ul style="list-style-type: none"> - Análisis Exploratorio: Se ocupa de derivar hipótesis de un conjunto de datos desconocido. - Análisis de Confirmación: Tiene como objetivo probar hipótesis encontradas o asumidas sobre un conjunto de datos. - Presentación: Trata de describir y exhibir resultados de análisis confirmados.
Proceso	<p>El proceso a través del cual se lleva a cabo una tarea de visualización (también llamada acción) determina el método para alcanzar el objetivo. Schulz <i>et al.</i> definen las siguientes acciones:</p> <ul style="list-style-type: none"> -Navegación: Cambiar la granularidad de los datos mostrados sin reorganizar los datos en sí. -Reorganización: Ajustar los datos a ser mostrados reduciendo o enriqueciendo el conjunto de datos. -Relación: Abarca todas las acciones que intentan poner los datos en su contexto. Esto se logra buscando similitudes a través de la comparación, buscando diferencias al buscar variaciones o discrepancias, etc.

Sigue en la página siguiente.

Espacio de Diseño de Tareas de Schulz <i>et al.</i>	
Características	<p>Las características de una tarea de visualización captan las facetas de los datos que la tarea pretende revelar. Si bien estas características dependen en gran medida del tipo de los datos que se están visualizando, podemos distinguir entre dos tipos generales:</p> <ul style="list-style-type: none"> -Características de bajo nivel de los datos: Corresponden a observaciones simples sobre los datos. Tales como los valores de los datos asociados a un objeto de dato en particular (búsqueda, identificación), o los objetos de datos correspondientes a un valor de datos en particular (localización). -Características de alto nivel de los datos: Corresponden a patrones más complejos en los datos, tales como tendencias, <i>outliers</i>¹⁷, <i>clusters</i>¹⁸, frecuencia, distribución, etc.
Destino	<p>El destino de una tarea de visualización (a menudo también llamada entidad de datos) determina sobre qué aspecto de las relaciones de los datos se efectúa la tarea. En este contexto se consideran los siguientes aspectos de los datos:</p> <ul style="list-style-type: none"> -Relaciones entre los atributos: Relaciona los objetos de datos con sus atributos. Estas relaciones incluyen: <ul style="list-style-type: none"> -Relaciones Temporales: Relacionan objetos de datos con atributos temporales (intervalos de tiempo, instantes, etc.). -Relaciones Espaciales: Relacionan objetos de datos con atributos espaciales (puntos, trayectorias, áreas geográficas, etc.). -Relaciones estructurales: Conecta objetos de datos entre sí.
Cardinalidad	<p>La cardinalidad de una tarea de visualización especifica cuántas instancias del destino elegido son consideradas. Schulz <i>et al.</i> consideran tres tipos de cardinalidad:</p> <ul style="list-style-type: none"> - Instancia Única. Para resaltar detalles, p. ej. - Múltiples Instancias. Para poner un dato o un conjunto de datos en contexto, p. ej. - Todas las Instancias. Para obtener un <i>overview</i> completo, p. ej.

Tabla 3.12: Espacio de Diseño de Tareas de Schulz *et al.* [SNHS13].

Las dimensiones de diseño no son independientes, sino que se utilizan conjuntamente para constituir el espacio de diseño. Una tarea individual puede representarse como un

¹⁷*Outlier*: ítem de dato cuyo valor cae fuera de los límites que toma la mayoría del resto de los valores correspondientes de la muestra. Puede indicar anomalías.

¹⁸*Cluster*: agrupamiento.

punto en este espacio de diseño. Las tareas individuales se encadenan para formar flujos de trabajo.

Con el objetivo de unificar las clasificaciones de tareas de visualización de alto y bajo nivel, Brehmer y Munzner [BM13] presentaron una clasificación multi-nivel de tareas abstractas de visualización. Ésta se definió en el contexto del Modelo Anidado para Diseño y Validación de Visualización de Munzner [Mun09] y fue refinada más tarde por Munzner en su libro “*Visualization Analysis and Design*” [Mun14].

La organización de Brehmer y Munzner provee una forma flexible y poderosa para describir las tareas complejas en secuencias de tareas simples independientes y se centra en tres preguntas:

- **¿Por qué** se lleva a cabo la tarea?
- **¿Cómo** se lleva a cabo la tarea?
- **¿Cuáles** son los parámetros de entrada y salida de la tarea?

Los autores mencionan no haber encontrado caracterizaciones previas que respondan estas tres preguntas simultáneamente. Típicamente, las caracterizaciones de bajo nivel se centran en *cómo una tarea es llevada a cabo* y las caracterizaciones de alto nivel se centran en *por qué es llevada a cabo una tarea*. En la figura 3.8 se ilustra la clasificación encapsulando las tres preguntas ya mencionadas.

El *¿Por qué?* de la clasificación permite describir por qué una tarea es realizada e incluye tres niveles de acciones que definen los objetivos del usuario. En el nivel más alto, distinguen entre dos posibles objetivos para los usuarios que desean **analizar** un conjunto de datos, esto es, **consumir** información existente o bien **producir** nueva información. Todos los casos de análisis requieren que el usuario busque aquellos elementos de interés dentro de la visualización. En este contexto el nivel medio de la clasificación cubre el tipo de **búsqueda** que está involucrada en la visualización en términos de si la locación y/o el destino es conocido o no. Una vez que se ha encontrado un objeto o un conjunto de objetos como resultado de una búsqueda, un objetivo es efectuar **consultas** sobre estos objetos con el objetivo de **identificar**, **comparar** o **resumir**. El nivel más bajo de la clasificación corresponde al tipo de consultas.

El *¿Cómo?* de la clasificación aborda cómo puede construirse un lenguaje de visualización a partir de un conjunto de decisiones de diseño. La tabla 3.13 muestra la clasificación

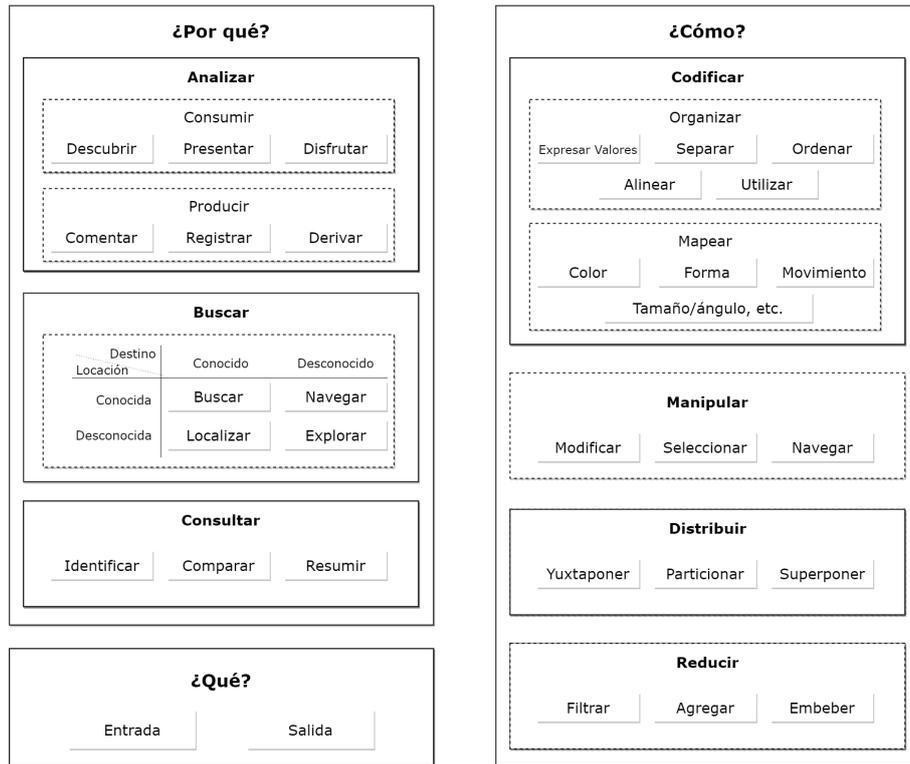
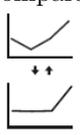


Figura 3.8: Clasificación Multi-Nivel de Tareas de Brehmer y Munzner [BM13, Mun14].

del conjunto de decisiones de diseño presentadas por Brehmer y Munzner.

Clasificación de Tareas de Brehmer y Munzner		
¿Por qué?		
Análisis	Consumir	<p>Descubrir</p>  <p>Se refiere a la utilización de una herramienta de visualización para obtener nuevos conocimientos. Este uso permite generar nuevas hipótesis, encontrar teorías nuevas e incluso verificar si una conjetura es verdadera o falsa.</p>
	Presentar	<p>Presentar</p>  <p>Se refiere al uso de la visualización para comunicar, informar, contar una historia a través de un conjunto de datos o guiar una audiencia a través de una serie de operaciones cognitivas.</p>
	Disfrutar	<p>Disfrutar</p>  <p>Se refiere al encuentro casual del usuario con la visualización. En estos casos el usuario no necesita confirmar o generar una hipótesis, sino que guiado por la curiosidad se puede sentir estimulado y satisfecho con una herramienta de visualización.</p>

Sigue en la página siguiente.

Clasificación de Tareas de Brehmer y Munzner	
Producir	<p>Anotar  Se refiere a la adición de anotaciones gráficas o textuales asociadas con uno o más elementos preexistentes en la visualización.</p>
	<p>Registrar  Se refiere al registro o almacenamiento de elementos de la visualización, tales como capturas de pantalla, marcadores de elementos o ubicaciones, configuraciones de parámetros, registros de interacciones, etc.</p>
	<p>Derivar  Se refiere a la producción de nuevos elementos de datos basados en el conjunto de datos existente.</p>
Búsqueda	<p>Buscar  Se conoce qué es lo que está buscando y dónde encontrarlo.</p>
	<p>Localizar  Se conoce qué es lo que está buscando pero no dónde se encuentra.</p>
	<p>Navegar  No se conoce qué es lo que está buscando pero sí su ubicación.</p>
	<p>Explorar  No se conoce qué es lo que está buscando ni su ubicación.</p>
Consultas	<p>Identificar  El alcance de esta consulta es un solo objeto.</p>
	<p>Comparar  El alcance de esta consulta es de múltiples objetos.</p>
	<p>Resumir  El alcance de esta consulta es todos los objetos.</p>

¿Cómo?

Codificación	Organizar	Expresar Valores 	Separar 	Ordenar 	Alinear 	Utilizar 
	Mapear	Color → Hue  → Saturation  → Luminance 	Tamaño, ángulo, curvatura 	Forma 	Movimiento 	
Manipulación	Modificar 	Seleccionar 	Navegar 			

Sigue en la página siguiente.

Clasificación de Tareas de Brehmer y Munzner			
Distribución	Juxtaponer 	Particionar 	Superponer 
Reducción	Filtrar 	Agregar 	Embeber 

Tabla 3.13: Clasificación de Tareas de Visualización de Brehmer y Munzner [BM13, Mun14].

El ¿Qué? de la clasificación de Brehmer y Munzner modela los parámetros de entrada y salida de las tareas. En la bibliografía se han abordado las entradas y salidas involucradas en las tareas en el contexto de visualización desde distintos enfoques. Algunos trabajos se focalizan en el conjunto de datos completo, representados como tablas compuestas de valores y atributos, o grafos, compuestos por nodos y enlaces [Shn96]. En otros casos, se permite una especificación más precisa de la semántica de los atributos de los datos, tales como atributos categóricos, ordinales y cuantitativos [CMS99]. Para su clasificación de tareas y con el objeto de adecuarse a todas las clasificaciones, Brehmer y Munzner sugieren una representación que califican como “agnóstica” (*bring your own what*). Es decir que el único requerimiento planteado consiste en distinguir explícitamente las entradas y las salidas de las tareas.

3.3. Interacciones en *Visual Analytics*

El uso de representaciones visuales e interacciones para acelerar la rápida comprensión de datos complejos es lo que distingue a las herramientas de *Visual Analytics* (VA) de otros tipos de herramientas analíticas. Las representaciones visuales invitan al usuario a explorar sus datos. Esta exploración requiere que el usuario pueda interactuar con los datos para encontrar y comprender tendencias y anomalías, aislar y reorganizar la información como sea apropiado, y participar así en el proceso de razonamiento analítico. En esta sección, se abordarán aspectos importantes sobre las técnicas de interacción necesarias para soportar herramientas de VA.

En su libro “*Illuminating the path: the research and development agenda for visual analytics*”, Thomas y Cook [TC05] identifican al menos tres enfoques distintos de la

ciencia de las interacciones en VA. En este contexto, definen tres niveles de interacción. En primera instancia, consideran las interacciones desde el punto de vista de las constantes de tiempo del humano. Este es un punto de vista muy importante ya que todas las interacciones están dirigidas por lo que el usuario es perceptual y cognitivamente capaz de hacer. En segundo lugar, consideran las interacciones en función de cómo son aplicadas. En este contexto, a los usos definidos por Card *et al.* [CMS99] (filtrado, manipulación del mapeo visual, navegación) se incorpora un nuevo uso de las interacciones, específico para VA, que consiste en interacciones para el diálogo del usuario con la información y que incluye interacciones para apoyar procesos tales como comparar y categorizar datos, extraer y recombinar datos, crear y probar hipótesis y enriquecer datos. Finalmente, consideran la naturaleza de las interacciones, incluyendo diferencias entre interacciones aplicadas en entornos de dos dimensiones y de tres dimensiones y cómo éstas se ven afectadas al realizarse sobre dispositivos en uso.

Niveles de Interacción en VA de Thomas y Cook

Niveles de Interacción

Constantes de tiempo del humano	<p>El análisis de las constantes de tiempo del humano para interacción humano-computadora fue inicialmente discutido por Card <i>et al.</i> [CNM83], abordado desde un punto de vista cognitivo por Newell [New94] y presentado en el contexto de visualización por Card <i>et al.</i> [CMS99].</p> <p>Hay tres tipos de constantes de tiempo para el ser humano a las que el sistema de VA deberá adaptarse: la constante de tiempo de procesamiento perceptual (para mantener la ilusión de animación, el sistema debe refrescarse al menos cada 0.1 segundos), la constante de tiempo de respuesta inmediata (una persona puede generar una respuesta inmediata a un estímulo dentro de 1 segundo) y la constante de tiempo de una unidad de tarea (un usuario debería ser capaz de completar una tarea elemental en aproximadamente 10 segundos).</p>
Usos de las interacciones	<p>Card <i>et al.</i> [CMS99] identifican tres usos primarios de las interacciones para Visualización de Información: modificar la transformación de los datos (filtrado), modificar el mapeo visual y modificar la transformación de la vista (navegación). En el contexto de VA se incorpora un cuarto uso: el diálogo a alto nivel del usuario con la información.</p>

Sigue en la página siguiente.

Niveles de Interacción en VA de Thomas y Cook	
Naturaleza de las interacciones	<p>La naturaleza de una interacción se ve afectada si tiene lugar en un ambiente de dos o tres dimensiones. En general, a pesar de que se ha trabajado mucho en el desarrollo de técnicas de interacción en tres dimensiones, éstas tienden a ser difíciles de utilizar y aprender.</p> <p>Las interacciones también se pueden ver muy afectadas por la pantalla y el dispositivo utilizado para las tareas de VA. Si bien puede haber una amplia gama de configuraciones que soportan VA, las técnicas de interacción deben ser diseñadas de forma tal que sean similares independientemente del dispositivo y la pantalla utilizados.</p>

Tabla 3.14: Niveles de Interacción en VA de Thomas y Cook [TC05].

En su trabajo, Thomas y Cook recomiendan la creación de una ciencia de las interacciones para soportar VA, destacando la necesidad de caracterizar el espacio de diseño y de identificar áreas no exploradas relevantes para VA, tales como las interacciones para el diálogo del usuario con la información, necesarias para el razonamiento analítico.

En 2012, Heer and Shneiderman [HS12] proponen una clasificación de interacciones en el contexto de VA en función de los objetivos del usuario. Este trabajo es extendido ese mismo año por Kerren y Schreiber [KS12], que modifican levemente la clasificación de Heer y Shneiderman presentando la taxonomía descrita en la tabla 3.15. En su taxonomía, Kerren y Schreiber distinguen tres categorías. La categoría *A* se enfoca en el espacio de los datos y en cómo los datos son representados visualmente. La categoría *B* corresponde a las interacciones que se aplican sobre las representaciones visuales. Finalmente, la categoría *C* cubre aquellas interacciones que tienen lugar en el proceso general de exploración de los datos (colaboración o modificación de los historiales de interacción, por ejemplo).

Taxonomía de las Interacciones para VA de Kerren y Schreiber	
Categoría A: Especificación de los Datos y la Vista	
Se corresponde con las transformaciones de los datos y el mapeo visual del Modelo de Referencia de Card <i>et al.</i> [CMS99].	
Codificar/visualizar	Estas interacciones permiten al usuario elegir la representación visual de los datos incluyendo las características gráficas, tales como el color, la forma, etc. La representación visual depende del tipo de los datos a representar, por ejemplo, datos relacionales pueden representarse con árboles nodo-enlace, o <i>treemaps</i> .

Taxonomía de las Interacciones para VA de Kerren y Schreiber

Reconfigurar	Algunas interacciones permiten al usuario mapear atributos específicos a entidades gráficas. Por ejemplo, es posible mapear diferentes atributos de un conjunto de datos multi-dimensional a diferentes ejes en un <i>scatterplot</i> .
Filtrar	Esta interacción es de vital importancia en VA ya que permite al usuario reducir interactivamente el conjunto de datos mostrado en la vista.
Ordenar	Esta interacción permite al usuario ordenar los registros de acuerdo a los valores de sus atributos. Es una operación fundamental en el proceso de VA.
Derivar	Estas interacciones permiten al usuario integrar los resultados de cómputos estadísticos (agregación, medianas, etc.) en el conjunto de datos a ser visualizado.
Ajustar	Este tipo de interacciones está estrechamente relacionada con las de derivación y permiten la modificación de parámetros para el análisis automático (incluyendo simulación). De este modo, es posible llevar adelante acciones en el espacio de los datos utilizando métodos computacionales (configurables) para complementar así los métodos visuales.

Categoría B: Manipulación de la Vista

Se corresponde con las transformaciones de la vista del Modelo de Referencia de Card *et al.* [CMS99].

Seleccionar	Este tipo de interacciones permite seleccionar un objeto individual o un conjunto de objetos con el objetivo de resaltarlos, manipularlos o filtrarlos.
Navegar/explorar	Este tipo de interacciones modifican el nivel de detalle de los datos visualizados.
Coordinar/conectar	Este tipo de interacciones permite conectar un conjunto de vistas para permitir al usuario descubrir ítems relacionados. Técnicas de interacción como <i>brushing</i> y <i>linking</i> son utilizadas en casi todos los sistemas de VA.
Organizar	La mayoría de los grandes sistemas de VA consisten en varias ventanas y espacios de trabajo que deben ser organizados en una pantalla. Agregar y eliminar vistas puede ser una tarea confusa para el analista. Algunos sistemas ayudan al usuario asignando espacios específicos o agrupando las vistas para lograr preservar el contexto y su mapeo mental.

Taxonomía de las Interacciones para VA de Kerren y Schreiber	
Categoría C: Proceso y Procedencia	
Para facilitar el proceso analítico.	
Registrar	Este tipo de interacciones permite guardar y visualizar el historial de interacciones llevado a cabo por el usuario, facilitándole de ese modo el proceso analítico iterativo. Rehacer o deshacer operaciones son ejemplos simples de este tipo de técnicas de interacción. Interacciones más avanzadas permiten guardar sesiones de análisis y pueden ser utilizadas para volver a distintos estados del análisis con propósitos colaborativos o de entrenamiento.
Comentar	Estas interacciones permiten realizar anotaciones gráficas o textuales posibilitando al analista el marcado de puntos, elementos o regiones dentro de la representación visual. Estas anotaciones pueden tratarse también de hipervínculos a otras vistas.
Compartir	Los sistemas de VA deben soportar discusiones y difusión de los resultados. Es importante que varios analistas interactúen en simultáneo en el mismo lugar (co-localizados) o en diferentes lugares (distribuidos). La posibilidad de compartir vistas o de publicar visualizaciones son ejemplos de requerimientos relevantes para que un sistema colaborativo sea eficiente.
Guiar	La especificación de flujos de trabajo es una tarea compleja en VA, ya que los procesos analíticos relacionados son, en general, no lineales. Un mismo resultado puede alcanzarse por varios caminos de análisis distintos. Sería muy beneficioso que un sistema de VA soportara asistencia para guiar a los analistas a través de flujos de trabajo de tareas comunes.

Tabla 3.15: Taxonomía de las Interacciones para VA de Kerren y Schreiber [KS12].

3.4. Conclusión

En este capítulo, se realizó un relevamiento exhaustivo del estado del arte correspondiente a taxonomías, caracterizaciones y clasificaciones de las interacciones en el contexto de visualización y *visual analytics*. Además, se estudiaron los trabajos disponibles en relación a la definición de taxonomías de tareas visuales.

En la literatura, existe una gran cantidad de trabajos que proponen clasificaciones de interacción que difieren en diferentes aspectos. Estas divergencias sugieren que la definición de una taxonomía exhaustiva es, sin duda, un reto. Con el objetivo de integrar todo

este trabajo previo se organizó el estado del arte en cuatro clases diferentes, distinguiendo entre las taxonomías definidas para las técnicas de interacción, de aquellas definidas para las dimensiones de las técnicas de interacción, las clasificaciones de tareas visuales y las taxonomías de interacciones orientadas a VA.

En base a este estudio exhaustivo del trabajo previo definimos una clasificación de interacciones de las técnicas de interacción en el contexto del Modelo Unificado de Visualización. Esta nueva clasificación multi-nivel fue definida con el objetivo de constituir un marco conceptual de las interacciones en visualización y es presentada en el capítulo 5 de esta tesis.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 4

Representación de los Datos

‘Without data you are just another person with an opinion.’

— W. Edwards Deming

4.1. Introducción

En este capítulo se abordará la clasificación y representación de los datos en el contexto del Modelo Unificado de Visualización (MUV). En este contexto, puede considerarse al MUV conformado conceptualmente por dos grandes etapas. En la primera etapa, que abarca desde la *Transformación de Datos* hasta los *Datos a Visualizar* inclusive, los datos se mantendrán en un formato interno al proceso que estará constituido por un conjunto de datos (a su vez compuesto por datos y atributos). En la segunda etapa del proceso, que abarca desde la *Transformación de Mapeo Visual* hasta los *Datos Visualizados*, los conjuntos de datos se verán enriquecidos con un mapeo visual.

Con el objetivo de dar soporte a la definición de las interacciones que se aplicarán en las distintas transformaciones y estados del proceso de visualización, surge la necesidad de definir una representación para los conjuntos de datos lo suficientemente flexible como para soportar las distintas clasificaciones de datos, atributos, conjuntos de datos y mapeos visuales presentes en la literatura.

En este capítulo introducimos una representación de los conjuntos de datos correspondientes a la primera etapa del *pipeline* (*dataset*) y una representación de los conjuntos de

Datos Mapeados Visualmente para la segunda etapa del *pipeline* (*mapeo visual*). Además de la definición del *dataset* se da el soporte para la representación de datos y de atributos. Además, al definir el *mapeo visual*, también se da soporte para el sustrato gráfico y el sustrato espacial.

Para definir las representaciones aquí presentadas se utilizarán diagramas de clases UML. Con el objetivo de lograr un trato uniforme para todos los conjuntos de datos, datos y atributos, se definieron interfaces genéricas que luego podrán ser implementadas por los tipos concretos en particular.

4.2. Trabajo Relacionado

La clasificación de los datos es un tema muy amplio y complejo y que ha sido tratado por diversos autores en el contexto de visualización. En la literatura, se han presentado distintas clasificaciones de conjuntos de datos, datos y atributos. Entre los trabajos más relevantes en el tema destacamos los aportes de Stevens [Ste46], Bertin [Ber77] y Card *et al.* [CMS99]. Posteriormente Munzner [Mun14] presenta una clasificación más detallada, que incluye clasificaciones para los datos, los atributos y los conjuntos de datos. En el contexto de Visualización Científica, Schroeder *et al.* [SML⁺06] presentan una caracterización de datos y de conjuntos de datos para el *Visualization Toolkit* (VTK), caracterización que más tarde fue retomada por Telea [Tel14]. En lo referente al mapeo visual, los trabajos de Bertin [Ber83], Card *et al.* [CMS99] y Munzner [Mun14] son los más relevantes.

En el capítulo 2 de esta tesis se han detallado las contribuciones de los autores aquí mencionadas. En esta sección nos limitaremos a mencionar brevemente aquellos trabajos que hemos considerado relevantes para definir nuestra propia representación para los datos, los atributos, los conjuntos de datos y el mapeo visual.

La representación de datos presentada en este capítulo está basada fuertemente en los trabajos de Munzner [Mun14], Schroeder *et al.* [SML⁺06] y Telea [Tel14]. Schroeder *et al.* [SML⁺06] y Telea [Tel14] se enfocan en Visualización Científica, en la que el dominio natural de los datos suele tratarse de una función continua que debe discretizarse para ser representada. En este contexto, Schroeder *et al.* [SML⁺06] definen los conjuntos de datos como una composición de celdas y puntos. Los conjuntos de datos se denominan

habitualmente Grillas y éstas están constituidas por Celdas. La topología de las grillas está dada, en parte, por la topología de las celdas constituyentes. Las coordenadas espaciales de los vértices de las celdas correspondientes a la grilla constituyen la geometría de las celdas y, por ende, de la grilla. Existe una gran variedad de tipos de celdas. Schroeder *et al.*, en el contexto de su *framework* VTK, distinguen entre las celdas lineales y las celdas no lineales. Entre las celdas lineales, distinguen entre los vértices y polivértices (cero dimensiones); las líneas y polilíneas (una dimensión); los triángulos, las tiras de triángulos, los cuadriláteros, los *pixels* y los polígonos (dos dimensiones); los tetraedros, los hexaedros, los *voxels*, los prismas, las pirámides, los prismas pentagonales y los prismas hexagonales (tres dimensiones). Además, VTK soporta un conjunto de celdas no lineales y provee un mecanismo para transformar las celdas no lineales en celdas lineales mediante un proceso de descomposición.

Munzner [Mun14] presenta una clasificación de datos no sólo en el contexto de Visualización Científica, sino también en el de Visualización de Información. La clasificación de Munzner considera, además de los datos provenientes del dominio continuo, otros tipos de datos incluyendo: los atributos, los ítems, las conexiones, las grillas y las posiciones. Munzner define los atributos como una propiedad específica que puede ser medida, observada o registrada y define al ítem como una entidad individual discreta. En nuestra propuesta de representación de los datos, a los atributos se les da un trato particular, proveyendo una clasificación especial para los mismos.

Con respecto a los tipos de los atributos se siguió, en parte, la clasificación de atributos de Munzner [Mun14], que los clasifica en dos tipos fundamentales: los atributos categóricos y los atributos ordenados. En cuanto a los atributos ordenados, Munzner distingue ordinales y cuantitativos. Los atributos ordenados pueden ser, a su vez, secuenciales, divergentes o cíclicos.

Con respecto a los conjuntos de datos, Munzner [Mun14] distingue cinco tipos: las Tablas, las Redes y Árboles, los Campos (para datos continuos), los conjuntos de datos geométricos y los conjuntos. Schroeder *et al.*, en el contexto de Visualización Científica, mencionan seis tipos de conjuntos de datos: los Conjuntos de Datos Poligonales, las Imágenes, las Grillas Rectilíneas, las Grillas Estructuradas, los Puntos No Estructurados y las Grillas No Estructuradas. En este mismo contexto, Telea [Tel14] define una clasificación de conjuntos de datos muy similar a la de Schroeder *et al.*, identificando la

Grilla Uniforme (se corresponde con el tipo de dato Imagen de Schroeder *et al.*), la Grilla Rectilínea, la Grilla Estructurada y la Grilla No Estructurada.

En lo referente a los *Datos Mapeados Visualmente*, su representación estará integrada por el conjunto de *Datos a Visualizar* enriquecido con el mapeo visual. Muchos autores han estudiado el mapeo visual pero, hasta el momento, no se ha publicado una representación explícita. Como se explicó en la sección 2.3.1.4 del capítulo 2, la estructura visual está constituida por el sustrato espacial y el sustrato gráfico.

El *sustrato espacial* está compuesto por los ejes y sus propiedades. Según Card *et al.* [CMS99] los ejes pueden ser lineales o radiales y pueden ser clasificados en cuatro tipos: no estructurados (la organización espacial no presenta ejes), nominales, ordinales y cuantitativos. Además, los ejes conllevan asociada una organización espacial considerando 5 alternativas: composición (ubicación ortogonal de los ejes), alineación (repetición de un eje en distintas posiciones del espacio), plegado (continuación de un eje en una dimensión ortogonal), recursión (subdivisión repetida del espacio) y sobrecarga (reuso del mismo espacio, por el mismo conjunto de datos). Con respecto a la orientación de los ejes, Munzner [Mun14] plantea tres posibilidades: orientación rectilínea, paralela y radial.

El *sustrato gráfico* está compuesto por los elementos (y sus propiedades gráficas) que se utilizarán para representar los datos en la vista a generar. La información presente en el sustrato gráfico se puede descomponer en marcas (elementos visuales) y canales (atributos de esos elementos visuales). Las marcas son primitivas geométricas que pueden clasificarse según el número de dimensiones que requieren. Card *et al.* [CMS99] identifican 4 tipos elementales de marcas (puntos, líneas, áreas y volúmenes). Por otro lado, Munzner [Mun14] clasifica las marcas en dos tipos: marcas de contenido (aquéllas que representan un ítem) y marcas de conexión (aquéllas que representan un enlace). Además de las marcas identificadas por Card *et al.* y Munzner, consideramos que también debe tenerse en cuenta los Glifos (son marcas n-dimensionales), y marcas especiales como áreas curvas, áreas compuestas, polilíneas, curvas, formas, etc. Los canales visuales son las propiedades de los elementos visuales que se pueden utilizar para codificar información adicional. Entre los autores que presentan clasificaciones de los canales visuales, distinguimos a Bertin [Ber83], MacEachren [Mac95], Ward *et al.* [WY04] y Munzner [Mun14] como los más relevantes.

Para una descripción más detallada de los aportes de los autores aquí mencionados en

materia de clasificación de datos, conjuntos de datos, atributos y mapeo visual, referirse a las secciones 2.3.1.2 y 2.3.1.4 del capítulo 2 de esta tesis.

Puede observarse que si bien en la literatura existen diversos trabajos que tratan la clasificación y representación de datos, conjuntos de datos y atributos, ninguna de las clasificaciones presentadas hasta el momento es completamente abarcativa. Los trabajos de Schroeder *et al.* y Telea están enfocados únicamente en Visualización Científica. Si bien el trabajo de Munzner es más genérico, carece de una definición detallada de los conjuntos de datos y de los datos necesarios en el campo de Visualización Científica. Además, aunque sí considera marcas de una, dos y tres dimensiones, Munzner no incorpora los glifos. En cuanto a éstos, cabe mencionar que su característica n-dimensional generaliza las otras marcas y también plantea nuevos desafíos de diseño. En lo que respecta a los *Datos Mapeados Visualmente*, si bien es un tema ampliamente tratado en la literatura disponible, en nuestro conocimiento aún no existe una representación formal de los mismos.

De lo expuesto previamente, surge la necesidad de definir una clasificación y representación de datos, atributos, conjuntos de datos y mapeo visual para el MUV en las que se consideren integralmente los atributos, los datos, los conjuntos de datos y las características del mapeo visual.

4.3. Clasificación de Los Datos

Para obtener una representación genérica de los datos involucrados en el proceso de visualización nos basaremos en las clasificaciones presentadas por Munzner [Mun14], Schroeder *et al.* [SML⁺06] y Telea [Tel14]. Si bien la clasificación de Munzner es la que nosotros consideramos más completa, es preciso incorporar a la misma los distintos tipos de celda presentados por Schroeder *et al.* [SML⁺06] y Telea [Tel14].

Para esto, definiremos un tipo abstracto *Dato*. Los tipos específicos para cada una de las clases de dato y la operaciones básicas necesarias para su manipulación. En la figura 4.1 se ilustra, mediante el uso de un diagrama de clases, la clasificación de datos propuesta.

Un *Dato* puede ser un *Ítem*, una *Conexión*, una *Celda* o una *Posición*.

Un *ítem* es definido por Munzner como una entidad individual discreta. Es posible

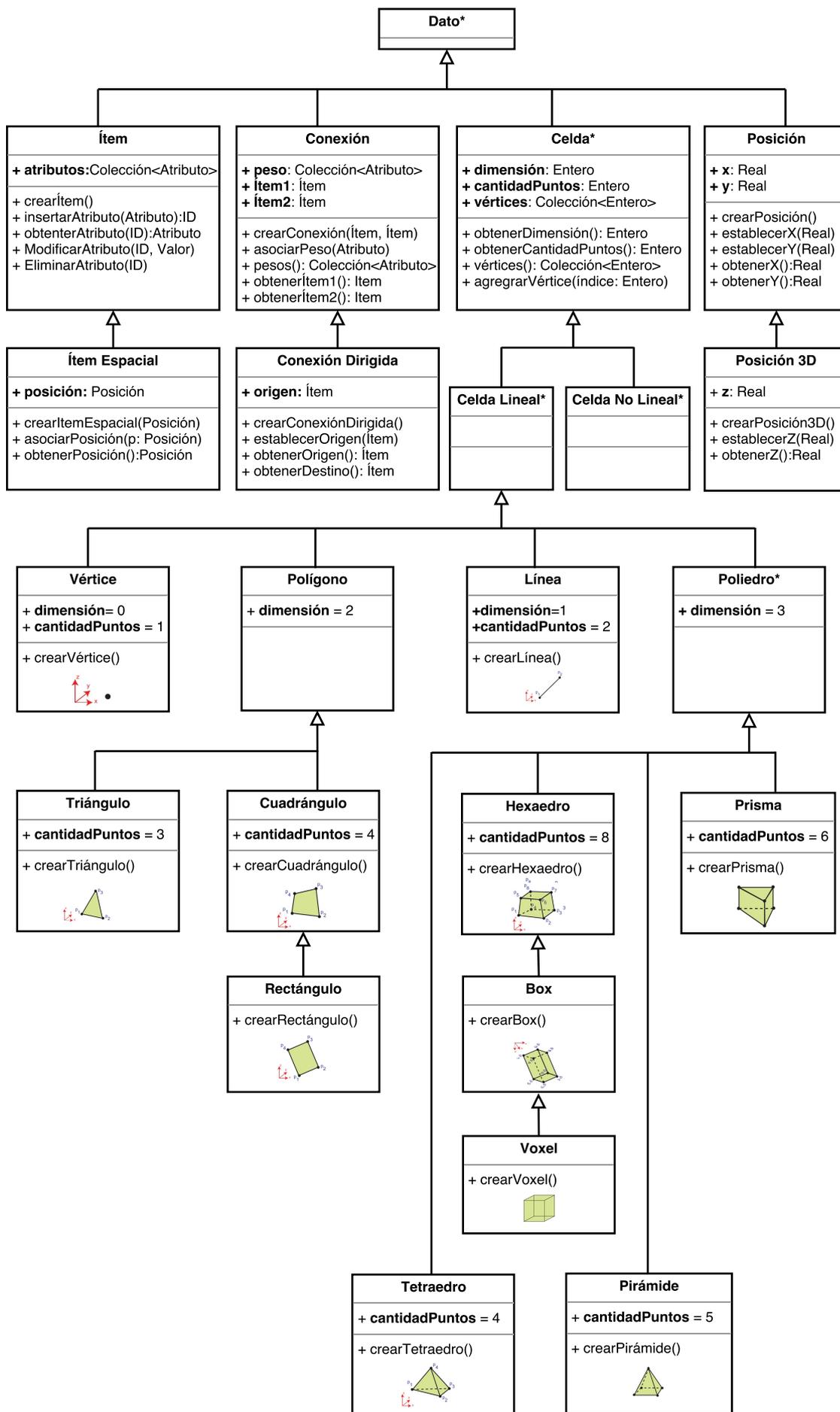


Figura 4.1: Clasificación de los Datos en el MUV.

representar un ítem como una colección de atributos, que podrán ser de distintos tipos, y un *ítem espacial* como un ítem convencional que tiene asociada una posición. Esto último es necesario porque, en la clasificación presentada, una posición no es un atributo; es decir que no es posible asociar una posición a un ítem de dato.

Un *Dato* de tipo *Conexión* representa una relación entre dos ítems. Una conexión, además de representar los ítems que conecta, puede tener asociado un conjunto de atributos. Una *Conexión Dirigida* es una conexión convencional donde se explicita una dirección, siendo uno de los ítems conectados el origen y el otro ítem el destino.

Una *Posición* representa un punto en un espacio de dos dimensiones. Una *Posición 3D*, en cambio, representa un punto en un espacio de tres dimensiones.

Una *Celda* considera medidas o cálculos provenientes de un dominio continuo. Los datos continuos requieren un tratamiento especial, que involucra la aplicación de una técnica de muestreo. Dos operaciones relacionan los datos muestreados y los datos continuos:

- Muestreo: Proceso que, dado un conjunto de datos continuo, permite producir datos muestreados a partir del mismo.
- Reconstrucción: Proceso que, dado un conjunto de datos muestreado, permite reconstruir una aproximación de los datos continuos originales.

Los conjuntos de datos obtenidos en este contexto se denominan *Grillas* y suelen componerse de un conjunto de celdas definidas en base a los puntos de muestreo.

Una *Celda* se compone de una colección de puntos muestreados, para cada uno de los cuales se debe almacenar la posición espacial del punto y los valores muestreados en ese punto. En cada punto, y para un determinado instante de tiempo, puede muestrearse un único valor para cada uno de los atributos asociados. Los valores muestreados pueden representarse como una tabla de ítems espaciales, donde cada ítem espacial tiene asociado un índice. La tabla de ítems espaciales se almacenará en la Grilla, y una Celda, será una lista ordenada de índices de ítems espaciales (que denominamos *vértices*). Así, cada punto muestreado se representará una única vez, y las celdas que compartan vértices, simplemente almacenarán el índice de ese vértice repetido.

En general, las celdas tendrán asociadas una dimensión y una topología. Esta topología está dada por un conjunto de puntos de muestra y la conexión entre los mismos. Una celda puede ser lineal o no lineal. En la presente clasificación de datos se incluyeron los

siguientes tipos de celda lineales:

- Celdas de cero dimensiones: Vértice.
- Celdas de una dimensión: Línea.
- Celdas de dos dimensiones: Triángulo, Cuadrángulo, Rectángulo y Polígono.
- Celdas de tres dimensiones: Tetraedro, Hexaedro, Box, *Voxel*, Pirámide, Prisma y Poliedro.

Las celdas no lineales presentan la misma topología que las celdas lineales, pero los vértices se unen mediante curvas expresadas en distintas bases (polinomiales, *splines*, etc.). Las celdas no lineales pueden proveer funciones de interpolación más precisas que aquéllas utilizadas en los procesos de muestreo que generan celdas lineales. Además, son más eficientes a la hora de aproximar geometrías curvas. Es posible transformar las celdas no lineales en celdas lineales mediante un proceso de descomposición. *The Visualization Toolkit* (VTK) [SLM04], por ejemplo, utiliza funciones de interpolación cuadrática subdividiendo las aristas de las celdas con vértices intermedios obteniendo así celdas lineales. Si bien en nuestra clasificación se han considerado las celdas no lineales, no se ha profundizado en su definición.

En la tabla 4.1 se ilustra la relación entre los tipos de datos definidos en esta tesis con los presentes en la bibliografía. Observar que nuestra clasificación de datos tuvo en cuenta todos los tipos de datos definidos por los autores, a excepción de las celdas polivértice, polilínea y tira de triángulos de Schroeder *et al.* [SML⁺06]. Las razón por la cual no fueron tenidas en cuenta es porque conceptualmente, a nuestro criterio, los polivértices, las polilíneas y las tiras de triángulos pueden ser tratados como un conjunto de celdas (vértices, líneas y triángulos, respectivamente). De modo que estas estructuras constituyen un conjunto de datos (una Grilla) y no un dato en particular, y serán tenidas en cuenta en nuestra clasificación de conjuntos de los datos (ver sección 4.5).

4.4. Clasificación de los Atributos

Algunos tipos de datos contienen atributos, como los ítems, los ítems espaciales y las conexiones. Un atributo se define como una propiedad específica, que puede ser medida,

observada o registrada. Los atributos pueden ser caracterizados por su dimensión y su semántica, dando lugar a una clasificación de tipos de atributos.

Al definir la clasificación de atributos para el MUV presentada en esta tesis, se siguió la clasificación de atributos presentada por Munzner [Mun14] (detallada en la sección 2.3.1.2 del capítulo 2), incorporando a la misma atributos de dimensión mayor a uno como vectores y tensores.

En la figura 4.2 se muestra la clasificación de los atributos en el contexto del MUV. En la cual, en función de su dimensión, clasificamos los atributos en:

Escalares

Los atributos escalares son atributos de una dimensión. Los atributos escalares pueden ser de dos tipos: categóricos u ordenados.

Los atributos **categóricos** no tienen un orden implícito, pero pueden presentar una estructura jerárquica. Podría imponerse un orden arbitrario externo; sin embargo, este orden no es implícito al atributo en sí mismo. Sobre las constantes de este tipo de atributo se definen operaciones de igualdad y desigualdad.

Los atributos **ordenados** sí tienen un orden implícito. Este tipo puede a su vez subdividirse en atributos **ordinales** y atributos **cuantitativos**. Sobre las constantes del tipo de atributo *ordinal* existe una relación de orden pero no pueden aplicarse operaciones aritméticas. Los atributos *cuantitativos* son medidas de magnitud y sobre éstas sí pueden aplicarse operaciones aritméticas. A su vez, los atributos *ordenados* pueden ser **secuenciales** o **divergentes**. Los atributos *ordenados* son *secuenciales* cuando se definen en un rango homogéneo que va de un valor mínimo a un valor máximo. Los atributos *ordinales* son divergentes cuando su rango de valores está caracterizado por dos secuencias que tienen un origen común a partir del cual se define una subsecuencia ascendente y una descendente. Adicionalmente, los atributos *ordenados* pueden ser **cíclicos** cuando los valores que toman parten de un valor inicial y retornan al mismo valor en vez de crecer infinitamente de manera secuencial o divergente.

Vectores

Los atributos vectoriales son atributos de dos o tres dimensiones. Este tipo de atributos puede codificar, por ejemplo, dirección, fuerza o gradientes de funciones

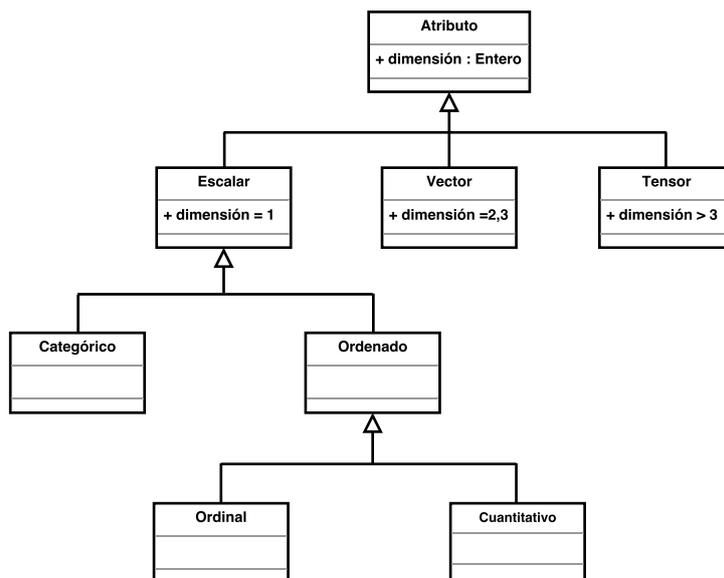


Figura 4.2: Clasificación de los Atributos en el MUV.

escalares.

Tensores

Los atributos tensoriales son generalizaciones de alta dimensionalidad de los vectores y matrices. Los tensores se caracterizan por su orden, un tensor de orden 0 es un escalar, un tensor de orden 1 es un vector, un tensor de orden 3 es una matriz, y un tensor de orden k ($k > 3$) puede ser considerado una tabla de dimensión k . Muchos fenómenos físicos se representan matemáticamente como tensores; ejemplos comunes son los tensores de fuerza y de tensión, que representan respectivamente la fuerza y la tensión en un punto de un objeto bajo carga.

4.5. Clasificación de los Conjuntos de Datos

En la figura 4.3 se ilustra nuestra propuesta de clasificación de los conjuntos de datos en el contexto del MUV. Esta clasificación integra los conjuntos de datos considerados por Munzner [Mun14], Telea [Tel14] y Schroeder *et al.* [SML⁺06], entre otros. Nuestro objetivo fue el de lograr una clasificación que integrase todos los conjuntos de datos presentes en las distintas áreas de visualización.

Denominaremos **Dataset** al conjunto de datos más general, que representa una colección de datos. A alto nivel, los *datasets* se clasifican en dos tipos, los conjuntos de datos

espaciales y los conjuntos de datos no espaciales. Los conjuntos de datos espaciales tienen asociadas posiciones espaciales. Como conjuntos de datos no espaciales distinguimos las *Tablas*, las *Tablas Multidimensionales*, los *Conjuntos*, las *Listas*, los *Grafos* y los *Árboles*. Como conjuntos de datos espaciales distinguimos las *Nubes de Puntos no Estructurados*, la *Geometría* y las *Grillas*. A su vez, una *Grilla* puede ser *uniforme* o *no estructurada*. Y una *Grilla Uniforme* puede ser *rectilínea* o *estructurada*.

A continuación se describe cada uno de los tipos de conjuntos de datos incluidos en el modelo junto con las operaciones básicas más generales asociadas a los mismos.

DATASET

Representa el conjunto de datos más general, compuesto por una colección de datos. Como se detalló en la sección 4.3, estos datos pueden ser de distintos tipos (ítems, posiciones, conexiones, etc.). A partir de la colección y en función del tipo del dato y la estructura del conjunto de datos a representar se definirán tipos de conjuntos de datos más específicos. Las operaciones básicas que deben estar disponibles en un *dataset* son las siguientes:

- **insertarDato(Dato):id** Incorpora un nuevo dato a la colección de datos, retornando un identificador único para ese dato.
- **eliminarDato(id):Dato** Elimina del conjunto de datos el dato con identificador *id*, retornando el dato eliminado.
- **obtenerDato(id):Dato** Esta operación de consulta retorna el dato con identificador *id*.
- **obtenerId(Dato):id** Esta operación de consulta permite obtener el identificador único asociado a un dato.
- **estaDato(id):Booleano** Esta operación de consulta decide si el dato con identificador *id* se encuentra almacenado en el *dataset*.
- **datos():Iterable<Dato>** Esta consulta retorna una colección iterable de los datos contenidos en el *dataset*.
- **agruparDatos(Colección<Dato>)** Esta operación agrupa la colección de datos especificados.

- **desagruparDatos(Colección<Dato>)** Esta operación desagrupa los datos contenidos en la colección especificada. Requiere que los datos contenidos en la colección hayan sido previamente agrupados.
- **clusters():Iterable<Dataset>** Esta consulta retorna una colección iterable de los clusters contenidos en el *dataset*.
- **eliminarDataset()** Esta operación elimina el *dataset*.

Además de las operaciones básicas de recorrido, creación y eliminación, incorporamos al *dataset* operaciones para agrupar y desagrupar datos con el objetivo de soportar operaciones de clusterización y almacenamiento de estructuras multi-resolución.

CONJUNTO DE DATOS NO ESPACIALES

Son aquellos *dataset* cuyos datos no tienen asociados una posición espacial.

Tabla

Una *tabla* es un *dataset* que almacena ítems. Una *tabla* está compuesta por filas y columnas, donde cada fila representa un ítem de dato y cada columna representa un atributo. Cada celda de la tabla está especificada por la combinación de una fila y una columna (es decir, un ítem y un atributo) y contendrá el valor asociado a ese par. Una *tabla* puede representarse genéricamente como un arreglo de ítems. Las operaciones básicas que deben estar disponibles en una *tabla* son las siguientes:

- **eliminarColumna(Atributo)** Esta operación elimina de la *tabla* el atributo especificado. Es decir, reduce en uno la dimensionalidad de los datos almacenados en la *tabla*, eliminando el *atributo* especificado de cada ítem de datos.
- **insertarColumna(Atributo)** Esta operación agrega a la *tabla* una columna que se asociará al *atributo* especificado. Incrementa la dimensionalidad del conjunto de datos almacenados en la *tabla* en uno.
- **modificarValorEnCelda(Item, Atributo, Valor)** Esta operación modifica el valor asociado a la celda identificada por el *ítem* y el *atributo* especificados.

- **obtenerValorEnCelda(Item, Atributo):Valor** Esta operación de consulta permite obtener el valor asociado a la celda identificada por el *ítem* y el *atributo* especificados.
- **atributos():Iterable<Atributo>** Esta consulta retorna una colección iterable de los atributos contenidos en la *tabla*.

Tabla Multidimensional

Una *tabla multidimensional* es una *tabla* con dimensión mayor a dos. Se trata de una estructura más compleja que permite indexar una celda con múltiples claves. Una *tabla multidimensional* puede representarse genéricamente como un arreglo de dos dimensiones de ítems.

Conjunto

Un *conjunto* es un *dataset* que representa una colección no ordenada de ítems. Las operaciones básicas de un *conjunto* son las mismas que las de un *dataset*.

Lista

Una *lista* es un *conjunto* ordenado. Las operaciones básicas de una *lista* son las mismas que las de un *conjunto*, que se implementarán acorde a la estructura representada, manteniendo la colección ordenada y haciendo uso eficiente de la misma.

Grafos

Los *grafos* representan relaciones entre dos o más ítems. Un ítem en una red se representa con un nodo y una conexión entre dos ítems mediante una arista. Un *grafo* puede representarse genéricamente como una colección de *ítems* y *conexiones*. Las operaciones básicas que deben estar disponibles en un *grafo* son las siguientes:

- **insertarEnlace(Conexión)** Esta operación inserta una nueva conexión en el *grafo*.
- **eliminarEnlace(Conexión)** Esta operación elimina la *conexión* especificada del *grafo*.
- **adyacentes(Ítem): Iterable<Conexión>** Esta consulta retorna una colección iterable de las conexiones asociadas al *ítem* especificado.
- **opuestos (Ítem, Ítem): Booleano** Esta operación de consulta decide

si los *ítems* especificados se encuentran conectados en el *grafo*. Es decir, si existe un enlace en el *grafo* que los conecte directamente.

Árbol

Un *árbol* es un tipo especial de grafo que mantiene una estructura jerárquica. Las operaciones básicas que deben estar disponibles en un *árbol* son las siguientes:

- **raíz():Ítem** Esta operación retorna el *ítem* correspondiente a la raíz del *árbol*.
- **padre(Ítem):Ítem** Esta consulta retorna el padre del *ítem* especificado en la estructura jerárquica representada por el *árbol*.
- **hijos(Ítem):Iterable<Ítem>** Esta consulta retorna una colección iterable de los hijos (*ítems*) del *ítem* especificado en la estructura jerárquica representada por el *árbol*.

CONJUNTO DE DATOS ESPACIALES

Se trata de aquellos *datasets* que tienen asociadas una o más posiciones espaciales.

Nube de Puntos No Estructurados

Este conjunto está compuesto de puntos ubicados en el espacio. No tienen una topología asociada y la geometría es completamente no estructurada. Una *nube de puntos no estructurados* puede representarse con una colección de *ítems espaciales*.

Geometría

Los conjuntos de datos geométricos son intrínsecamente espaciales. Este tipo de conjunto de datos especifica información sobre la forma de los *ítems* con posiciones espaciales específicas. Los *ítems* pueden ser puntos, líneas, curvas, superficies, regiones o volúmenes. Se utilizan en el contexto de tareas que requieran entendimiento de formas. Este tipo de conjunto de datos no tiene necesariamente atributos. Para representar una geometría exclusivamente espacial, donde no intervienen atributos, este tipo de conjunto de datos puede representarse como una colección de datos de tipo *posición*. Si se desea almacenar atributos junto con la geometría, este tipo de conjunto de datos puede

representarse como una colección de datos de tipo *ítem espacial*. En ambos casos, debe estar implícito el orden entre las posiciones o los ítems espaciales.

Grilla

Una Grilla se compone básicamente de una colección de *celdas* C_i cuyos vértices son los puntos de muestreo p_i y además contiene valores de atributos asociados a éstas. Cada celda contiene medidas o resultados provenientes de un dominio continuo. Conceptualmente es posible tomar infinita cantidad de medidas, de modo que siempre se puede tomar una nueva medida entre dos existentes. Ejemplos de fenómenos continuos que pueden ser medidos en el mundo físico o simulados por *software* podrían ser, por ejemplo, temperatura, presión, velocidad, fuerza, densidad, etc. Dado un conjunto de datos muestreados en los puntos p_i , las celdas se utilizan para definir el soporte a las funciones que interpolan los datos entre los puntos muestreados. Así, la forma en que los puntos de muestra se conectan para formar celdas depende del dominio que se desea muestrear, así como también del tipo de las funciones de la base que se desean utilizar para llevar a cabo la reconstrucción. En nuestra propuesta, representamos una *grilla* como una colección de *celdas* y un arreglo de *ítems espaciales*. Cada ítem espacial representará un punto muestreado, almacenando la posición espacial del punto, y los valores muestreados en sus atributos. Las celdas, tal como se definió en la sección 4.3, representarán sus vértices como índices al arreglo de ítems espaciales almacenado en la Grilla. De este modo, cada punto muestreado se almacena una única vez.

Grilla Uniforme

Cuando el muestreo se realiza a intervalos completamente regulares las celdas forman una Grilla Uniforme. En este caso no es necesario explicitar la geometría de la *grilla* (en términos de posiciones espaciales) ni la topología de la misma (en términos de cómo se conectan las celdas con sus celdas vecinas). Se trata de una colección de puntos y celdas organizadas en una grilla rectangular y regular. Tanto la topología como la geometría es regular. En lo que se refiere a la geometría, basta con explicitar el punto de origen de la Grilla, la dimensión de la misma y la separación entre los puntos muestreados en cada dimensión.

Grilla Rectilínea

Las Grillas Rectilíneas soportan muestreo no uniforme, permitiendo un almacenamiento eficiente de los datos que tiene más complejidad en ciertas áreas y menos complejidad en otras. Las Grillas Rectilíneas mantienen la definición implícita de celdas utilizada por las Grillas Uniformes, pero relajan la restricción de muestreo equidistante para cada eje. Para este tipo de grillas, es preciso explicitar las distintas separaciones de muestreo para cada eje, las cuales es posible representar con una lista de separaciones para cada eje. La dimensión de estas listas dependerá de la cantidad de puntos de muestreo efectuados sobre cada eje.

Grilla Estructurada

Estas grillas permiten almacenar formas curvilíneas, donde la locación geométrica de cada celda debe ser especificada. Para esto se deben especificar las coordenadas x , y , z de todos los puntos de muestreo y la función utilizada para interpolarlos. Al mismo tiempo, estas grillas preservan el ordenamiento de matriz de los puntos de muestra, lo que permite la construcción de celdas implícitas a partir del ordenamiento de puntos. Intuitivamente, se puede ver como una deformación libre de las grillas uniformes, donde el orden topológico de los puntos se mantiene, pero su posición geométrica puede variar libremente.

Grilla No Estructurada

Las Grillas No Estructuradas son el tipo de grilla más general y flexible, y permiten representar formas que no pueden ser modeladas con Grillas Estructuradas. Dados los puntos de muestreo, las celdas se definen de manera explícita. Esto implica que la información topológica sobre cómo se conectan las celdas entre sí debe ser especificada junto con sus posiciones espaciales.

Una Grilla No Estructurada puede modelarse como un arreglo de ítems espaciales (los cuales representan los puntos de muestreo) y una colección de *celdas*, tal como se define la *grilla* general en nuestra representación. Este modelo nos permite definir cada celda de forma separada e independiente de otras celdas. Observar que si bien es posible extraer las relaciones de adyacencia entre las celdas, éste sería un proceso muy costoso con la representación expuesta. En una celda poligonal, por ejemplo, se pueden obtener relaciones entre sus caras,

vértices y lados. De modo que puede considerarse extender el modelo propuesto, incorporando una representación para considerar distintas alternativas de relaciones de adyacencia de las celdas (como un grafo de celdas, por ejemplo). Si bien es preferible utilizar grillas no estructuradas homogéneas (es decir, compuestas por celdas del mismo tipo) celdas de diferente tipo e incluso diferente dimensionalidad podrían integrarse en la misma grilla, dando lugar a las grillas no estructuradas heterogéneas.

En la tabla 4.2 se ilustra la relación entre los tipos de conjuntos de datos definidos en esta tesis con los presentes en la bibliografía. Observar que nuestra clasificación de datos integra todos los tipos de conjuntos de datos definidos por distintos autores.

4.6. Representación de los Datos Mapeados Visualmente

En esta sección introducimos una representación para los *Datos Mapeados Visualmente*. Los *Datos Mapeados Visualmente* comprenden los *Datos a Visualizar* enriquecidos con información necesaria para su representación en pantalla. Los datos contenidos en el conjunto de *Datos Mapeados Visualmente* son directamente visualizables aplicando una técnica que los soporte. La estructura de datos definida para representar los *Datos Mapeados Visualmente* debe contemplar tanto el almacenamiento del *dataset* a representar, como el de la estructura visual que será utilizada para representar esos datos en pantalla.

Como se mencionó anteriormente, la estructura visual está compuesta por el sustrato espacial y el sustrato gráfico. El sustrato espacial refleja la organización del espacio que utilizará la técnica de visualización y contiene información sobre las características geométricas y topológicas de la representación a generar. El sustrato espacial está integrado por los ejes y sus propiedades. Teniendo fuertemente en cuenta los aportes de Card *et al.* [CMS99] y Munzner [Mun14] definimos entonces la representación para el sustrato espacial. Con respecto a la orientación de los ejes, a las alternativas presentadas por Munzner [Mun14] (orientación rectilínea, paralela y radial) incorporamos la denominada orientación libre, en la cuál, la orientación de los ejes no se restringe a ninguna de las otras tres alternativas.

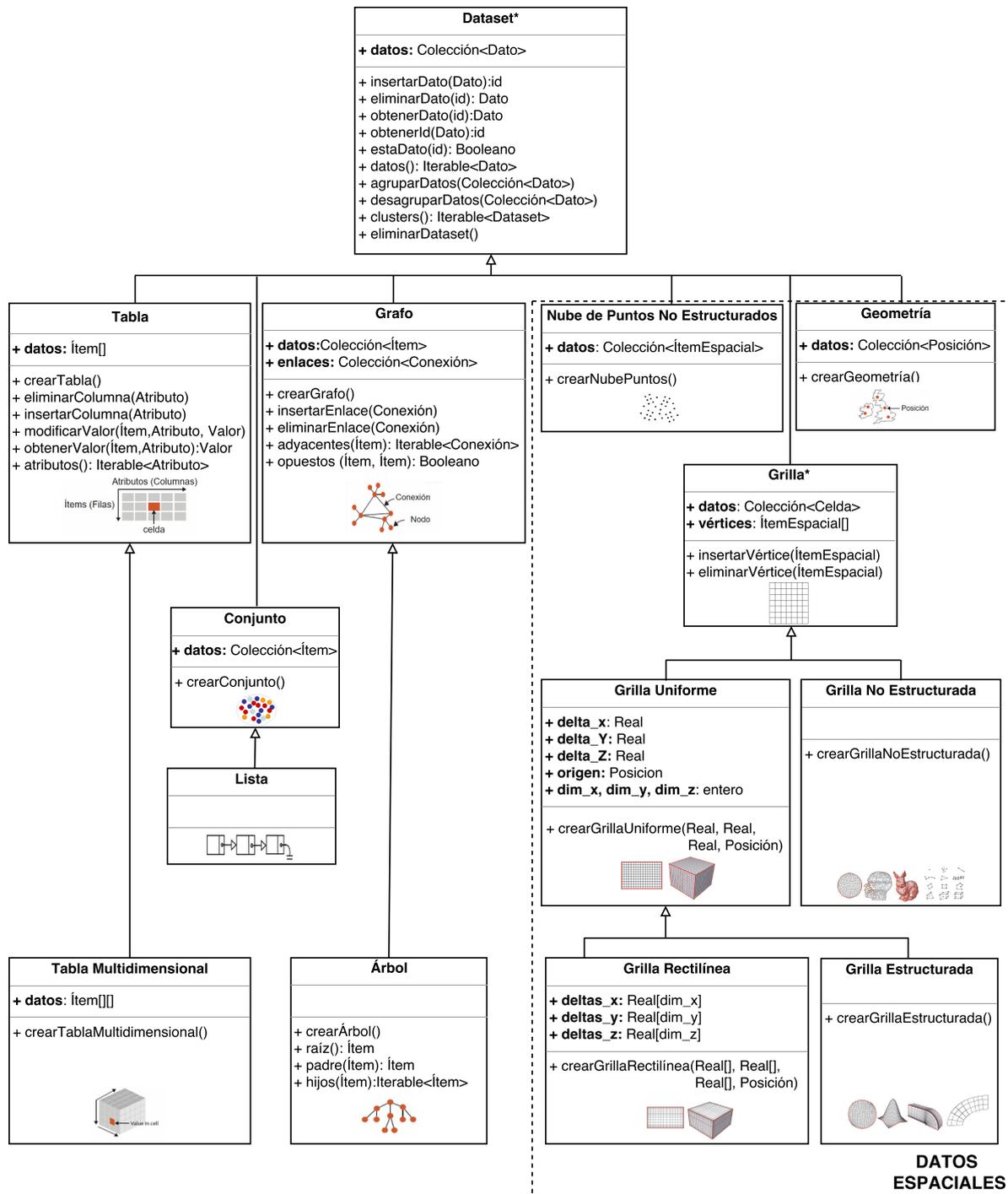


Figura 4.3: Clasificación de los Conjuntos de Datos en el MUV.

Telea [Tel14]	Grilla No Estructurada									✓			✓
	Grilla Estructurada									✓			✓
	Grilla Rectilínea									✓		✓	
	Grilla Uniforme									✓	✓		
Schroeder <i>et al.</i> [SML ⁺ 06]	Grilla No Estructurada									✓			✓
	Puntos No Estructurados							✓		✓			
	Grilla Estructurada									✓			✓
	Grilla Rectilínea									✓		✓	
	Imagen									✓	✓		
	Datos Poligonales									✓			
Munzner [Mun14]	Conjuntos			✓	✓								
	Geometría								✓				
	Campo									✓	✓	✓	✓
	Árbol							✓					
	Red						✓						
	Tabla	✓	✓										
	Tabla												
	Tabla Multidimensional												
	Conjuntos												
	Lista												
	Grafo												
	Árbol												
	Nube de Puntos No Estructurada												
	Geometría												
	Grilla												
	Grilla Uniforme												
	Grilla Rectilínea												
	Grilla Estructurada												
	Grilla No Estructurada												

Tabla 4.2: Tipos de Conjuntos de Datos y su relación con las Clasificaciones Presentadas por Munzner [Mun14], Schroeder *et al.* [SML⁺06] y Telea [Tel14]

Sustrato Espacial*	Eje*
+ cantidad_ejes :Entero + orientación_ejes :{RECTILÍNEA, PARALELA, RADIAL, LIBRE} + organización_ejes : {COMPOSICIÓN, ALINEACIÓN, PLEGADO, RECURSIÓN, SOBRECARGA} + ejes : Colección<Eje>	+ tipo : Escalar + rango : par<Escalar, Escalar>
+ crearSustratoEspacial() + configurarCantidadEjes(Entero) + configurarOrientaciónEjes(Orientación) + configurarOrganizaciónEjes(Organización) + insertarEje(Eje) + cantidadEjes(): Entero + orientaciónEjes():Orientación + organizaciónEjes(): Organización + ejes(): Colección<Eje>	+ crearEje() + configurarTipo(Escalar) + configurarRango (Escalar, Escalar) + tipo(): Escalar + rango(): Par<Escalar, Escalar>

Figura 4.4: Representación del Sustrato Espacial en el MUV.

Esto se muestra en la figura 4.4. En la misma, un *Eje* tiene asociado un tipo y un rango de valores, y el *Sustrato Espacial* se representa por la cantidad de ejes involucrados, su orientación (rectilínea, paralela, radial o libre), su organización en pantalla (por composición, alineación, plegado, recursión o sobrecarga) y la colección de ejes asociados. Tanto para el sustrato espacial como para un eje en particular se deben proveer las operaciones básicas de configuración y consulta para cada uno de sus atributos. Es posible representar un sustrato espacial cuya organización espacial no presenta ejes con una colección de ejes vacía.

El sustrato gráfico está compuesto por los elementos (y sus propiedades gráficas) que se utilizarán para representar los datos en la vista a generar. La información presente en el sustrato gráfico se puede descomponer en marcas (elementos visuales) y canales (atributos de esos elementos visuales).

Las marcas son primitivas geométricas que pueden clasificarse según el número de dimensiones que requieran para su construcción. En la figura 4.5 se ilustra el diagrama de clases diseñado para representar una *Marca* en el contexto de visualización. Para obtener nuestra representación nos basamos en los tipos de marcas presentados por Card *et al.* [CMS99] y Munzner [Mun14] e incorporamos los *glifos* y marcas especiales como áreas curvas, áreas compuestas, polilíneas, curvas, formas, etc. Las clases *Marca Item* y *Marca Enlace* se corresponde con las marcas de contenido y conexión de Munzner [Mun14] respectivamente.

Los canales visuales controlan la apariencia de las marcas visuales. Muchos autores presentan clasificaciones de los canales visuales, siendo Bertin [Ber83], Mac Ea-

chren [Mac95], Ward *et al.* y Munzner [Mun14] los más relevantes.

En nuestra representación de la *marca*, los canales visuales son representados como atributos dentro de la clase. Todas las marcas tendrán canales visuales como color, tamaño, orientación, posición, etc. El canal de posición tomará distintas formas. Para marcas de cero dimensiones, como los puntos y las formas, la posición de la marca estará dada por la posición del centro de la marca. Para marcas unidimensionales, como líneas, polilíneas, conexiones y curvas, la posición estará dada por las posiciones de los vértices asociados a la línea. Una marca de tipo curva, tendrá a su vez asociado un canal de curvatura. Canales como transparencia y luminancia estarán representados por el canal de color.

Los datos a representar en la visualización se asociarán a una o varias marcas. Se define entonces una clase llamada *Ítem Visual*, que representará la asociación entre un *dato* del *dataset* y una *marca* en particular y almacenará para los mismos los mapeos de los atributos de los datos a los canales de la marca. Al crear un *ítem visual*, se asocia un dato a una marca. Dadas las operaciones básicas provistas por esta clase, la operación *asociarAtributo*(*idAtributo*, *Canal*) será la encargada de asociar el atributo con identificador *idAtributo* (del dato) con el *Canal* especificado de la marca.

Finalmente, la clase *Mapeo Visual* representará el mapeo visual completo, integrando el *dataset* que representa los *Datos a Visualizar*, el sustrato espacial y una colección de asociaciones entre datos e ítems visuales. Entre las operaciones básicas provistas por esta clase, se encuentra la operación *incorporarItemVisual*(*Dato*, *ItemVisual*) que será la encargada de asociar un *dato* con un *Ítem Visual*.

Si los *Datos a Visualizar* son datos espaciales, es preciso soportar el mapeo implícito de la posición del dato al canal posición de la marca. Para esto, se cuenta con el método *establecerMapeoEspacialImplícito*() provisto por *ítem visual*. Si bien para datos espaciales es común realizar este mapeo espacial implícito, es posible que el usuario desee descomponer la *posición* del dato y mapearla así a otros atributos de la marca. Para lograr esto, será necesario descomponer la Posición generando metadatos (nuevos atributos) en etapas anteriores del proceso de visualización, de modo que al momento de realizar el mapeo visual, estos metadatos sean tratados como atributos convencionales.

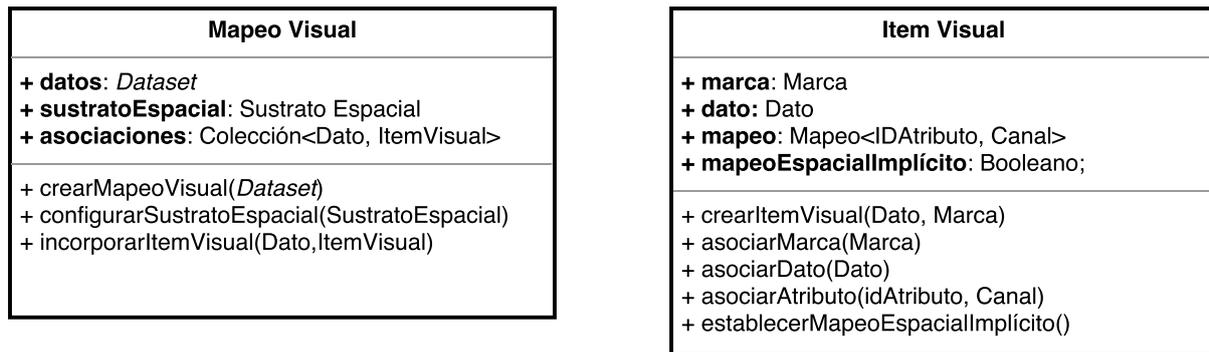


Figura 4.6: Representación del Mapeo Visual en el MUV.

4.7. Conclusión

En este capítulo presentamos una clasificación y una representación para los datos, los conjuntos de datos y los atributos. El proceso de desarrollo de esta representación involucró un profundo estudio de los principales aportes en materia de clasificación de datos, atributos, conjuntos de datos y mapeo visual y un análisis detallado de estos aportes. En las secciones 6.4 y 6.5.1 del capítulo 6 se ejemplifica la representación de los conjuntos de datos, datos, atributos y mapeo visual presentada en este capítulo con dos casos de estudio. Los casos de estudio utilizados para instanciar esta representación son dos sistemas de visualización de datos geológicos.

La representación obtenida es general y abarca las definiciones propuestas por los autores que consideramos más relevantes en el área. Se logró así una integración que permitirá sentar las bases de un proceso de visualización independiente del campo de aplicación y del área de visualización. Como parte de este proceso, tanto la clasificación como las representaciones obtenidas constituyen la base para definir las interacciones que se llevarán a cabo en cada estado y transformación del MUV.

Capítulo 5

Clasificación de Interacciones en el MUV

“I tell you and you forget. I show you and you remember. I involve you and you understand.”

— Confucius, 500 BC

En los últimos años se han presentado varias metáforas y técnicas de interacción en los distintos campos de la visualización. Estas técnicas permiten a los usuarios deformar el espacio, deformar los objetos, ver los objetos a diferentes niveles de abstracción y toda una amplia gama de otras interacciones. Las técnicas de interacción desarrolladas, si bien posibilitan las visualizaciones, han sido desarrolladas e implementadas enfocadas en la resolución de problemas particulares. Una de las mayores diferencias entre los desarrollos de visualización actuales y los previos es el desarrollo de la interactividad. Un diálogo efectivo entre el humano y la computadora enriquece la comunicación de información.

Una herramienta de visualización debe ser lo suficientemente general y flexible como para soportar las necesidades individuales, ofreciendo un esquema de interacción consistente y válido independientemente del dominio de aplicación. Esta característica habilita a los usuarios a transferir el conocimiento del uso de la herramienta entre los distintos dominios basándose en un modelo conceptual consistente.

Resulta de suma importancia, al momento de interactuar con una visualización, tener en claro en cada punto del proceso cuáles son las interacciones disponibles. El usuario

necesita saber no solo qué es lo que puede hacer, sino cómo hacerlo y cuáles serán las consecuencias de sus acciones para la representación visual y a nivel de los datos subyacentes. A pesar del cambio de representación, el usuario debe tener la seguridad de estar controlando el proceso. Esto se logra en parte con la elección de una metáfora adecuada (*representación visual + interacciones disponibles*) encuadradas en un marco conceptual del proceso que le permita saber en cada momento en qué punto del proceso se encuentra y hacia dónde puede dirigirse. Es importante asegurar:

- Predictibilidad: El usuario debe poder predecir los resultados de una interacción.
- Sintetizabilidad: Se debe poder inferir qué hizo para alcanzar el estado actual.
- Generalizabilidad: Extensión del conocimiento previo de otras interacciones a situaciones similares a las que no se enfrentó previamente.

5.1. Interactuando en el MUV

Como se explicó anteriormente, el Modelo Unificado de Visualización (MUV) [MCFE03] es un modelo de estados representado como un flujo entre los distintos estados que van asumiendo los datos a lo largo del proceso. El MUV se define en función de sus estados y las transformaciones necesarias para pasar de un estado al próximo. En la sección 2.3 del capítulo 2 se describen los estados y las transformaciones del mismo.

En cada estado del MUV hay presente un único conjunto de datos, cuya representación y semántica dependerá del estado en particular. El hecho de que cada estado de los datos esté completamente definido permite especificar (si existen) cuáles son las operaciones disponibles sobre ese estado, así como los resultados de su aplicación.

Además, como se detalló en la sección 2.4 del capítulo 2, el *pipeline* puede sufrir ramificaciones en sus transformaciones, obteniendo de este modo diferentes visualizaciones del mismo conjunto de datos. En cada transformación hay presente un único conjunto de datos de entrada (el proveniente del estado anterior) y tantos conjuntos de datos de salida como ramificaciones del *pipeline* se inicien en esa transformación. En la figura 5.1 se ilustra un ejemplo de Visualización Científica, en el cual se pueden observar distintas visualizaciones posibles de acuerdo a las elecciones hechas en cada etapa del proceso. En este ejemplo, se cuenta con datos correspondientes a una tomografía. De acuerdo a

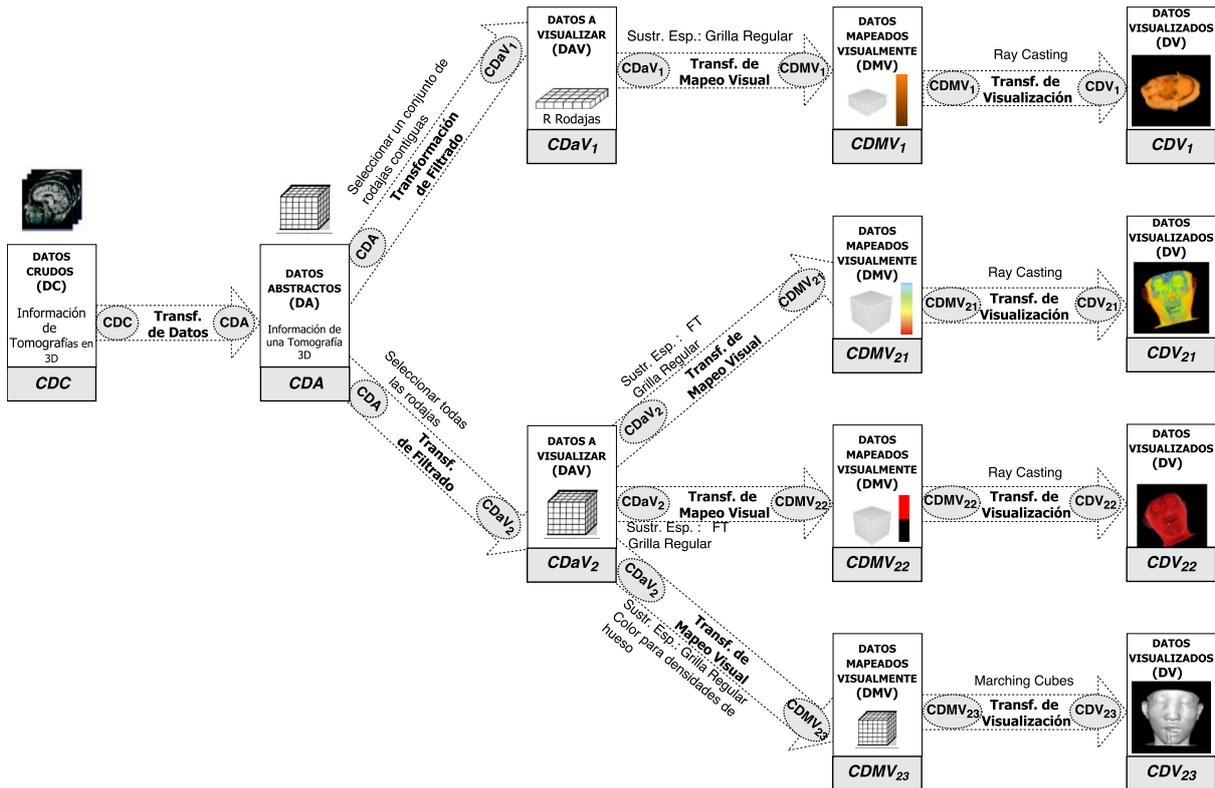


Figura 5.1: Ejemplo del MUV en ejecución [MCFE03].

distintas necesidades, se puede visualizar el conjunto 3D de la tomografía o parte de ésta. Observar que, en la *Transformación de Filtrado*, el *pipeline* se ramifica en dos caminos hacia la vista. Decimos entonces que, para el ejemplo expuesto, en la *Transformación de Filtrado* hay un único conjunto de datos de entrada CDA y dos conjuntos de datos de salida $CDaV_1$ y $CDaV_2$, uno por cada ramificación del *pipeline* originada en esta transformación. Algo similar ocurre en la *Transformación de Mapeo Visual* de la segunda rama, donde el *pipeline* se ramifica en tres caminos hacia la vista. En esta transformación se cuenta con único conjunto de datos de entrada ($CDaV_2$) y tres conjuntos de datos de salida $CDMV_{21}$, $CDMV_{22}$ y $CDMV_{23}$. Cabe aclarar que las transformaciones comparten el conjunto de datos de entrada con el estado anterior y el conjunto de datos de salida con el estado siguiente de la rama correspondiente. Es decir que, cualquier modificación que se haga a estos conjuntos en una transformación se verá reflejada en el estado correspondiente, y viceversa. En resumen, el modelo permite modelar el proceso y operar sobre sus estados y sobre sus transformaciones. Además, los resultados de operar sobre un elemento del modelo se propagará en las transformaciones y estados en el camino a la vista.

5.2. Clasificación de Interacciones

A la hora de definir una clasificación de interacciones para el MUV, una de los primeros interrogantes a resolver fue el de especificar **quién** interactúa con el proceso de visualización. Es decir, **qué usuario** va a hacer uso de este conjunto de interacciones.

El humano es un elemento esencial en el proceso de Visualización. Él quien desarrolla y crea la representación (**el programador**) y también es quien la usa (**el usuario**). En algunos casos, el desarrollador y el usuario son la misma persona, pero en general, se trata de personas diferentes.

Al utilizar una herramienta de visualización, el usuario puede estar o no familiarizado con los detalles del proceso de visualización, con la representación de los distintos conjuntos de datos presentes en cada estado ni con las transformaciones a las que se someten los datos a lo largo de todo el proceso. En base a lo expuesto, nuestra clasificación de interacciones se divide en dos grandes grupos:

- **Interacciones a Nivel del Programador:** Se trata de un conjunto de interacciones especialmente diseñadas para el programador del sistema de visualización que conoce en detalle el proceso de visualización. Se agruparon las interacciones a nivel del programador en dos clases:
 - *Interacciones de Bajo Nivel del Programador:* Interacciones que manipulan directamente los conjuntos de datos. En función del conjunto de datos sobre el que se aplican estas interacciones, se clasifican en:
 - Interacciones de Bajo Nivel del Programador Aplicadas a los *datasets*.
 - Interacciones de Bajo Nivel del Programador Aplicadas al *mapeo visual*.
 - *Interacciones de Alto Nivel del Programador:* Interacciones que manipulan directamente los estados de los datos y las transformaciones del proceso. Son interacciones simples o elementales.
- **Interacciones a Nivel del Usuario:** Se trata de un conjunto de interacciones especialmente diseñadas para el usuario del sistema de visualización que no necesariamente conoce los detalles del proceso de visualización. Las interacciones a nivel del usuario se agruparon en cuatro clases:

- *Interacciones a Nivel del Usuario Sobre los Conjuntos de Datos*: Interacciones a nivel del usuario que afectan a los conjuntos de datos.
- *Interacciones a Nivel del Usuario Sobre el Mapeo Visual*: Interacciones a nivel del usuario que manipulan el mapeo visual.
- *Interacciones a Nivel del Usuario Sobre la Vista*: Interacciones a nivel de usuario relativas a la vista.
- *Interacciones Compuestas a Nivel del Usuario*: Interacciones a nivel del usuario que se componen de otras interacciones a nivel del usuario.

5.2.1. Características de los operadores

Teniendo en cuenta la naturaleza semántica de los operadores podemos clasificarlos en:

- **Operadores de Valor**: Modifican de alguna manera el conjunto de datos subyacente, generando como resultado un nuevo conjunto de datos. Ejemplos: operadores de filtrado, selección, datos derivados, reducción dimensional y las transformaciones propiamente dichas.
- **Operadores de Vista**: Operaciones que actúan sobre los objetos gráficos y/o sobre la escena en pantalla, como rotaciones, panning, zooming y algunas interacciones más complejas como distorsiones, operaciones que modifican la apariencia y/o contenido de las vistas sin modificar el conjunto de datos subyacente.

Intuitivamente se podría pensar que las operaciones a nivel de los datos crudos son indudablemente del tipo de operadores de valor, y las aplicables a nivel de vistas son puramente del tipo de operadores de vista. Sin embargo, a nivel de las vistas esta generalización no es válida ya que las interacciones sobre las vistas pueden llegar a modificar el conjunto de datos subyacente, debiéndoselas considerar por consiguiente como operaciones de valor o al menos como interacciones complejas, es decir, como composición de distintas operaciones donde al menos una de ellas será del valor. En resumen, en el Modelo Unificado de Visualización (MUV) encontramos un cuerpo troncal de operaciones operacionalmente similares entre los distintos dominios de aplicación contemplando las características distintivas de los distintos dominios y/o técnicas particulares, constituyendo un modelo conceptual de interacción.

5.3. Interacciones a Nivel del Programador

Definimos un conjunto de interacciones básicas a nivel del programador. Éstas están especialmente diseñadas para el usuario desarrollador, que tiene pleno conocimiento del *pipeline* de visualización, y será el encargado de programar el sistema de visualización que el usuario (no programador) utilizará. Clasificamos las interacciones a nivel del programador en dos niveles: las interacciones de bajo nivel del programador (que manipulan directamente los conjuntos de datos) y las interacciones de alto nivel del programador (que manipulan directamente los estados y transformaciones de los datos en el *pipeline* de visualización). Las interacciones de bajo nivel del programador se definieron en base a las operaciones definidas para las representaciones de los conjuntos de datos y para el mapeo visual. Las interacciones de alto nivel del programador se definieron en base a las interacciones de bajo nivel del programador.

5.3.1. Interacciones de Bajo Nivel del Programador

En el MUV coexistirán dos tipos de representaciones de conjuntos de datos (ver figura 5.2). Desde la *Transformación de Datos* (TD), y hasta la *Transformación de Mapeo Visual* (TMV), los conjuntos de datos se representarán con un conjunto de *datasets* como el especificado en la sección 4.5 del capítulo 4. A partir de la *Transformación de Mapeo Visual* (TMV) y hasta la *Transformación de Visualización* (TV) este *dataset* se verá enriquecido con un mapeo visual, utilizando la representación detallada en la sección 4.6 del capítulo 4. En la *Transformación de Visualización* (TV) y el estado *Datos Visualizados* (DV) entra en juego lo que denominamos *Soporte Gráfico*, que se refiere al resultado de derivar una imagen a partir de la descripción geométrica de una escena. Dado que este proceso involucra el *pipeline* gráfico y cae en un dominio diferente al abordado en esta tesis (Computación Gráfica), no profundizaremos en la definición del mismo.

Los estados y transformaciones del proceso que comparten la misma representación del conjunto de datos compartirán también las interacciones básicas a bajo nivel del programador que podrán aplicarse a esos estados y transformaciones. En base a lo expuesto, clasificaremos las interacciones de bajo nivel del programador en dos grupos: las interacciones de bajo nivel aplicadas a los *datasets*, y las interacciones de bajo nivel aplicadas al *mapeo visual* (ver figura 5.3). Todas estas interacciones se componen de operadores de

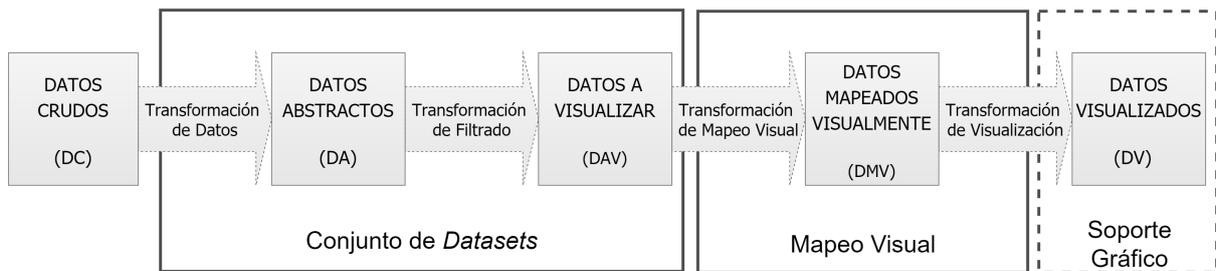


Figura 5.2: Conjuntos de Datos en el MUV.

valor, ya que modifican el conjunto de datos subyacente, ya sea un *dataset* o un *mapeo visual*.

Cabe aclarar que dado que los *Datos Crudos* provienen de distintos dominios de aplicación y en distintos formatos, no es posible definir una representación generalizada ni un conjunto generalizado de interacciones para los mismos.

5.3.1.1. Interacciones de Bajo Nivel del Programador Aplicadas a los *Datasets*

Estas interacciones se aplican a los estados *Datos Abstractos* (DA) y *Datos a Visualizar* (DaV), y a las transformaciones de *Datos* (TD) y de *Filtrado* (TF), que comparten la misma representación para los conjuntos de datos que manejan (*datasets*).

A continuación se define el conjunto básico de interacciones de bajo nivel del programador aplicadas a los *datasets*.

Interacciones de Bajo Nivel del Programador Aplicadas a los *Datasets*

crearDataset():Dataset

Esta interacción permite crear un nuevo *dataset* vacío.

eliminarDataset(Dataset)

Esta interacción permite eliminar el *Dataset* especificado.

insertarDato(Dataset, Dato):idDato

Esta interacción permite incorporar un nuevo *Dato* al *Dataset* retornando su identificador *idDato* asociado.

eliminarDato(Dataset, Dato)

Esta interacción permite eliminar un *Dato* del *Dataset* especificado.

obtenerDato(Dataset, idDato):Dato

Esta interacción permite obtener el *Dato* asociado al *idDato* en el *Dataset* especificado.

obtenerIdDato(Dataset, Dato):idDato

Esta interacción permite obtener el *idDato* asociado al *Dato* en el *Dataset* especificado.

Sigue en la página siguiente.

Interacciones de Bajo Nivel del Programador Aplicadas a los <i>Datasets</i>	
<i>eliminarAtributo(Dataset, idAtributo)</i>	Esta interacción permite eliminar el atributo con identificador <i>idAtributo</i> del <i>Dataset</i> especificado.
<i>insertarAtributo(Dataset, Nombre, Tipo):idAtributo</i>	Esta interacción permite agregar un nuevo atributo al <i>Dataset</i> , incrementando la dimensionalidad de los datos en uno y retornando el identificador <i>idAtributo</i> asociado al nuevo atributo. El nuevo atributo tendrá un <i>Nombre</i> y un <i>Tipo</i> especificados.
<i>obtenerIdAtributo(Dataset, Atributo):idAtributo</i>	Esta consulta permite obtener el identificador <i>idAtributo</i> asociado al <i>Atributo</i> en el <i>Dataset</i> especificado.
<i>modificarValor(Dataset, Dato, idAtributo, Valor)</i>	Esta interacción permite modificar el <i>Valor</i> asociado a un atributo identificado por <i>idAtributo</i> , para un <i>Dato</i> en particular en el <i>Dataset</i> especificado.
<i>obtenerValor(Dataset, Dato, idAtributo):Valor</i>	Esta consulta retorna el <i>Valor</i> asociado a un atributo identificado por <i>idAtributo</i> para un <i>Dato</i> en particular en el <i>Dataset</i> especificado.
<i>agrupar(Dataset, Colección<Dato>)</i>	Esta operación agrupa una colección de datos en el <i>Dataset</i> especificado.
<i>desagrupar(Dataset, Colección<Dato>)</i>	Esta operación desagrupa una colección de datos previamente agrupada en el <i>Dataset</i> especificado.
<i>clusters(Dataset):Iterable<Dataset></i>	Esta consulta retorna un iterable de los clusters del <i>Dataset</i> especificado.
<i>datos(Dataset):Iterable<Dato></i>	Esta consulta retorna una colección iterable de datos que permite recorrer todos los datos almacenados en el <i>Dataset</i> especificado.
<i>atributos(Dataset):Iterable<Atributo></i>	Esta consulta retorna una colección iterable de atributos del <i>Dataset</i> especificado.

Tabla 5.1: Interacciones de Bajo Nivel del Programador Aplicadas a los *Datasets*.

5.3.1.2. Interacciones de Bajo Nivel del Programador Aplicadas al *Mapeo Visual*

Estas interacciones se aplican en el estado *Datos Mapeados Visualmente* (DMV) y en las transformaciones de *Mapeo Visual* (TMV) y de *Visualización* (TV), que comparten la misma representación para sus conjuntos de datos (*mapeo visual*).

A continuación se define el conjunto básico de interacciones de bajo nivel del programador aplicadas al *mapeo visual*.

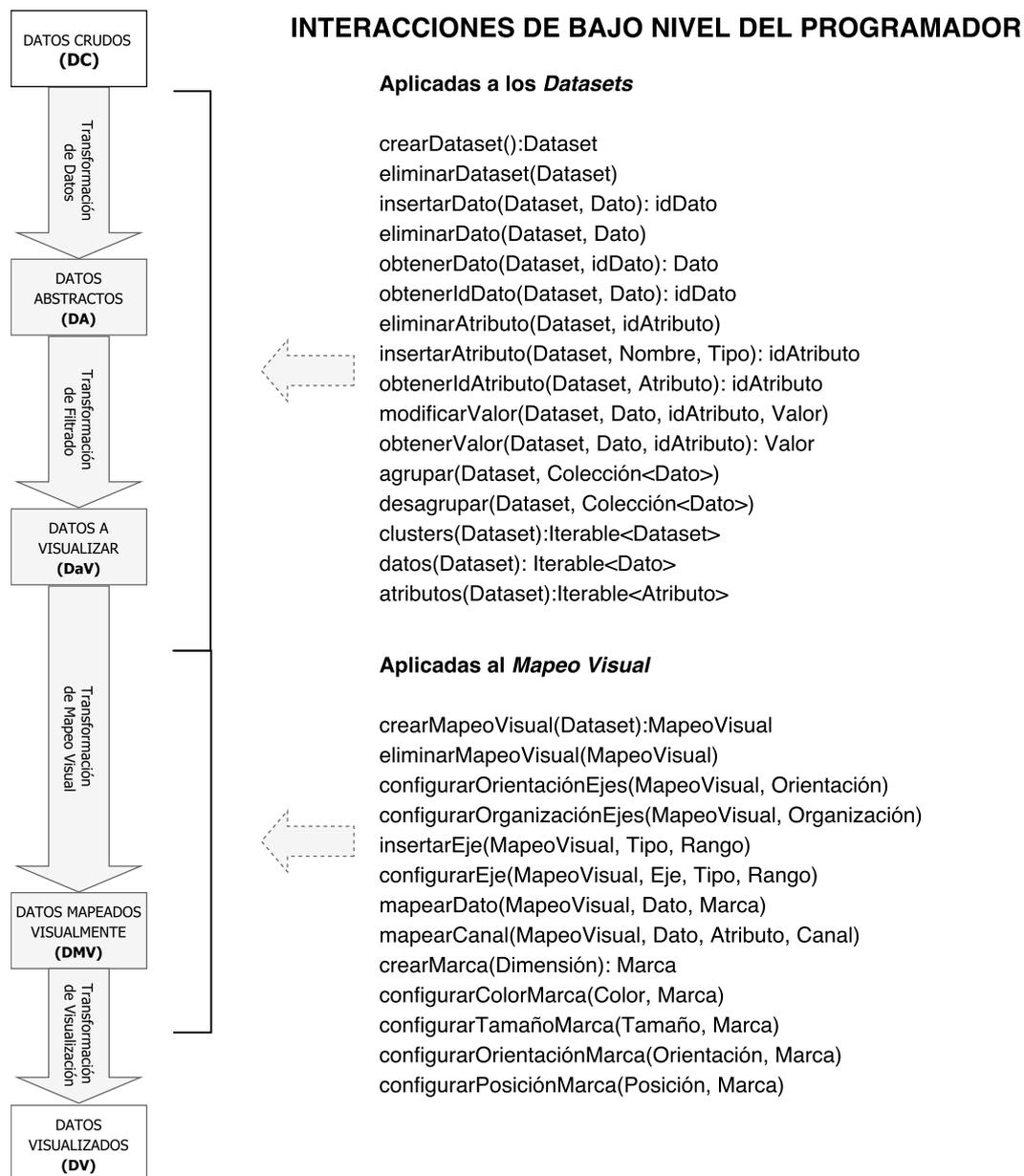


Figura 5.3: Interacciones de Bajo Nivel del Programador: en la figura se listan las interacciones de bajo nivel del programador a aplicadas a los *dataset* (arriba) y al *mapeo visual* (abajo).

Interacciones de Bajo Nivel del Programador Aplicadas al *Mapeo Visual*

crearMapeoVisual(Dataset): MapeoVisual

Esta interacción permite crear un nuevo *Mapeo Visual*, asociando al mismo el *Dataset* especificado.

eliminarMapeoVisual(MapeoVisual)

Esta interacción permite eliminar el *Mapeo Visual* especificado.

configurarOrientaciónEjes(MapeoVisual, Orientación)

Esta interacción permite configurar la *orientación* que tendrán los ejes en el *Mapeo Visual* especificado.

Sigue en la página siguiente.

Interacciones de Bajo Nivel del Programador Aplicadas al <i>Mapeo Visual</i>	
<i>configurarOrganizaciónEjes(Mapeo Visual, Organización)</i>	Esta interacción permite configurar la <i>organización</i> que tendrán los ejes en el <i>Mapeo Visual</i> especificado.
<i>insertarEje(Mapeo Visual, Tipo, Rango)</i>	Esta interacción permite incorporar un nuevo eje al sustrato espacial del <i>Mapeo Visual</i> especificado.
<i>configurarEje(Mapeo Visual, Eje, Tipo, Rango)</i>	Esta interacción permite configurar el <i>Eje</i> en el sustrato espacial del <i>Mapeo Visual</i> especificado.
<i>mapearDato(Mapeo Visual, Dato, Marca)</i>	Esta interacción permite asociar un <i>Dato</i> con una <i>Marca</i> en el <i>Mapeo Visual</i> especificado.
<i>mapearCanal(Mapeo Visual, Dato, Atributo, Canal)</i>	Esta interacción permite asociar un <i>Atributo</i> de un <i>Dato</i> con un <i>Canal</i> de la marca que lo representa en el <i>Mapeo Visual</i> especificado.
<i>crearMarca(Dimensión):Marca</i>	Esta interacción permite crear una <i>Marca</i> .
<i>configurarColorMarca(Color, Marca)</i>	Esta interacción permite configurar una <i>Marca</i> asociándole el <i>Color</i> especificado.
<i>configurarTamañoMarca(Tamaño, Marca)</i>	Esta interacción permite configurar una <i>Marca</i> asociándole el <i>Tamaño</i> especificado.
<i>configurarOrientaciónMarca(Orientación, Marca)</i>	Esta interacción permite configurar una <i>Marca</i> asociándole la <i>Orientación</i> especificada.
<i>configurarPosiciónMarca(Posición, Marca)</i>	Esta interacción permite configurar una <i>Marca</i> asociándole la <i>Posición</i> especificada.

Tabla 5.2: Interacciones de Bajo Nivel del Programador Aplicadas al *Mapeo Visual*.

5.3.2. Interacciones de Alto Nivel del Programador

Las interacciones de alto nivel del programador son aquellas que manipulan directamente los estados y las transformaciones del proceso de visualización. Estas interacciones se definen en función de las interacciones de bajo nivel del programador.

Si bien Martig *et al.* al presentar su MUV [MCFE03] sugieren que es posible definir interacciones para cada uno de los estados y transformaciones del mismo (ver figura 2.6), en nuestra propuesta de clasificación hay dos estados para los cuales no se definirán interacciones de alto nivel del programador: estos son los *Datos a Visualizar* (DaV) y los *Datos Mapeados Visualmente* (DMV). Como se explicó anteriormente (sección 2.3.1.3), los DaV se componen de todos los DA o de un subconjunto de ellos y están compuestos

por los datos que van a estar presentes en la visualización. Este conjunto de datos se obtiene a través de la *Transformación de Filtrado* (TF). Consideramos que definir interacciones que modifiquen el conjunto de DaV en su propio estado no sería correcto, ya que conceptualmente es en la TF donde se realizan estas acciones de modificación. Una situación similar ocurre en el estado de (DMV), donde cualquier modificación del mapeo visual debe realizarse conceptualmente en la *Transformación de Mapeo Visual*.

En la figura 5.4 se listan las interacciones de alto nivel del programador disponibles en cada estado y transformación del *pipeline*. Todas las interacciones definidas para la etapa del *pipeline* que comienza en el estado DA y termina en el estado DMV se componen de operadores de valor, ya que modifican el conjunto de datos subyacente, ya sea un *dataset* o un *mapeo visual*. En cambio, las interacciones definidas para la transformación TV y DV se componen de operadores de vista, ya que afectan únicamente la apariencia de la vista sin afectar el conjunto de datos subyacente.

5.3.2.1. Interacciones de Alto Nivel del Programador en los DC

Dado que los datos crudos provienen de distintos dominios de aplicación y en distintos formatos, no es posible definir una representación generalizada para los mismos, y en consecuencia tampoco es posible definir un conjunto de operaciones para su manipulación. Se supone que los distintos dominios de aplicación proveen estos datos y serán los que disponen de las operaciones para su manipulación. Limitaremos el alcance de las interacciones sobre este estado únicamente a cargar la fuente de datos.

Interacciones de Alto Nivel del Programador en los Datos Crudos
<i>CargarFuenteDatos(DC)</i>
Esta interacción permite seleccionar una fuente de datos y cargarla en el sistema. Si el conjunto de DC no se encuentra vacío, éste será modificado.

Tabla 5.3: Interacciones de Alto Nivel del Programador en los Datos Crudos.

5.3.2.2. Interacciones de Alto Nivel del Programador en la TD

Esta transformación tiene un rol esencial en cuanto a la manipulación de los datos, ya que es la que permite introducir los mismos en el *pipeline* llevándolos a un formato interno manejable en el MUV. En esta transformación, denominaremos *CDC* al conjunto de DC de entrada y *CDA_i* al conjunto de DA de salida correspondiente a la *i*-ésima

INTERACCIONES DE ALTO NIVEL DEL PROGRAMADOR

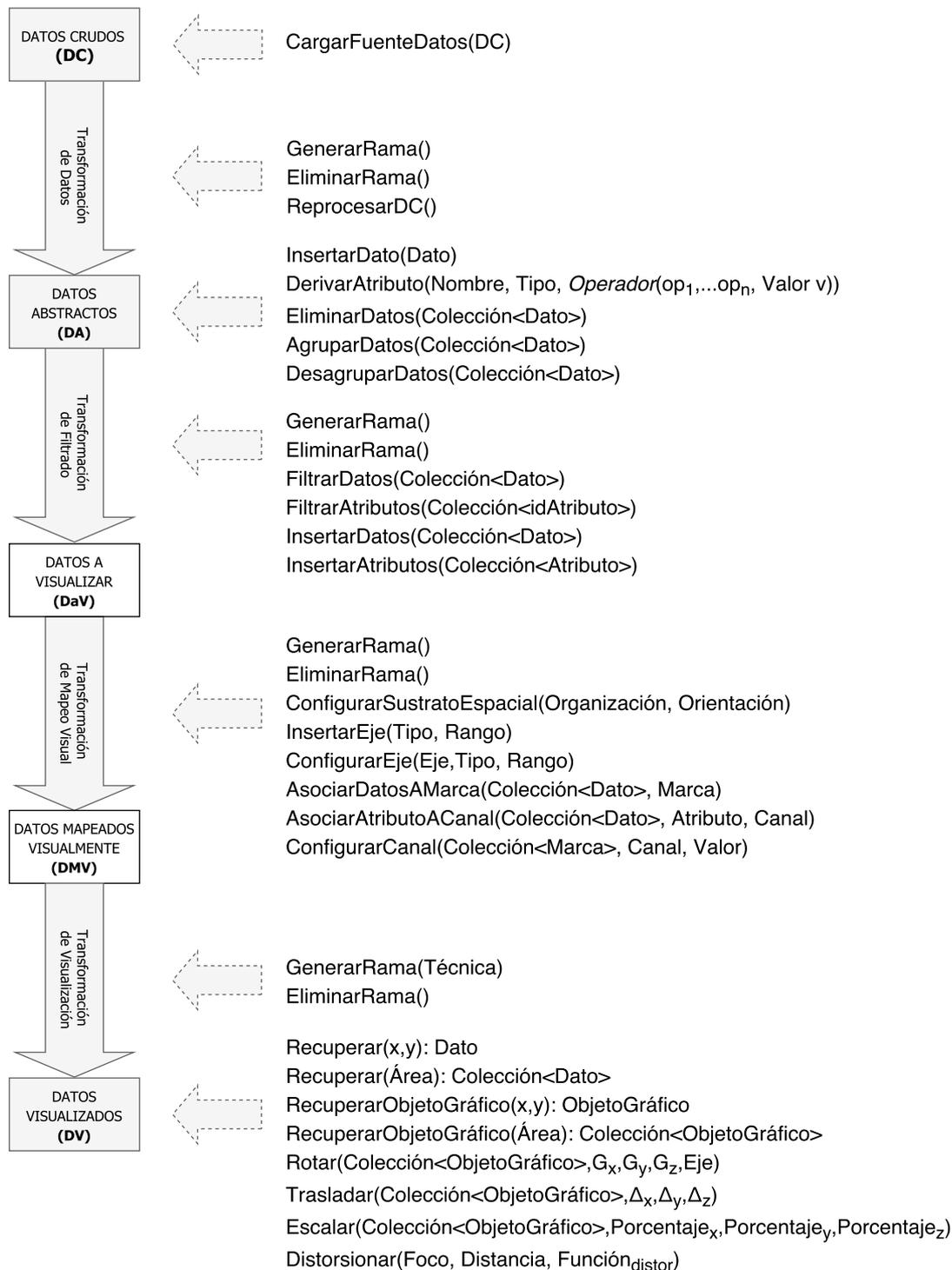


Figura 5.4: Interacciones de Alto Nivel del Programador.

rama. Los conjuntos de DA de salida se representarán mediante alguno de los *datasets* detallados en la sección 4.5 del capítulo 4.

Limitamos el alcance de las operaciones sobre este estado a nivel de procesar la fuente de datos y generar (o modificar) el conjunto de DA correspondiente a la rama en la que se originó la interacción. En esta transformación, las tres interacciones de alto nivel del programador disponibles son *GenerarRama*, *EliminarRama* y *ReprocesarDC*. Las interacciones *GenerarRama* y *ReprocesarDC* involucran un procesamiento del *CDC* para obtener o modificar un conjunto de datos abstractos. En función del tipo de conjunto de datos con el que se esté trabajando, se creará el tipo de *dataset* correspondiente. Si, por ejemplo, el conjunto de DC representa un conjunto de datos relacionales, entonces el resultado de las interacciones que se aplican en esta transformación será un *dataset* con estructura de árbol o grafo. Si los datos provienen de un dominio continuo y son resultado de un muestreo, entonces el *dataset* indicado para representarlos tendrá estructura de Grilla. Si estamos trabajando con Visualización de Información, el *dataset* que probablemente sea más adecuado para representar los DA será una tabla, conjunto, etc. Esta interacción ejecuta algoritmos específicos que harán uso de las operaciones provistas por los *datasets* y sus implementaciones específicas.

A continuación se detallan las interacciones de alto nivel del programador provistas en esta transformación:

Interacciones de Alto Nivel del Programador en la Transformación de Datos

GenerarRama()

Esta interacción genera un nuevo conjunto de DA a partir del conjunto de DC. El resultado de esta operación es la ramificación del proceso a partir de esta transformación.

Implementación en función de las interacciones de bajo nivel:

GenerarRama()

DE: *CDC*

DS: $CDA_{NuevaRama}$

$CDA_{NuevaRama} \leftarrow \text{ProcesarDatosCrudos}(CDC)$

fin

EliminarRama()

Esta interacción elimina el conjunto de datos CDA_i , siendo i la i -ésima rama dónde se generó la interacción. El resultado de esta operación es la eliminación de una rama completa del *pipeline*.

Sigue en la página siguiente.

Interacciones de Alto Nivel del Programador en la Transformación de Datos

Implementación en función de las interacciones de bajo nivel:

EliminarRama()

DE: i : Rama

eliminarDataset(CDA_i)

▷ Interacción de bajo nivel del progr. (*datasets*).

fin

ReprocesarDC()

Esta interacción reprocesa el *CDC* y modifica el conjunto CDA_i , siendo i la i -ésima rama dónde se aplicó esta interacción. Esta interacción no genera una nueva rama en el *pipeline*, sino que modifica el conjunto de DA correspondiente a la rama dónde se generó la interacción. Esta interacción tiene sentido en un contexto en el que el usuario desea incorporar a los DA un nuevo conjunto o subconjunto de DC. Es claro que este reprocesamiento será posible únicamente si los nuevos datos crudos incorporados son compatibles con los almacenados previamente en los DA.

Implementación en función de las interacciones de bajo nivel:

ReprocesarDC()

DE: *CDC*, i : Rama

DS: CDA_i

$CDA_i \leftarrow$ ProcesarDatosCrudos(*CDC*)

fin

Tabla 5.4: Interacciones de Alto Nivel del Programador en la Transformación de Datos.

El procesamiento de los DC (*ProcesarDatosCrudos(CDC)*) invocado en las interacciones *GenerarRama()* y *ReprocesarDC()* es fuertemente dependiente del campo de aplicación y no es posible definir un algoritmo general para su ejecución. Sin embargo es claro que involucrará los procesos de filtrado y derivación descritos a continuación:

- *Filtrado de Datos*

Permite realizar un filtrado de los datos a incluir en el conjunto de DA con el objetivo de seleccionar qué datos específicos del conjunto de los DC constituirán los DA.

- *Filtrado de Atributos*

Este proceso permite realizar un filtrado de los atributos que serán incluidos en el conjunto de DA. Cabe aclarar que esta interacción tendrá lugar únicamente si el conjunto de DC corresponde a un tipo de conjunto de datos que soporte atributos.

- *Derivar Dato*

Permite insertar un nuevo dato al conjunto de DA que se deriva a partir de otros datos del conjunto de DC (*metadatos*).

- *Derivar Atributos*

Permite insertar un nuevo atributo a los datos almacenados al conjunto de DA que se derivan a partir de otros atributos del conjunto de DC (*metadatos*). Por ejemplo, el agregar al conjunto de datos el cálculo de un promedio de un conjunto de atributos.

5.3.2.3. Interacciones de Alto Nivel del Programador en los DA

En este estado se proveerán interacciones para la generación de *metadatos* y la eliminación de datos del conjunto de DA. Con respecto a la generación de *metadatos*, se proveerán únicamente interacciones para insertar nuevos datos (*InsertarDato*) y atributos (*DerivarAtributo*) ambos derivados a partir de otros datos o atributos del conjunto de datos. Estos nuevos datos o atributos generados se podrán derivar únicamente a partir los datos y/o atributos presentes en el conjunto de DA, denominado *CDA*, único conjunto de datos presente en este estado. Además, se proveerá una interacción para eliminar datos del conjunto de *CDA*. Esta operación no podrá deshacerse sin volver a reprocesar los DC en la transformación anterior.

Interacciones de Alto Nivel del Programador en los Datos Abstractos

InsertarDato(Dato *d*)

Permite insertar un nuevo dato al conjunto *CDA*.

Implementación en función de las interacciones de bajo nivel:

InsertarDato(DATO *D*)

DE: *CDA*, *d*: Dato a insertar.

DS: *CDA*

insertarDato(*CDA*, *d*)

▷ Interacción de bajo nivel del progr. (*datasets*).

fin

DerivarAtributo(Nombre *n*, Tipo *t*, OPERADOR(*op*₁, ..., *op*_{*n*}, Valor *v*)) o)

Esta interacción permite insertar un nuevo atributo a los datos almacenados en el conjunto de DA.

Este nuevo atributo se deriva a partir de otros atributos del conjunto de DA (*metadatos*).

Sigue en la página siguiente.

Interacciones de Alto Nivel del Programador en los Datos Abstractos

Implementación en función de las interacciones de bajo nivel:

DerivarAtributo(NOMBRE n , TIPO t , OPERADOR(op_1, \dots, op_n , Valor v)) o)

DE: CDA , n : Nombre del nuevo atributo

t : Tipo del nuevo atributo.

o : Operador que se aplicará para generar los valores del nuevo atributo.

DS: CDA

$idAtributo_{nuevo} \leftarrow insertarAtributo(CDA, n, t)$ ▷ Interacción de bajo nivel del progr. (*datasets*).

para todo Dato d en CDA **realizar**

$operador_1 \leftarrow obtenerValor(CDA, d, op_1)$ ▷ Interacción de bajo nivel del progr. (*datasets*).

$operador_2 \leftarrow obtenerValor(CDA, d, op_2)$ ▷ Interacción de bajo nivel del progr. (*datasets*).

...

$operador_n \leftarrow obtenerValor(CDA, d, op_n)$ ▷ Interacción de bajo nivel del progr. (*datasets*).

$valor \leftarrow o(operador_1, operador_2, \dots, operador_n, v)$

$modificarValor(CDA, d, idAtributo_{nuevo}, valor)$ ▷ Interacción de bajo nivel del progr. (*datasets*).

fin para

fin

EliminarDatos(Colección<Dato> c)

Permite eliminar una colección de datos del conjunto de DA.

Implementación en función de las interacciones de bajo nivel:

EliminarDato(COLECCIÓN<DATO> c)

DE: CDA , c : Colección de datos a eliminar del conjunto CDA .

DS: CDA

para todo Dato d en c **realizar**

$eliminarDato(CDA, d)$ ▷ Interacción de bajo nivel del progr. (*datasets*).

fin para

fin

AgruparDatos(Colección<Dato> c)

Permite agrupar los datos contenidos en la colección especificada en el conjunto de DA.

Sigue en la página siguiente.

Interacciones de Alto Nivel del Programador en los Datos Abstractos

Implementación en función de las interacciones de bajo nivel:

AgruparDatos(COLECCIÓN<DATO> *c*)

DE: *CDA*, *c*: Colección de datos a agrupar del conjunto *CDA*.

DS: *CDA*

agruparDatos(*CDA*, *c*)

▷ Interacción de bajo nivel del progr. (*datasets*).

fin

DesagruparDatos(Colección<Dato> *c*)

Permite desagrupar los datos contenidos en la colección especificada en el conjunto de DA. Requiere que los datos a desagrupar hayan sido previamente agrupados.

Implementación en función de las interacciones de bajo nivel:

DesagruparDatos(COLECCIÓN<DATO> *c*)

DE: *CDA*, *c*: Colección de datos a desagrupar del conjunto *CDA*.

DS: *CDA*

desagruparDatos(*CDA*, *c*)

▷ Interacción de bajo nivel del progr. (*datasets*).

fin

Tabla 5.5: Interacciones de Alto Nivel del Programador en los Datos Abstractos.

5.3.2.4. Interacciones de Alto Nivel del Programador en la TF

Esta transformación permite al usuario filtrar los datos que desea visualizar. En este punto el usuario decide qué datos y qué atributos presentes en los DA participarán en la vista a generar. Esta participación puede ser para su mostrado inicial o ante un requerimiento por parte del usuario en una etapa posterior. En esta transformación, denominaremos *CDA* al conjunto de DA de entrada y *CDaV_i* al conjunto de DaV de salida correspondiente a la *i*-ésima rama.

Como en todas las transformaciones del *pipeline*, se proveen interacciones de alto nivel del programador para generar y para eliminar una rama (*GenerarRama()* y *EliminarRama()*). Además, se proveen interacciones específicas de filtrado e inserción de datos y atributos.

A continuación se detallan las interacciones de alto nivel del programador provistas en esta transformación:

Interacciones de Alto Nivel del Programador en la Transformación de Filtrado

GenerarRama()

Esta interacción genera un nuevo conjunto $CDaV$ a partir del CDA de entrada, resultando en la ramificación del proceso a partir de esta transformación.

Implementación en función de las interacciones de bajo nivel:

GenerarRama()

DE: CDA

DS: $CDaV_{NuevaRama}$

$CDaV_{nuevarama} \leftarrow \text{crearDataset}()$

▷ Interacción de bajo nivel del progr. (*datasets*).

para todo $Dato\ d$ en CDA **realizar**

$\text{insertarDato}(CDaV_{NuevaRama}, d)$

▷ Interacción de bajo nivel del progr. (*datasets*).

fin para

fin

EliminarRama()

Esta interacción elimina el conjunto de datos DaV_i , siendo i la i -ésima rama dónde se aplicó esta interacción, resultando en la eliminación de la i -ésima rama del *pipeline*.

Implementación en función de las interacciones de bajo nivel:

EliminarRama()

DE: i : Rama

$\text{eliminarDataset}(CDaV_i)$

▷ Interacción de bajo nivel del progr. (*datasets*).

fin

FiltrarDatos(Colección<Dato> c)

Permite realizar un filtrado de los datos incluidos en el DaV_i , siendo i la i -ésima rama dónde se aplicó esta interacción. De este modo, se pueden excluir algunos valores de datos del CDA .

Observar que este filtrado es acumulativo, ya que se opera directamente sobre el $CDaV_i$.

Sigue en la página siguiente.

Interacciones de Alto Nivel del Programador en la Transformación de Filtrado

Implementación en función de las interacciones de bajo nivel:

FiltrarDatos(COLECCIÓN<DATO> *c*)

DE: *c*, *CDaV_i*, *i:rama*

DS: *CDaV_i*

para todo *Dato d* en *c* **realizar**

si *d* está presente en *CDaV_i* **entonces**

 eliminarDato(*CDaV_i*, *d*)

▷ Interacción de bajo nivel del progr. (*datasets*).

fin si

fin para

fin

FiltrarAtributos(Colección<idAtributo> *c*)

Este proceso permite realizar un filtrado de los atributos incluidos en *CDA*, excluyendo los atributos especificados. Esta interacción reduce la dimensionalidad del conjunto de datos en la cantidad de atributos que se deseen eliminar. Cabe aclarar que esta interacción tendrá lugar únicamente si el conjunto de DA corresponde a un tipo de conjunto de datos que soporte atributos, como por ejemplo una Tabla, una Grilla, Geometría, etc.

Implementación en función de las interacciones de bajo nivel:

FiltrarAtributos(COLECCIÓN<IDATRIBUTO> *c*)

DE: *c*, *CDaV_i*, *i:rama*

DS: *CDaV_i*

para todo *idAtributo idA* en *c* **realizar**

si *idA* está presente en *CDaV_i* **entonces**

 eliminarAtributo(*CDaV_i*, *idA*)

▷ Interacción de bajo nivel del progr. (*datasets*).

fin si

fin para

fin

InsertarDatos(Colección<Dato> *c*)

Permite insertar datos del *CDA* en el conjunto de DaV que previamente habían sido filtrados.

Sigue en la página siguiente.

Interacciones de Alto Nivel del Programador en la Transformación de Filtrado

Implementación en función de las interacciones de bajo nivel:

InsertarDatos(COLECCIÓN<DATO> c)

DE: c, CDaV_i, CDA, i:rama

DS: CDaV_i

para todo Dato d en c **realizar**

si d está presente en CDA y no está presente en CDaV_i **entonces**

insertarDato(CDaV_i, d)

▷ Interacción de bajo nivel del progr. (datasets).

fin si

fin para

fin

InsertarAtributos(Colección<idAtributo> c)

Este proceso permite insertar atributos del CDA al conjunto de DaV que inicialmente habrían sido filtrados.

Implementación en función de las interacciones de bajo nivel:

InsertarAtributos(COLECCIÓN<IDATRIBUTO> c)

DE: c, CDA, CDaV_i, i:rama

DS: CDaV_i

para todo idAtributo idA en c **realizar**

si idA está presente en CDA y no está presente en CDaV_i **entonces**

insertarAtributo(CDaV_i, idA)

▷ Interacción de bajo nivel del progr. (dataset).

fin si

fin para

para todo idAtributo idA en c **realizar**

para todo Dato d en CDaV_i **realizar**

modificarValor(CDaV_i, d, idA, obtenerValor(CDA,d,idA))) ▷ Interacciones de bajo

nivel del progr. (dataset).

fin para

fin para

fin

Tabla 5.6: Interacciones de Alto Nivel del Programador en la Transformación de Filtrado.

Las interacciones de filtrado (*FiltrarDatos* y *FiltrarAtributos*) permiten decidir que datos y atributos del conjunto de DA no serán incorporados al conjunto de DaV resultante.

Dado que la interacción *GenerarRama()*, al crear un nuevo conjunto de datos lo inicializa con todos los datos y atributos presentes en el conjunto de DA, es posible implementar las operaciones de filtrado de modo tal que operen directamente sobre el conjunto de DaV, sin anular operaciones de filtrado realizadas previamente. Es decir que las operaciones de filtrado son acumulativas.

Finalmente, para deshacer una interacción de filtrado, los datos deben volver a traerse del conjunto de DA. Con tal fin se proveen interacciones para incorporar datos al conjunto de DaV que se encuentran presentes en el conjunto de DA pero pueden haber sido previamente filtrados (*InsertarDatos* e *InsertarAtributos*).

5.3.2.5. Interacciones de Alto Nivel del Programador en la TMV

Esta transformación es netamente interactiva y le permite al usuario definir cómo quiere visualizar sus datos. Aquí se realiza el mapeo visual, es decir, se establecen las estructuras visuales, qué atributos se mapearán espacialmente y qué elementos visuales (marcas y canales) se utilizarán.

Las interacciones de alto nivel del programador provistas en esta transformación tienen como principal objetivo definir el sustrato espacial y el sustrato gráfico que tendrá la visualización. En esta transformación, el conjunto de DaV (representado con un conjunto de *datasets*) se transformará en un *mapeo visual* como el definido en la sección 4.6 del capítulo 4.

En esta transformación, denominaremos *CDaV* al conjunto de DaV de entrada y *CDMV_i* al conjunto de DMV de salida correspondiente a la *i*-ésima rama.

A continuación se detallan las interacciones de alto nivel del programador provistas en esta transformación:

Interacciones de Alto Nivel del Programador en la Transformación de Mapeo Visual

GenerarRama()

Esta interacción genera un nuevo conjunto de DMV a partir del conjunto de DaV actual. El resultado de esta operación es la ramificación del proceso a partir de esta transformación.

Sigue en la página siguiente.

Interacciones de Alto Nivel del Programador en la Transformación de Mapeo Visual

Implementación en función de las interacciones de bajo nivel:

GenerarRama()

DE: $CDaV$

DS: $CDMV_{NuevaRama}$

$CDMV_{NuevaRama} \leftarrow \text{crearMapeoVisual}(CDaV)$ ▷ Interacción de bajo nivel del progr. (*mapeo visual*).

fin

EliminarRama()

Esta interacción elimina el conjunto de datos $CDMV_i$, siendo i la i -ésima rama dónde se aplicó esta interacción. El resultado de esta operación es la eliminación de una rama completa del *pipeline*.

Implementación en función de las interacciones de bajo nivel:

EliminarRama()

DE: i : Rama

$\text{eliminarMapeoVisual}(CDMV_i)$ ▷ Interacción de bajo nivel del progr. (*mapeo visual*).

fin

ConfigurarSustratoEspacial(Organización *org*, Orientación *or*)

Permite configurar el sustrato espacial del *mapeo visual* configurando la organización y orientación de los ejes.

Implementación en función de las interacciones de bajo nivel:

ConfigurarSustratoEspacial(ORGANIZACIÓN ORG, ORIENTACIÓN OR)

DE: i : Rama

org: Organización del sustrato espacial

or: Orientación del sustrato espacial

DS: $CDMV_i$

$\text{configurarOrientaciónEjes}(CDMV_i, \text{or})$ ▷ Interacción de bajo nivel del progr. (*mapeo visual*).

$\text{configurarOrganizaciónEjes}(CDMV_i, \text{org})$ ▷ Interacción de bajo nivel del progr. (*mapeo visual*).

fin

InsertarEje(Tipo t , Rango r)

Esta interacción permite incorporar un nuevo eje al sustrato espacial del *mapeo visual*. El nuevo eje tendrá el tipo y el rango especificados.

Sigue en la página siguiente.

Interacciones de Alto Nivel del Programador en la Transformación de Mapeo Visual

Implementación en función de las interacciones de bajo nivel:

InsertarEje(TIPO T, RANGO R)

DE: *i*: Rama

t: Tipo del eje

r: Rango del eje

DS: $CDMV_i$

insertarEje($CDMV_i$, *t*, *r*)

▷ Interacción de bajo nivel del progr. (*mapeo visual*).

fin

ConfigurarEje(Eje *e*, Tipo *t*, Rango *r*)

Esta interacción permite configurar el Eje *e* con el tipo y el rango especificados. Implementación en función de las interacciones de bajo nivel:

ConfigurarEje(EJE E, TIPO T, RANGO R)

DE: *i*: Rama

e: Eje a configurar

t: Tipo del eje

r: Rango del eje

DS: $CDMV_i$

configurarEje($CDMV_i$, *e*, *t*, *r*)

▷ Interacción de bajo nivel del progr. (*mapeo visual*).

fin

AsociarDatosAMarca(Colección<Dato> *c*, Marca *m*)

Esta interacción permite asociar un conjunto de datos con una *Marca* en particular.

Sigue en la página siguiente.

Interacciones de Alto Nivel del Programador en la Transformación de Mapeo Visual

Implementación en función de las interacciones de bajo nivel:

AsociarDatosAMarca(COLECCIÓN<DATO> *c*, MARCA *m*)

DE: *CDaV*, *i*: Rama

c: Colección de datos que se asociará con la marca *m*

m: Marca a asociar al conjunto de datos *c*

DS: *CDMV_i*

para todo *Dato d* en *c* **realizar**

mapearDato(*CDMV_i*, *d*, *m*) ▷ Interacción de bajo nivel del progr. (*mapeo visual*).

fin para

fin

AsociarAtributoACanal(Colección<Dato> *col*, Atributo *a*, Canal *c*)

Esta interacción permite asociar un atributo de un conjunto de datos a un *Canal* en particular de la marca que lo representa. Es requisito que todos los datos almacenados en la colección *c* estén asociados a la misma marca o a marcas que soporten el canal *c*.

Implementación en función de las interacciones de bajo nivel:

AsociarAtributoACanal(COLECCIÓN<DATO> *COL*, ATRIBUTO *A*, CANAL *C*)

DE: *i*: Rama

col: Conjunto de datos que se verán afectados.

a: Atributo que se asociará al canal *c*.

c: Canal.

DS: *CDMV_i*

para todo *Dato d* en *c* **realizar**

mapearCanal(*CDMV_i*, *d*, *a*, *c*) ▷ Interacción de bajo nivel del progr. (*mapeo visual*).

fin para

fin

ConfigurarCanal(Colección<Marca> *m*, Canal *c*, Valor *v*)

Esta interacción permite configurar el valor de un canal *c* a una colección de marcas en particular. La colección de marcas debe ser homogénea, o debe estar compuesta por marcas que soporten el canal *c*.

Sigue en la página siguiente.

Interacciones de Alto Nivel del Programador en la Transformación de Mapeo Visual

Implementación en función de las interacciones de bajo nivel:

ConfigurarCanal(COLECCIÓN<MARCA> COL, CANAL C, VALOR V)

DE: *i*: Rama

col: Colección de marcas a configurar

c: Canal a configurar

v: Valor

DS: $CDMV_i$

si *c* es: **entonces**

Color: configurarColorMarca(*v*, *m*) ▷ Interacción de bajo nivel del progr. (*mapeo visual*).

Tamaño: configurarTamañoMarca(*v*, *m*) ▷ Interacción de bajo nivel del progr. (*mapeo visual*).

Posición: configurarPosiciónMarca(*v*, *m*) ▷ Interacción de bajo nivel del progr. (*mapeo visual*).

Orientación: configurarOrientaciónMarca(*v*, *m*) ▷ Int. de bajo nivel del progr. (*mapeo visual*).

fin si

fin

Tabla 5.7: Interacciones de Alto Nivel del Programador en la Transformación de Mapeo Visual.

5.3.2.6. Interacciones de Alto Nivel del Programador en la TV

Es la transformación encargada de generar la representación visual en pantalla según lo expresado en los DMV. Una vez seleccionada una técnica de visualización que soporte las restricciones presentes en el estado anterior, esta transformación es la que genera la vista. Para un determinado conjunto de DMV pueden existir varias técnicas que lo soporten, debiendo el usuario optar por alguna de ellas para obtener de esta manera los DV.

A partir de esta transformación es necesario contar con un *Soporte Gráfico* que presente el resultado de derivar una imagen a partir de la descripción de una escena. Ese proceso involucra el *pipeline* gráfico y cae en un dominio diferente al abordado en esta tesis (Computación Gráfica), razón por la cual no profundizaremos en la definición del mismo. Asumiremos que se cuenta con un soporte gráfico básico que permite representar una escena como una colección de objetos gráficos, cámaras y luces, y que proveerá las operaciones básicas para el manejo de la misma. Dado que este soporte gráfico afecta directamente la implementación de las interacciones de alto nivel del programador definidas

en esta transformación, no se provee una implementación para las mismas.

En esta transformación, denominaremos *CDMV* al conjunto de DMV de entrada y *CDV_i* al conjunto de DV de salida correspondiente a la *i*-ésima rama.

A continuación se detallan las interacciones de alto nivel del programador provistas en esta transformación:

Interacciones de Alto Nivel del Programador en la Transformación de Visualización

GenerarRama(Técnica)

Esta interacción genera un nuevo conjunto de DV a partir del conjunto de DMV. El resultado de esta operación es la ramificación del proceso a partir de esta transformación. La primera vez que se ejecuta el *pipeline*, se creará el primer estado de DV. Esta interacción permite seleccionar una técnica de visualización que soporte el mapeo visual establecido en la *Transformación de Mapeo Visual*.

EliminarRama()

Esta interacción elimina el conjunto de datos *DV_i*, siendo la *i*-ésima rama dónde se aplicó esta interacción. El resultado de esta operación es la eliminación de una rama completa del *pipeline* a partir de esta transformación.

Tabla 5.8: Interacciones de Alto Nivel del Programador en la Transformación de Visualización.

5.3.2.7. Interacciones de Alto Nivel del Programador en los DV

Es el último estado del proceso, pero no el final, ya que constituye el espacio de exploración para el usuario. En este estado, las interacciones de alto nivel del programador dependen fuertemente del soporte gráfico, razón por la cual, no se provee una implementación para las mismas.

Las interacciones de alto nivel del programador provistas en este estado son las siguientes:

Interacciones de Alto Nivel del Programador en los Datos Visualizados

Recuperar(x,y):Dato

Esta interacción permite recuperar de la vista el dato asociado a un objeto gráfico cuya representación incluye la posición (x, y) especificada. Si existe más de un objeto gráfico ocupando la posición especificada, se recuperará el que se encuentre más cercano a la cámara. El mapeo del objeto gráfico al dato representado por el mismo se resuelve en el soporte gráfico.

Recuperar(Área): Colección<Dato>

Esta interacción permite recuperar de la vista una colección de datos asociados a los objetos gráficos cuya representación se encuentra posicionada en la vista dentro del área especificada.

Interacciones de Alto Nivel del Programador en los Datos Visualizados	
Recuperar Objeto Gráfico $(x,y): ObjetoGráfico$	Esta interacción permite recuperar de la vista el objeto gráfico cuya representación ocupa la posición (x,y) especificada. Si existe más de un objeto gráfico ocupando la posición especificada, se recuperará el que se encuentre más cercano a la cámara.
Recuperar Objeto Gráfico (Área) : $Colección< ObjetoGráfico >$	Esta interacción permite recuperar de la vista una colección de objetos gráficos que se encuentran posicionados en la vista en el área especificada.
Rotar $(Colección< ObjetoGráfico >, G_x, G_y, G_z, Eje)$	Esta interacción permite efectuar una rotación de los objetos gráficos especificados. El <i>Eje</i> de rotación y los grados de rotación en x, y y z son especificados.
Trasladar $(Colección< ObjetoGráfico >, \Delta_x, \Delta_y, \Delta_z)$	Esta interacción permite efectuar una traslación de los objetos gráficos especificados. Los mismos se trasladarán en eje x Δ_x unidades, en el eje y Δ_y unidades y en el eje z Δ_z unidades.
Escalar $(Colección< ObjetoGráfico >, Porcentaje_x, Porcentaje_y, Porcentaje_z)$	Esta interacción permite efectuar un escalado de los objetos gráficos especificados. El escalado se realizará según los porcentajes especificados para cada eje.
Distorsionar $(Foco, Distancia, Función_{distor})$	Las técnicas de distorsión preservan un <i>overview</i> de los datos durante el proceso de exploración a través de operaciones <i>drill-down</i> . Se aplica una función de distorsión que deformará la vista a partir de un punto de <i>foco</i> y una distancia <i>especificadas</i> .

Tabla 5.9: Interacciones de Alto Nivel del Programador en los Datos Visualizados.

5.4. Interacciones a Nivel del Usuario

Las interacciones básicas a nivel del usuario están especialmente diseñadas para el usuario del sistema de visualización, que puede no estar familiarizado con los detalles del proceso de visualización. El usuario final de un sistema de visualización interactuará con el mismo a través de la vista, sin conocer necesariamente la representación de los distintos conjuntos de datos ni el rol de los estados y/o las transformaciones a las que se someten los datos a lo largo del proceso. Si bien el usuario final interactuará sobre la vista, estas interacciones pueden resolverse en etapas y/o transformaciones anteriores en el *pipeline* de visualización, re ejecutando el proceso a partir de éstas.

Para definir un conjunto básico de interacciones a nivel del usuario se realizó una recopilación exhaustiva de la bibliografía disponible y se definió un conjunto de interacciones

que consideramos lo suficientemente abarcativas (ver figura 5.5). A continuación, en función de el conjunto de datos que afectan, distinguimos cuatro clases de interacciones a nivel del usuario:

- *Interacciones a Nivel del Usuario Sobre los Conjuntos de Datos.*
- *Interacciones a Nivel del Usuario Sobre el Mapeo Visual.*
- *Interacciones a Nivel del Usuario Sobre la Vista.*
- *Interacciones Compuestas a Nivel del Usuario.*

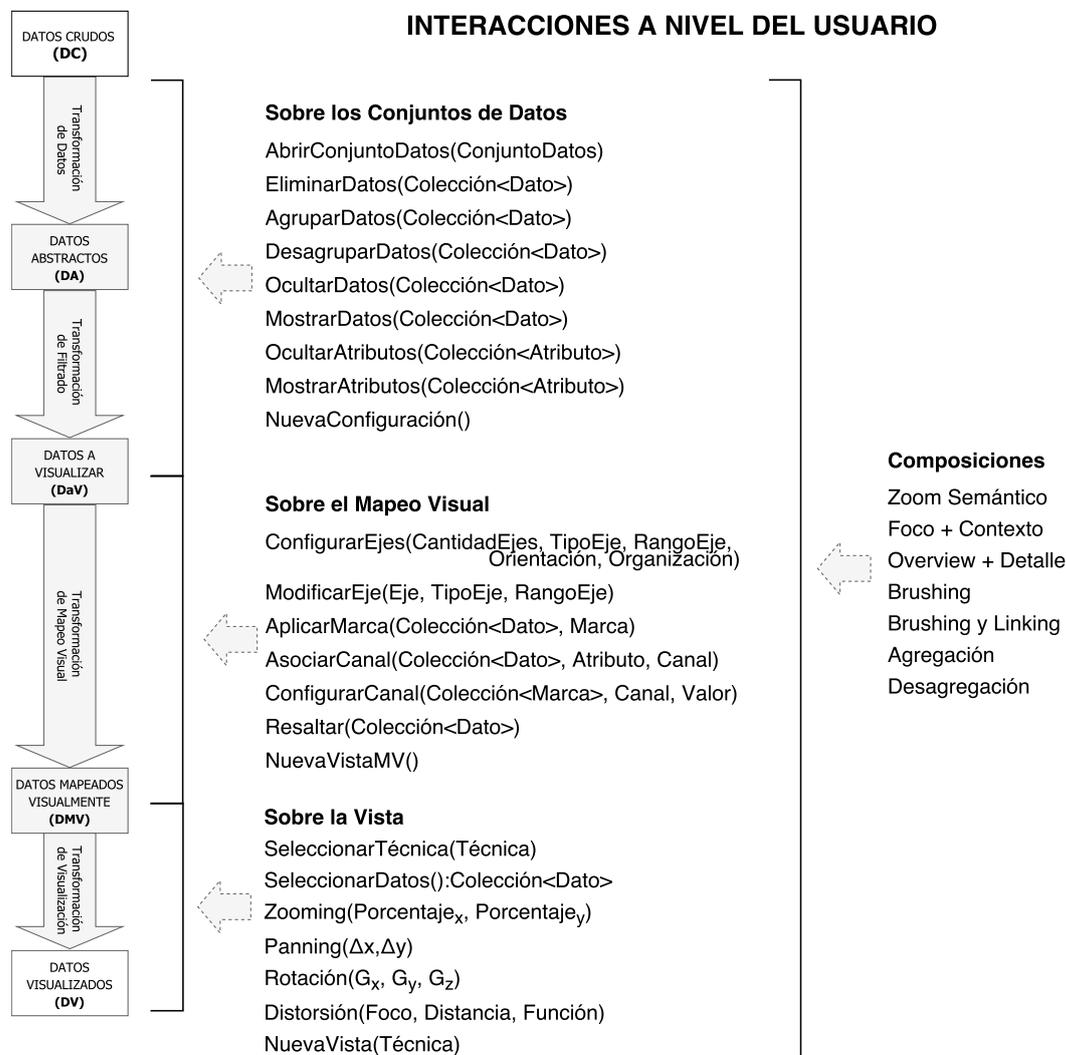


Figura 5.5: Interacciones a Nivel del Usuario.

5.4.1. Interacciones a Nivel del Usuario Sobre los Conjuntos de Datos

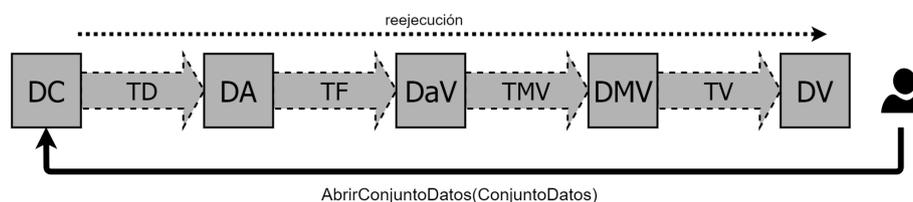
En este grupo se incluyen aquellas interacciones diseñadas para el usuario del sistema de visualización que afecten o manipulen directamente los conjuntos de datos pero no el mapeo visual. En general las interacciones incluidas en este grupo se resuelven en la primera etapa del *pipeline*, que comienza en el estado de *Datos Crudos* y termina en el estado *Datos a Visualizar*.

A continuación se describen las interacciones a nivel del usuario sobre los conjuntos de datos, especificando para las mismas en qué etapa o transformación del *pipeline* se resuelven. Todas las interacciones presentes en este grupo se componen de operadores de valor, ya que modifican el *dataset* subyacente.

Interacciones a Nivel del Usuario Sobre los Conjuntos de Datos

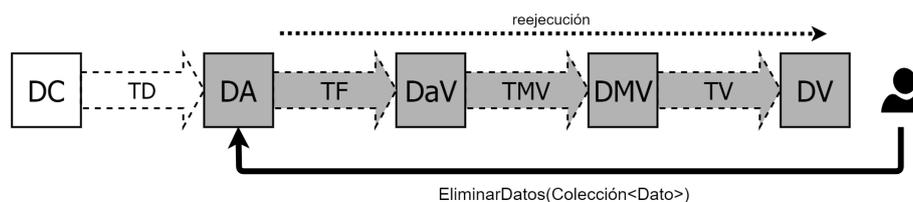
AbrirConjuntoDatos(ConjuntoDatos)

Esta interacción permite al usuario abrir un conjunto de datos e incorporarlo al *pipeline*. Esta interacción se resuelve en el estado *Datos Crudos* del *pipeline*, el cual se re ejecuta a partir del mismo. A nivel del programador, esta interacción de usuario se traduce en *CargarFuenteDatos(ConjuntoDatos)* (interacción de alto nivel del programador en DC).



EliminarDatos(Colección<Dato>)

Esta interacción permite al usuario eliminar la colección de datos especificada. Esta eliminación no puede deshacerse, ya que los datos especificados se eliminan del proceso. Para volver a ingresarlos al mismo se deberán volver a procesar los datos crudos nuevamente. Esta interacción se resuelve en el estado *Datos Abstractos* del *pipeline*.



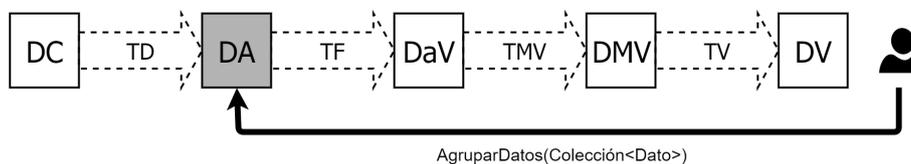
A nivel del programador, esta interacción de usuario se traduce en *EliminarDato(Colección<Dato>)* (interacción de alto nivel del programador en el estado DA).

Sigue en la página siguiente.

Interacciones a Nivel del Usuario Sobre los Conjuntos de Datos

AgruparDatos(Colección<Dato>)

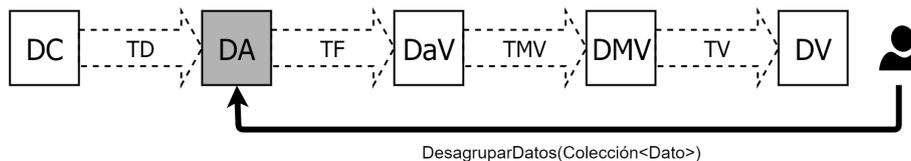
Esta interacción permite al usuario agrupar una colección de datos. Esta agrupación no modifica la vista, sino que agrupa los datos especificados para que futuras interacciones sobre uno de éstos afecte a los demás del grupo.



Esta interacción se resuelve en el estado *Datos Abstractos* del *pipeline*. A nivel del programador, esta interacción de usuario se traduce en **AgruparDatos(Colección<Dato>)** (interacción de alto nivel del programador en el estado **DA**).

DesagruparDatos(Colección<Dato>)

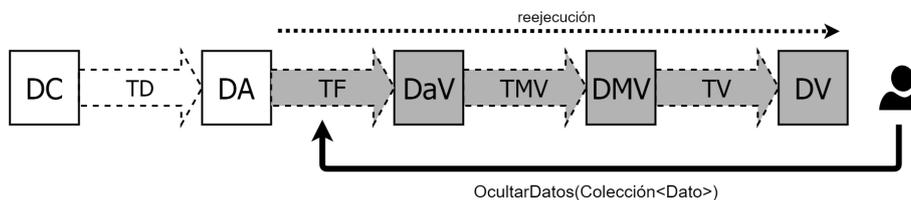
Esta interacción permite al usuario desagrupar una colección de datos previamente agrupada. Al igual que la interacción **AgruparDatos**, esta interacción se resuelve en el estado **DA**.



A nivel del programador, esta interacción de usuario se traduce en **DesagruparDatos(Colección<Dato>)** (interacción de alto nivel del programador en el estado **DA**).

OcultarDatos(Colección<Dato>)

Esta interacción permite al usuario ocultar de la vista la colección de datos especificada. Esta interacción se resuelve en la *Transformación de Filtrado* del *pipeline*.

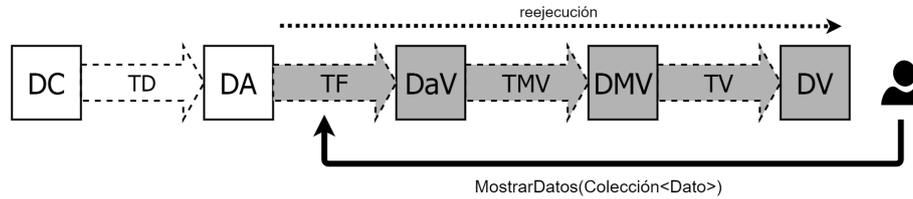


A nivel del programador, esta interacción de usuario se traduce en **FiltrarDatos(Colección<Dato>)** (interacción de alto nivel del programador en la transformación **TF**).

Interacciones a Nivel del Usuario Sobre los Conjuntos de Datos

MostrarDatos(Colección<Dato>)

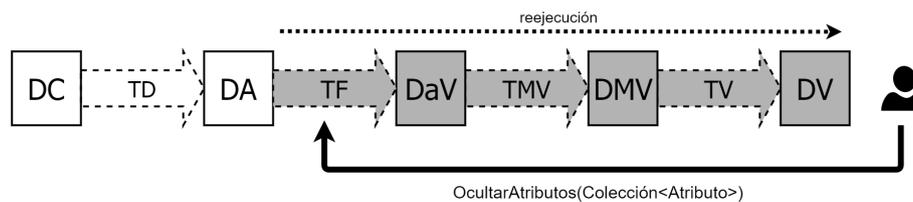
Esta interacción permite al usuario mostrar en la vista la colección de datos especificada. Esta interacción se resuelve en la *Transformación de Filtrado* del *pipeline*.



A nivel del programador, esta interacción de usuario se traduce en *InsertarDatos*(Colección<Dato>) (interacción de alto nivel del programador en la transformación TF), que se encargará de volver a cargar al conjunto de DaV el conjunto de datos que se desea mostrar.

OcultarAtributos(Colección<Atributo>)

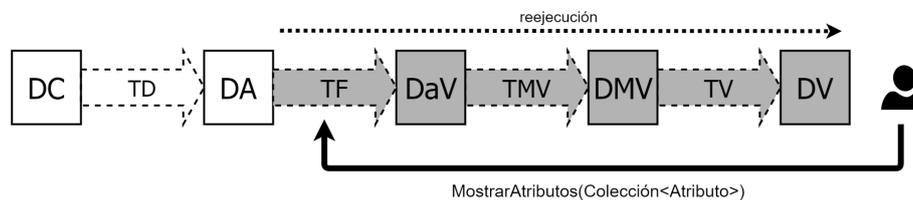
Esta interacción permite al usuario ocultar de la vista la colección de atributos especificada. Esta interacción se resuelve en la *Transformación de Filtrado* del *pipeline*.



A nivel del programador, esta interacción de usuario se traduce en *FiltrarAtributos*(Colección<idAtributo>) (interacción de alto nivel del programador en la transformación TF).

MostrarAtributos(Colección<Atributo>)

Esta interacción permite al usuario mostrar en la vista la colección de atributos especificada. Esta interacción se resuelve en la *Transformación de Filtrado* del *pipeline*.

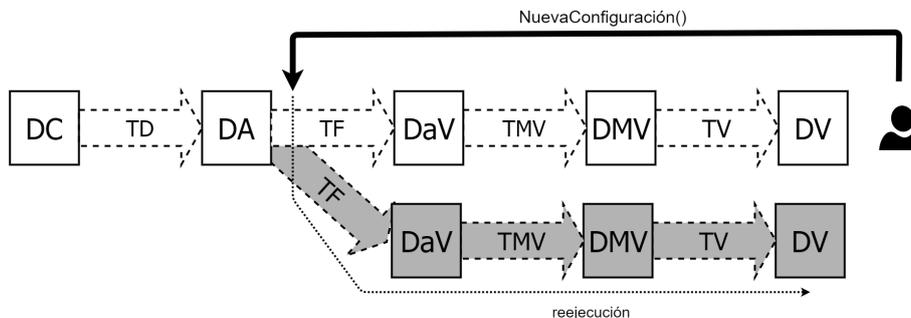


A nivel del programador, esta interacción de usuario se traduce en *InsertarAtributos*(Colección<idAtributo>) (interacción de alto nivel del programador en la transformación TF).

Interacciones a Nivel del Usuario Sobre los Conjuntos de Datos

NuevaConfiguración()

Esta interacción permite obtener una nueva visualización de los datos sin modificar el conjunto de DA del proceso. Esta interacción permite generar una nueva rama en el *pipeline* que compartirá el estado DA con la rama donde se inició la interacción pero podrá mantener una configuración totalmente diferente. Esta interacción se resuelve en la *Transformación de Filtrado* del *pipeline*.



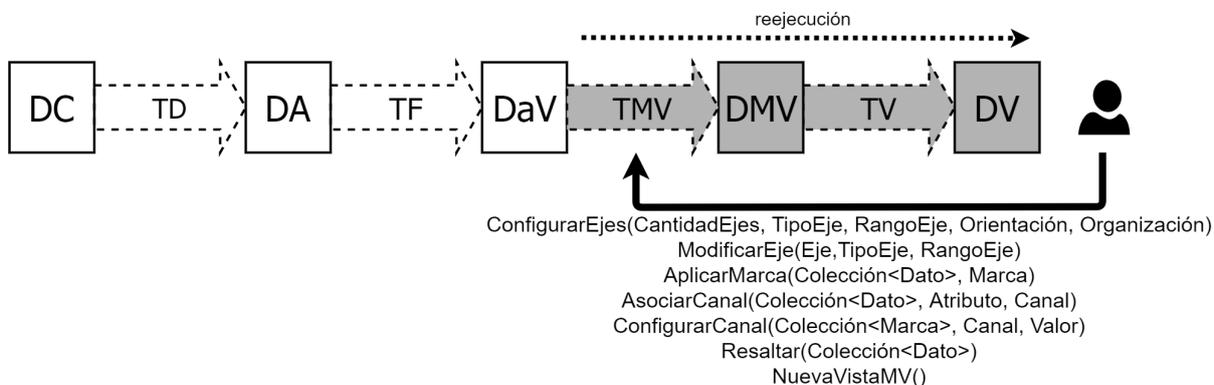
A nivel del programador, esta interacción de usuario se traduce en la interacción *GenerarRama()* (interacción de alto nivel del programador en la transformación TF).

Tabla 5.10: Interacciones a Nivel del Usuario Sobre los Conjuntos de Datos.

5.4.2. Interacciones a Nivel del Usuario Sobre el Mapeo Visual

En esta clase se incluyen aquellas interacciones diseñadas para el usuario del sistema de visualización que afecten o manipulen directamente el mapeo visual.

Todas las interacciones incluidas en este grupo se resuelven en la *Transformación de Mapeo Visual* y se componen de operadores de valor, ya que modifican el *mapeo visual* subyacente.



A continuación se describen las interacciones a nivel del usuario sobre el mapeo visual.

Interacciones a Nivel del Usuario Sobre el Mapeo Visual

ConfigurarEjes(CantidadEjes, TipoEje, RangoEje, Orientación, Organización)

Esta interacción permite al usuario seleccionar la cantidad de ejes que tendrán lugar en la visualización, y el tipo asociado por defecto a los ejes, que luego podrá ser modificado si se desea que la representación tenga ejes de distinto tipo.

A nivel del programador, esta interacción de usuario se traduce en las interacciones *ConfigurarSustratoEspacial(Organización, Orientación)* e *InsertarEje(TipoEje, RangoEje)* (interacciones de alto nivel del programador en la transformación TMV). *InsertarEje* se invocará tantas veces como ejes se desee configurar.

ModificarEje(Eje, TipoEje, RangoEje)

Esta interacción permite al usuario configurar el tipo y el rango que representa un eje en particular.

A nivel del programador, esta interacción de usuario se traduce en la interacción *ConfigurarEje(Eje, TipoEje, RangoEje)* (interacción de alto nivel del programador en la transformación TMV).

AplicarMarca(Colección<Dato>, Marca)

Esta interacción permite al usuario asociar una colección de datos a una marca en particular. A nivel del programador, esta interacción de usuario se traduce en la interacción *AsociarDatosAMarca(Colección<Dato>, Marca)* (interacción de alto nivel del programador en la transformación TMV).

AsociarCanal(Colección<Dato>, Atributo, Canal)

Esta interacción permite al usuario especificar qué atributo de una colección de datos asociar al canal de una marca. A nivel del programador, esta interacción de usuario se traduce en la interacción *AsociarAtributoACanal(Colección<Dato>, Atributo, Canal)* (interacción de alto nivel del programador en la transformación TMV).

ConfigurarCanal(Colección<Marca>, Canal, Valor)

Esta interacción permite al usuario configurar el valor de un canal en una colección de marcas en particular. Las marcas especificadas deben soportar el canal indicado. A nivel del programador, esta interacción de usuario se traduce en la interacción *ConfigurarCanal(Colección<Marca>, Canal, Valor)* (interacción de alto nivel del programador en la transformación TMV).

Resaltar(Colección<Dato>)

Esta interacción permite al usuario resaltar una colección de datos.

A nivel del programador, esta interacción de usuario se traduce en las interacciones del programador que permiten modificar la codificación visual, tales como *ConfigurarCanal* y/o *AsociarDatosAMarca* (interacción de alto nivel del programador en la transformación TMV).

Interacciones a Nivel del Usuario Sobre el Mapeo Visual

NuevaVistaMV()

Esta interacción permite obtener una nueva visualización de los datos aplicando un nuevo mapeo visual pero sin modificar los DaV.

A nivel del programador, esta interacción de usuario se traduce en la interacción *GenerarRama()* (interacción de alto nivel del programador en la transformación TMV).

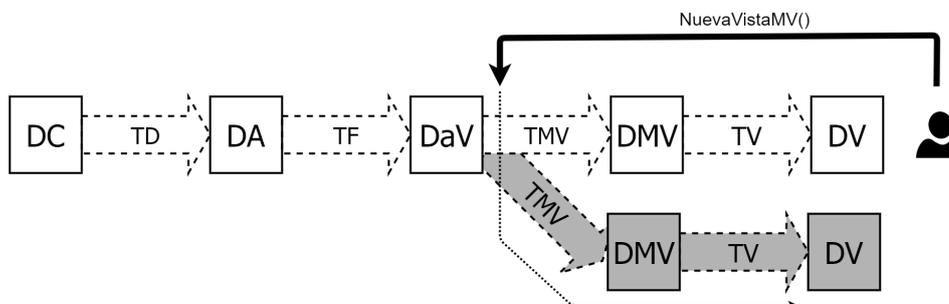
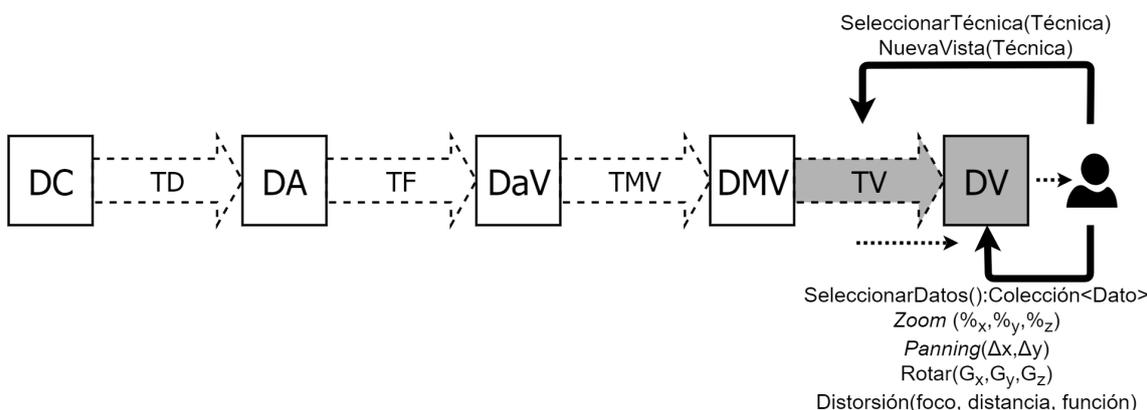


Tabla 5.11: Interacciones a Nivel del Usuario Sobre el Mapeo Visual.

5.4.3. Interacciones a Nivel del Usuario Sobre la Vista

En este grupo se incluyen aquellas interacciones diseñadas para el usuario del sistema de visualización que afecten o manipulen directamente la vista y no necesitan volver a etapas o transformaciones previas a la Transformación de Visualización (TV) para resolverse. Todas las interacciones incluidas en este grupo se resuelven en la *Transformación de Visualización* o en el estado *Datos Visualizados* del *pipeline*.



A continuación se describen las interacciones a nivel del usuario sobre la vista. Todas las interacciones presentes en este grupo se componen de operadores de vista, ya que pueden alterar la apariencia de la vista sin modificar el conjunto de datos subyacente.

Interacciones a Nivel del Usuario Sobre la Vista

SeleccionarTécnica(Técnica)

Esta interacción permite al usuario seleccionar qué técnica de visualización aplicar. Las técnicas disponibles dependerán de las elecciones que el usuario haya realizado en el mapeo visual. Esta interacción se resuelve en la *Transformación de Visualización* del *pipeline*.

A nivel del programador, esta interacción de usuario se traduce en *GenerarRama(Técnica)* (interacción de alto nivel del programador en la transformación TV).

SeleccionarDatos():Colección<Dato>

Esta interacción permite al usuario seleccionar un conjunto de datos. La selección puede efectuarse de forma directa en la vista, y permite recuperar una colección de datos. El usuario, con esta selección de datos, podrá efectuar cualquier otra interacción como resaltado, filtrado, etc. Esta interacción se resuelve en el estado *Datos Visualizados* del *pipeline*.

A nivel del programador, esta interacción de usuario se traduce en *Recuperar(x,y)*, si se especifica una posición en la pantalla, o en *Recuperar(Área)*, si se especifica un área, ambas interacciones de alto nivel del programador en el estado DV.

Zooming(Porcentaje_x,Porcentaje_y)

Esta interacción permite al usuario efectuar un escalado de la escena en el porcentaje especificado. Esta interacción se resuelve en el estado *Datos Visualizados* del *pipeline*.

A nivel del programador, esta interacción de usuario se traduce en *RecuperarObjetoGráfico(Área)*, que permite seleccionar los objetos de la escena que se desea escalar, y *Escalar(Colección<ObjetoGráfico>,Porcentaje_x,Porcentaje_y, 0)*, ambas interacciones de alto nivel del programador en el estado DV.

Panning(Δ_x , Δ_y)

Esta interacción permite efectuar una traslación de la escena. Esta interacción se resuelve en el estado *Datos Visualizados* del *pipeline*.

A nivel del programador, esta interacción de usuario se traduce en *RecuperarObjetoGráfico(Área)*, que permite seleccionar los objetos de la escena que se desea trasladar, y *Trasladar(Colección<ObjetoGráfico>, Δ_x , Δ_y , 0)*, ambas interacciones de alto nivel del programador en el estado DV.

Rotación(G_x , G_y , G_z)

Esta interacción permite rotar la escena. Esta interacción se resuelve en el estado *Datos Visualizados* del *pipeline*.

A nivel del programador, esta interacción de usuario se traduce en *RecuperarObjetoGráfico(Área)*, que permite seleccionar los objetos de la escena que se desea rotar, y *Rotar(Colección<ObjetoGráfico>, G_x , G_y , G_z)*, ambas interacciones de alto nivel del programador en el estado DV.

Interacciones a Nivel del Usuario Sobre la Vista

Distorsión(Foco, Distancia, Función)

Esta interacción permite distorsionar la escena. Esta interacción se resuelve en el estado *Datos Visualizados* del *pipeline*.

A nivel del programador, esta interacción de usuario se traduce en *Distorsionar*(Foco, Distancia, Función), interacción de alto nivel del programador en el estado DV.

Nueva Vista(Técnica)

Esta interacción permite obtener una vista de los datos aplicando una nueva técnica de visualización pero sin modificar el mapeo visual.

A nivel del programador, esta interacción de usuario se traduce en *Generar Rama*(Técnica), interacción de alto nivel del programador en la transformación TV.

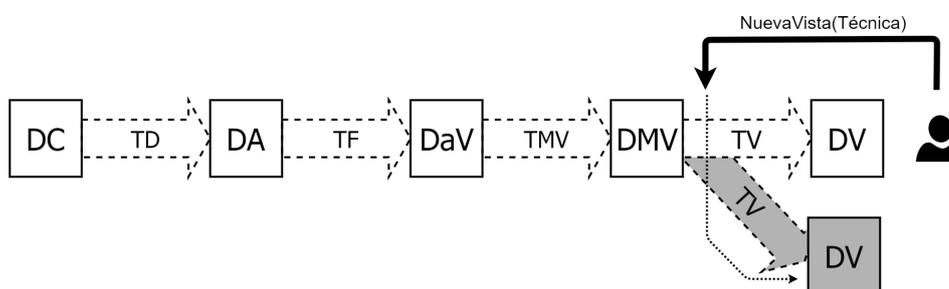


Tabla 5.12: Interacciones a Nivel del Usuario sobre la Vista.

5.4.4. Interacciones Compuestas a Nivel del Usuario

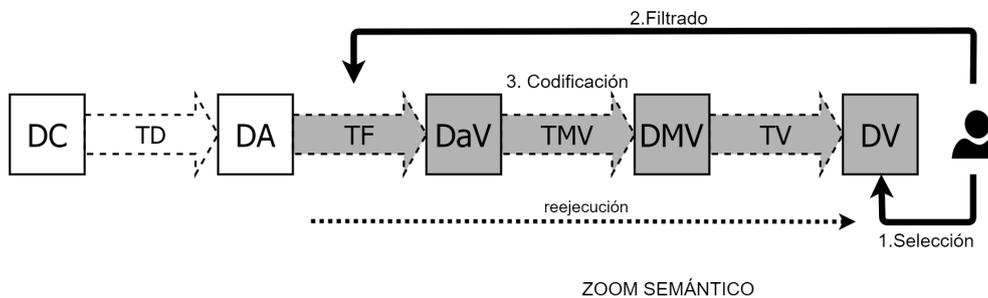
En este grupo incluiremos aquellas interacciones a nivel del usuario que se componen de otras interacciones de usuario básicas. Estas son:

Interacciones Compuestas a Nivel del Usuario

Zoom Semántico

El *zoom semántico* es una técnica de *drill down* que permite revelar información extra por demanda del usuario. Este tipo de técnicas de interacción permite, a través de una selección de usuario, mostrar más información de los datos que los que al momento de efectuar la interacción se muestran en la vista. Dado que esta interacción afecta el mapeo visual, está compuesta por operadores de valor. Esta interacción se compone de las siguientes interacciones a nivel del usuario:

- Selección: Se aplicarán interacciones a nivel del usuario sobre la vista como *SeleccionarDatos* que permitirán seleccionar sobre qué datos se desea aplicar el zoom semántico.
- Filtrado: Se aplicarán interacciones sobre los conjuntos de datos como *MostrarAtributos* que permitirán modificar el conjunto de DaV incorporando aquella información adicional que se quiere mostrar en la vista.
- Codificación: Se aplicarán interacciones sobre el mapeo visual como *AplicarMarca* o *AsociarCanal* que permitirán especificar cómo se desea que se muestre la nueva información.



Foco + Contexto

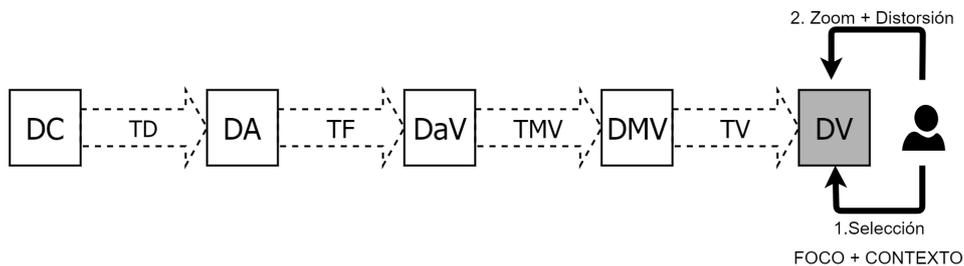
Las técnicas de *Foco + Contexto* permiten resaltar zonas o datos de interés, en una única vista, efectuando una distorsión en la misma. En general, se aplica un lente de distorsión sobre la zona de interés, lo que permite visualizarla en mayor detalle sin perder el contexto general de la misma. Dado que esta interacción no afecta los conjuntos de datos subyacentes, está compuesta por operadores de vista.

Sigue en la página siguiente.

Interacciones Compuestas a Nivel del Usuario

Para esta alternativa, las interacciones a nivel del usuario que componen esta interacción son las siguientes.

- Selección: Se aplicarán interacciones a nivel del usuario sobre la vista como *SeleccionarDatos* que permitirán seleccionar sobre que sector de la vista se desea obtener más detalle.
- Zooming: Se aplicarán interacciones de escalado sobre la vista (*Zooming*).
- Distorsión: Se aplicarán interacciones de distorsión sobre la vista (*Distorsión*).



Overview + Detalle

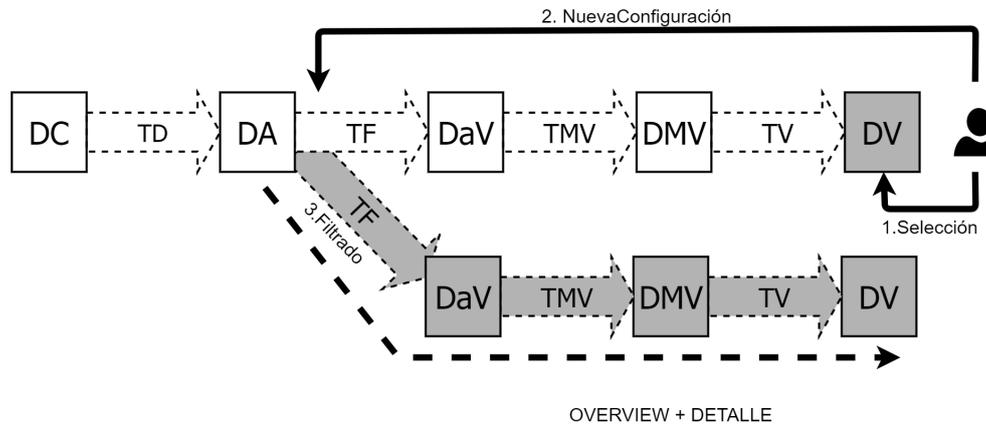
Estas técnicas de interacción permiten al usuario tener un *overview* de todos los datos y detalle de un subconjunto de interés. Involucra dos vistas de los datos, una general que provee el *overview* y otra vista que provee el detalle. El usuario invoca esta interacción efectuando una selección sobre la vista que provee el *overview* del sector que desea ver en más detalle. Dado que esta interacción ramifica el *pipeline* y afecta los conjuntos de datos subyacentes, está compuesta por operadores de valor.

Las interacciones involucradas en el Overview + Contexto son:

- Selección: Se aplicarán interacciones a nivel del usuario sobre la vista que representa el *overview* para seleccionar sobre que sector de la vista se desea obtener más detalle (*SeleccionarDatos*).
- Nueva Visualización: Se crea una nueva visualización para la selección efectuada por el usuario. Para esto, se invoca la interacción de usuario sobre los conjuntos de datos *Nueva-Configuración*.
- Filtrado: Se aplicarán interacciones sobre los conjuntos de dato como *MostrarDatos* y/o *OcultarDatos* que permitirán modificar el conjunto de DaV asegurándonos de solo incorporar la selección inicial efectuada por el usuario.

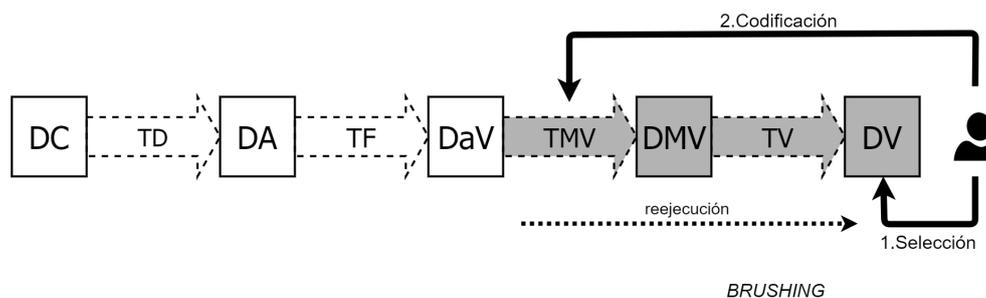
Sigue en la página siguiente.

Interacciones Compuestas a Nivel del Usuario

***Brushing***

La técnica de *brushing* es el proceso de seleccionar datos de la vista interactivamente con la intención de resaltarlos de alguna manera. Dado que esta interacción afecta el mapeo visual, está compuesta por operadores de valor. Esta interacción se compone de las siguientes interacciones a nivel del usuario:

- Selección: Se aplicará la interacción de usuario sobre la vista *SeleccionarDatos* que permitirán seleccionar los datos ubicados en un sector específico de la vista.
- Codificación: Se aplicarán interacciones sobre el mapeo visual resaltando los datos seleccionados.

***Brushing y Linking***

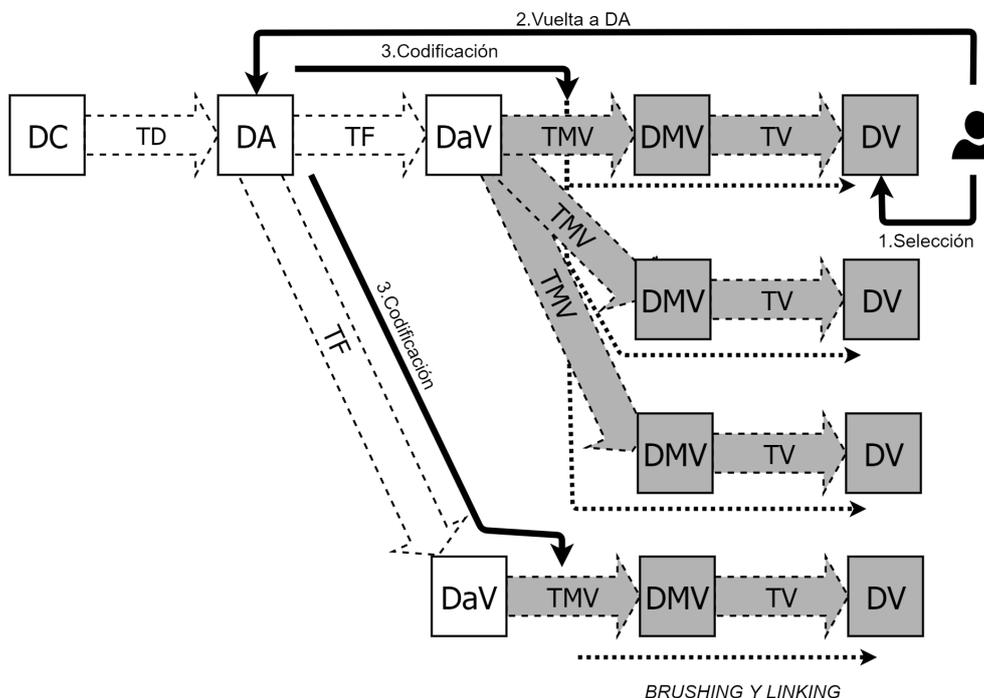
Se aplica a múltiples vistas, permitiendo seleccionar un conjunto de datos en una vista y resaltarlo en todas las demás vistas coordinadas. Los cambios generados interactivamente en una vista se reflejan automáticamente en las vistas restantes. Conectar múltiples visualizaciones a través de esta técnica suele proveer más información que al tratar cada visualización de forma independiente.

Sigue en la página siguiente.

Interacciones Compuestas a Nivel del Usuario

En el MUV, todas las vistas resultantes de aquellas ramas que compartan el estado *Datos Crudos* estarán coordinadas, ya que se estará visualizando, con diferentes configuraciones, una parte o la totalidad del mismo conjunto de datos. Por consiguiente, al aplicar la interacción *Brushing y Linking*, primero se deberá efectuar una selección en la vista, y luego se deberá volver en el estado DA, y para cada una de las ramificaciones del proceso que partan de ese estado, se deberá aplicar la codificación visual correspondiente en la *Transformación de Mapeo Visual* para que la selección se haga efectiva en todas las vistas relacionadas. Esta interacción se resuelve entonces en el estado DA. Dado que esta interacción afecta el mapeo visual, está compuesta por operadores de valor. Esta interacción se compone de las siguientes interacciones a nivel del usuario:

- Selección: Se aplicará la interacción de usuario sobre una vista *SeleccionarDatos* que permitirán seleccionar los datos ubicados en un sector específico de la vista.
- Retornar al estado DA para detectar todas las ramas que compartan este estado.
- Codificación: En todas las ramas del *pipeline* que compartan el estado DA se aplicarán interacciones sobre el mapeo visual resaltando los datos seleccionados.



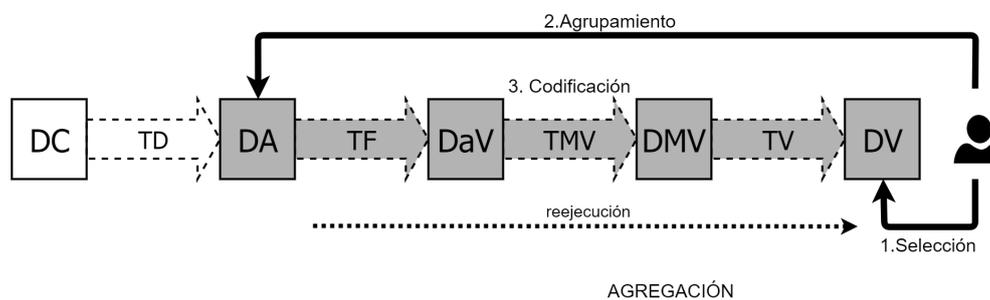
Agregación

Permite seleccionar un conjunto de datos, agruparlos, y representarlos con una única marca. Dado que esta interacción requiere agrupar los datos, se resuelve en el estado DA. Esta interacción afecta tanto los *datasets* como el mapeo visual, de modo que está compuesta por operadores de valor.

Interacciones Compuestas a Nivel del Usuario

La interacción *Agregación* se compone de las siguientes interacciones a nivel del usuario:

- Selección: Se aplicará la interacción de usuario sobre una vista *SeleccionarDatos* que permitirán seleccionar los datos ubicados en un sector específico de la vista.
- Agrupamiento: Se retornar al estado DA para efectuar un agrupamiento de los datos seleccionados aplicando la interacción de usuario sobre los conjuntos de datos *AgruparDatos*.
- Codificación: Se modificará el mapeo visual asociando a todos los datos agrupados la misma marca. Para esto se aplicarán interacciones de usuario sobre el mapeo visual.



Desagregación

Permite seleccionar un objeto de la vista que represente un conjunto de datos agregados y desagregarlos. El efecto de esta interacción es desagrupar los datos previamente agrupados y representar cada uno de ellos individualmente. Es claro que esta interacción podrá aplicarse únicamente sobre objetos de la vista que representen un conjunto de datos agregados previamente. Dado que esta interacción requiere desagrupar los datos, se resuelve en el estado DA. Esta interacción afecta tanto los *datasets* como el mapeo visual, de modo que está compuesta por operadores de valor.

La interacción *Desagregación* se compone de las siguientes interacciones a nivel del usuario:

- Selección: Se aplicará la interacción de usuario sobre una vista *SeleccionarDatos* que permitirá seleccionar el objeto ubicado en la posición especificada de la vista.
- Desagrupamiento: Se retornar al estado DA efectuar una desagrupamiento de los datos seleccionados aplicando la interacción de usuario sobre los conjuntos de datos *DesagruparDatos*.
- Codificación: Se modificará el mapeo visual asociando a todos los datos desagrupados una marca individual. Para esto se aplicarán interacciones de usuario sobre el mapeo visual.

Sigue en la página siguiente.

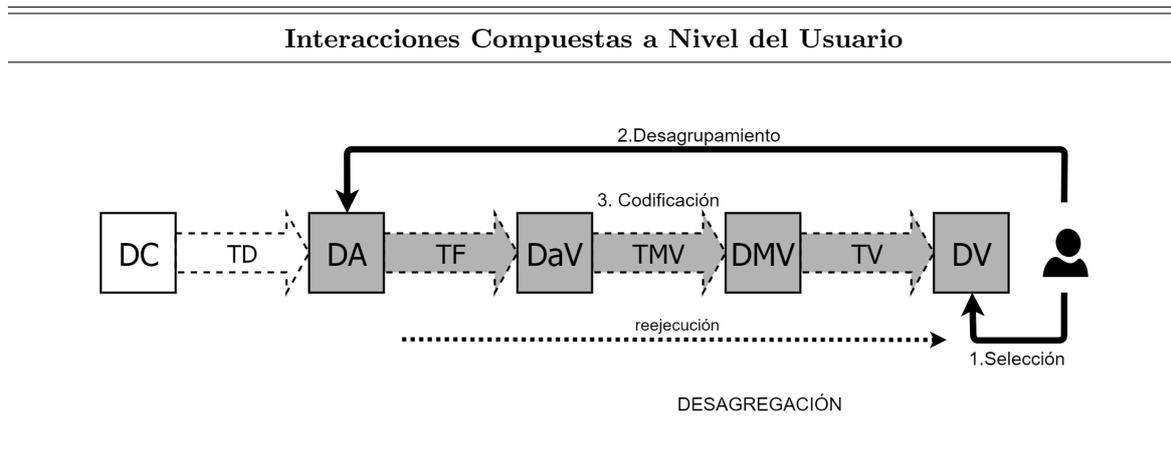


Tabla 5.13: Interacciones Compuestas a Nivel del Usuario.

5.5. Conclusión

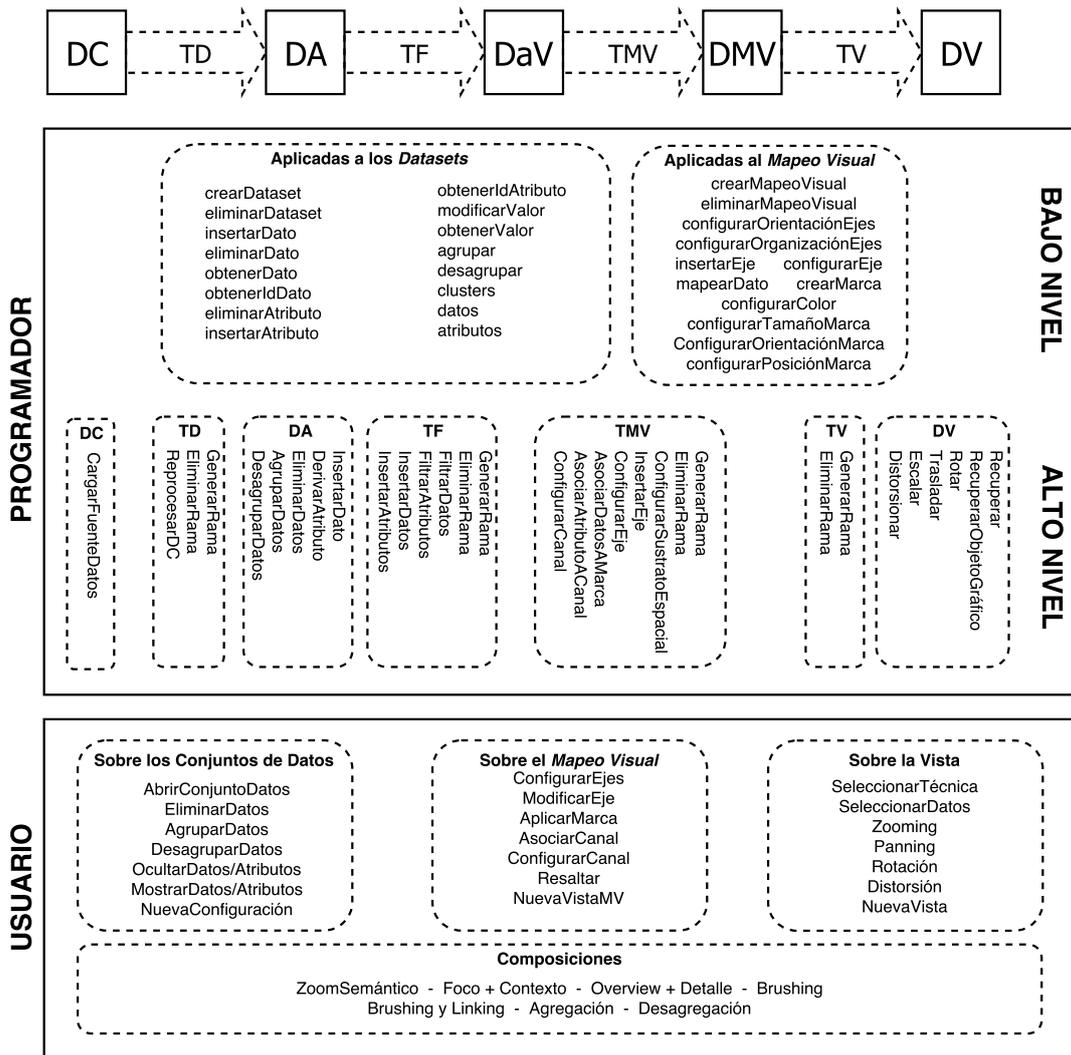
En este capítulo definimos una clasificación de interacciones para el Modelo Unificado de Visualización (MUV) (ver figura 5.6). Encontramos en el MUV un cuerpo troncal de interacciones operacionalmente similares entre los distintos dominios de aplicación contemplando las características distintivas de los distintos dominios y/o técnicas particulares, constituyendo un modelo conceptual de interacción.

La clasificación presentada se compone de dos clases principales, en función del usuario al que están destinadas: las interacciones a nivel del programador y las interacciones a nivel del usuario. Las interacciones a nivel de programador, a su vez, se clasifican en dos grupos: las interacciones de bajo nivel del programador y las interacciones de alto nivel del programador. Las interacciones de bajo nivel del programador se definieron en función de las representaciones para los conjuntos de datos y para el *mapeo visual* presentadas en el capítulo 4. Las interacciones de alto nivel del programador se definieron en función de las interacciones de bajo nivel del programador.

Las interacciones a nivel del usuario se clasificaron en 4 clases principales, en función de la etapa del proceso de visualización que afectan: las interacciones a nivel del usuario sobre los conjuntos de datos, sobre el mapeo visual, sobre la vista y composiciones de estas tres. Las interacciones a nivel del usuario se definieron en base de las interacciones a nivel del programador, especificando para cada una en que estado o transformación del proceso de visualización se resuelven.

En el capítulo siguiente, se instanciará esta clasificación con dos casos de estudio de

un dominio en particular.



CLASIFICACIÓN DE INTERACCIONES EN EL CONTEXTO DEL MUV

Figura 5.6: Clasificación de Interacciones en el Contexto del MUV.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 6

Casos de Estudio

“Any powerful idea is absolutely fascinating and useless until we choose to use it.”

— Richard Bach

En el marco de esta tesis se han desarrollado dos prototipos de visualización que sirvieron de base para el estudio y exploración de los conceptos introducidos, el *SpinelViz* [GCF⁺12] y el *Spinel Explorer* [GFG⁺14, GGF⁺15, GFG⁺17]. Ambos prototipos consisten en herramientas de visualización de datos geológicos, en particular de un conjunto de minerales denominados espinelos.

Tanto el *SpinelViz* como el *Spinel Explorer* fueron desarrollados en colaboración con un equipo interdisciplinario integrado por miembros del VyGLab (Laboratorio de Investigación en Visualización y Computación Gráfica) del Departamento de Ciencias e Ingeniería de la Computación (Universidad Nacional del Sur, Bahía Blanca) y del INGEOSUR (Instituto Geológico del Sur, UNS-CONICET, Bahía Blanca). En el *Spinel Explorer*, en particular, también se trabajó en colaboración con el Dr. Eduard Gröller, del Institute of Computer Graphics and Algorithms de la TUWien (Universidad Técnica de Viena) y el Dr. Krešimir Matković, del VRVis Center de Viena, Austria.

En este capítulo describiremos el *SpinelViz* [GCF⁺12] y el *Spinel Explorer* [GFG⁺14, GGF⁺15, GFG⁺17] situándolos en el marco del Modelo Unificado de Visualización (MUV), describiremos su funcionalidad y las interacciones provistas que fueron generadas *ad hoc* detallando, para cada una de éstas, en qué etapa del pipeline se resuelven y su implementación en términos de la clasificación de interacciones para visualización presentadas en

el capítulo 5.

6.1. Introducción

Los conjuntos de datos originados desde las Ciencias Naturales, y en particular desde las Ciencias Geológicas, son cada vez más grandes, involucrando la aplicación de una gran variedad de herramientas de visualización para su análisis y exploración; tal es el caso de conjuntos de datos topográficos, proyecciones cartográficas, datos geofísicos, etc., que requieren un soporte visual adecuado para su exploración.

En las Ciencias Geológicas, uno de los objetivos de los trabajos desarrollados por los geólogos es tratar de reconstruir los ambientes geológicos donde se formaron las rocas y los eventos que posteriormente las afectaron. Esto permite entender la evolución geológica de la Tierra y, por ende, identificar regiones donde se localizan recursos naturales (minerales, combustibles, agua), identificar áreas que presentan riesgos geológicos para la vida humana o para la implantación de obras de ingeniería, etc. Para esto los geólogos realizan tanto trabajos de campo como de laboratorio y de gabinete donde analizan los posibles vínculos existentes entre lo observado en el campo, los datos allí obtenidos, la información aportada por las muestras recolectadas y la brindada por otras disciplinas estrechamente vinculadas, tales como la geoquímica y la isotopía. Tanto el trabajo de campo como el realizado en el laboratorio implican grandes desafíos en lo que respecta al procesamiento y análisis de los datos que se obtienen en cada uno de estos contextos. Debido a las características de los datos que se manejan, no hay una única metáfora visual que pueda satisfacer todas las necesidades analíticas. Por esta razón, resulta imperioso contar con un conjunto de metáforas visuales y de técnicas de visualización asociadas. Estas representaciones deben poder comunicar efectivamente las características del espacio de información así como incentivar el descubrimiento.

En el esfuerzo por reconstruir los ambientes geológicos, los geólogos trabajan usualmente con rocas que tienen edades oscilando entre pocos a miles de millones de años. Para alcanzar este objetivo, se debe procesar y analizar una gran cantidad de información obtenida de rocas y minerales. Para esto, una de las herramientas más adecuadas y confiables es el estudio de los minerales que integran el grupo de los espinelos. Este grupo de minerales resulta un excelente candidato a ser explorado y visualizado ya que es

considerado un importante “indicador petrogenético”, no sólo porque brinda información vital en lo referido al ambiente tectónico de las rocas presentes en un área determinada en relación a ambientes tectónicos definidos a nivel global, sino también porque provee información valiosa acerca de los procesos que sufrieron las rocas que contienen a esos minerales [Irv65, BLoA91, Roe94, BR01].

Para cada muestra de espinelo se analizan 11 elementos químicos mayoritarios (aquéllos cuyo contenido es superior al 0,01 % en peso) y a partir de éstos se obtienen 22 metadatos, denominados miembros finales, que son combinaciones de elementos químicos mayoritarios. Sin embargo, sólo 8 de estos miembros finales son usados habitualmente por los geólogos para su representación en gráficos específicos. Resulta frecuente representar los espinelos en un ambiente prismático denominado Prisma de Espinelos. En este prisma se representa cada análisis químico obtenido. Adicionalmente, el prisma puede dividirse en distintos campos que representan los distintos ambientes tectónicos. Los análisis que corresponden a espinelos provenientes de un determinado ambiente tectónico se agrupan en un patrón de referencia determinado y único y se agrupan en una determinada región del prisma [BLoA91] [CDD⁺97]. Para evaluar las correlaciones entre los elementos geoquímicos se analizan tanto los diagramas prismáticos 3D como la información proyectada sobre las caras del prisma.

En 2001, Barnes y Roeder [BR01] compilaron una base de datos correspondiente a más de 26000 análisis de espinelos de rocas ígneas y metamórficas de todo el mundo, que les permitió caracterizar cerca de 40 ambientes tectónicos posibles. Para cada uno de estos ambientes tectónicos se extrajeron campos composicionales característicos y se construyeron gráficos con contornos, que suelen ser utilizados por los geólogos para estimar el ambiente tectónico donde una muestra de espinelo podría haberse formado. Para determinarlos, se digitalizan los gráficos de los contornos de Barnes y Roeder para luego poder compararlos manualmente con los diagramas generados previamente. Esta comparación es exhaustiva y debe realizarse para cada uno de los cerca de 40 ambientes tectónicos y en cada ambiente para cada una de las proyecciones sobre las caras del prisma. Es claro que tanto la digitalización de los gráficos con los contornos como la comparación exhaustiva son tareas propensas a errores y sumamente tediosas y por lo tanto, excelentes candidatas a ser automatizadas.

También es importante notar que el análisis simultáneo de todos los miembros finales

y de los datos correspondientes a elementos mayoritarios obtenidos mediante análisis geoquímicos, no se lleva a cabo en el proceso de desarrollo de modelos petrogenéticos debido a la alta dimensionalidad de estos datos. En este contexto se considera relevante el diseño y el desarrollo de métodos de visualización de datos altamente dimensionales apropiados que permitan al geólogo considerar todos los datos obtenidos en los análisis.

Debe señalarse que las técnicas desarrolladas para analizar datos multidimensionales también podrían aplicarse para la visualización de datos químicos requeridos por otras disciplinas tales como medicina, biología, química, ciencias agrarias y ciencias de los materiales en general y a los datos obtenidos por los geofísicos (magnéticos, gravimétricos, sismológicos, geoelectricos, etc).

En este marco, se diseñaron dos prototipos de visualización de minerales del grupo de los espinelos. En 2012 se publicó el *SpinelViz* [GCF⁺12], herramienta de visualización de datos geológicos en 3 dimensiones que permite al usuario observar, explorar e interactuar con el Prisma de los Espinelos, brindando la posibilidad de cargar diferentes conjuntos de datos e interactuar con ellos en simultáneo. Más tarde, en 2014, se presentó el *Spinel Explorer* que consiste en un entorno unificado de análisis visual para datos geológicos que hace uso de múltiples vistas coordinadas y está orientado no solo a la exploración de los espinelos sino también a la categorización semi-automática de los mismos en el ambiente geológico al que pertenecen.

6.2. Campo de Aplicación

Como se mencionó anteriormente, los espinelos conforman una solución sólida con 22 miembros finales [GSF⁺97], de los cuales 8 son especialmente utilizados en el análisis de estos minerales. Estos 8 miembros finales se componen de dos grupos solapados de 6 miembros, que se utilizan para la representación de estos minerales en espacios prismáticos. Estos miembros finales se calculan en función de los elementos químicos mayoritarios de cada muestra y cada vértice del prisma se corresponde con un miembro final. El Prisma de los Espinelos soporta dos representaciones, el *Prisma de la Magnetita* y el *Prisma Reducido Ulvöespinelo*. Los vértices del Prisma de la Magnetita se corresponden con los miembros finales: Magnetita (Fe_3O_4), Magnesioferrita ($MgFe_2O_4$), Espinelo ($MgAl_2O_4$), Hercinita ($FeAl_2O_4$), Cromita ($FeCr_2O_4$) y Magnesiocromita ($MgCr_2O_4$). El Prisma Re-

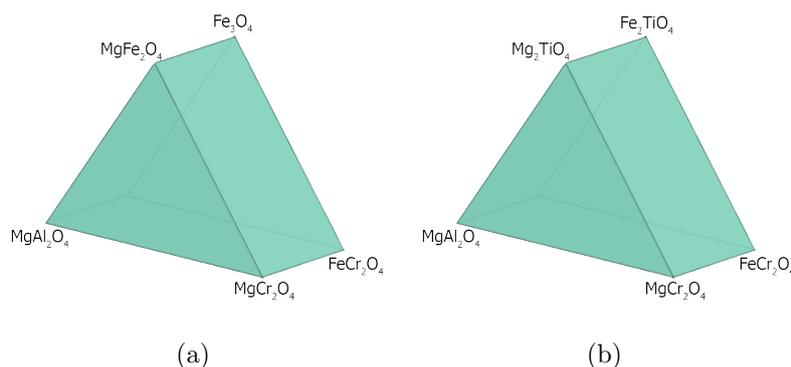


Figura 6.1: Prismas de los Espinelos: (a) Prisma de la Magnetita y (b) Prisma Reducido (Ülvöspinel).

ducido Ulvöspinel está compuesto en su base por los mismos 4 miembros finales que el Prisma de la Magnetita, incorporando Qandilita (Mg_2TiO_4) y Ulvöspinel (Fe_2TiO_4) en sus vértices superiores. La figura 6.1 ilustra ambas representaciones. Estos espacios prismáticos constituyen el espacio natural de exploración de los geólogos.

Las muestras de espinelos que provienen de un determinado ambiente geológico se agrupan en un patrón de referencia que caracteriza dicho ambiente. Un conjunto de muestras correspondiente a rocas de un ambiente tectónico en particular establece una referencia (tendencia) dentro del espacio composicional. En la figura 6.2(a) se puede observar la representación de un determinado patrón de referencia o campo composicional.

Una muestra en particular se podría representar en el espacio prismático por medio de un símbolo adecuado (ver Figura 6.2(b)) y la clasificación de dicha muestra en función de un patrón de referencia consistiría simplemente en determinar si dicho símbolo reside dentro o fuera del volumen delimitado por el patrón en cuestión.

Dado que estos espacios prismáticos son difíciles de graficar sin *software* dedicado, los geólogos acostumbran a utilizar diagramas en dos dimensiones, tales como diagramas triangulares y scatterplots para representar las proyecciones de las caras del prisma. En la figura 6.3 se muestra el espacio prismático según la bibliografía y las proyecciones sobre distintas caras del mismo. Las proyecciones se logran con un *software* de diagramado 2D pero, hasta la presentación de los prototipos aquí descritos, no se disponía de herramientas 3D específicas que permitieran visualizar y explorar el espacio en tres dimensiones.

En 2001, Barnes y Roeder [BR01] compilaron una base de datos de aproximadamente

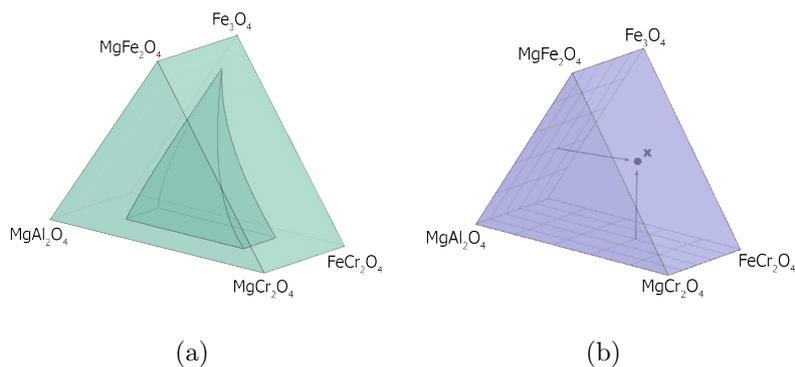


Figura 6.2: (a) Patrón de referencia establecido para Basaltos [Hag76] y (b) una única muestra de espinelo ploteada en el Prisma de los Espinelos.

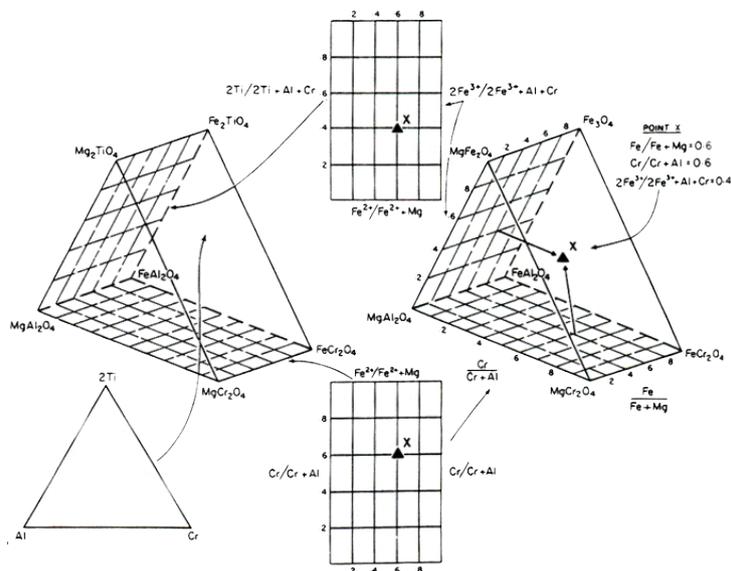


Figura 6.3: El Espacio Prismático y sus Proyecciones [BLoA91].

26000 espinelos de rocas ígneas y metamórficas. A partir de esta base de datos extrajeron y definieron un conjunto de campos composicionales para los minerales del grupo de los espinelos. Clasificaron todos las muestras de espinelos en una jerarquía de campos composicionales y para cada campo definieron un conjunto de contornos característico sobre las proyecciones del espacio prismático. Los contornos fueron generados para los percentiles 50 y 90, que envuelven respectivamente el 50% y el 90% de todas las muestras recopiladas para un campo composicional en particular. En la figura 6.4 se ilustra un ejemplo de la colección de contornos de percentil 50 y de percentil 90 de Barnes y Roeder visualizado en 3 proyecciones representativas del Prisma de la Magnetita. Los contornos de Barnes y Roeder [BR01] son comúnmente utilizados para clasificar muestras de espinelos, es de-

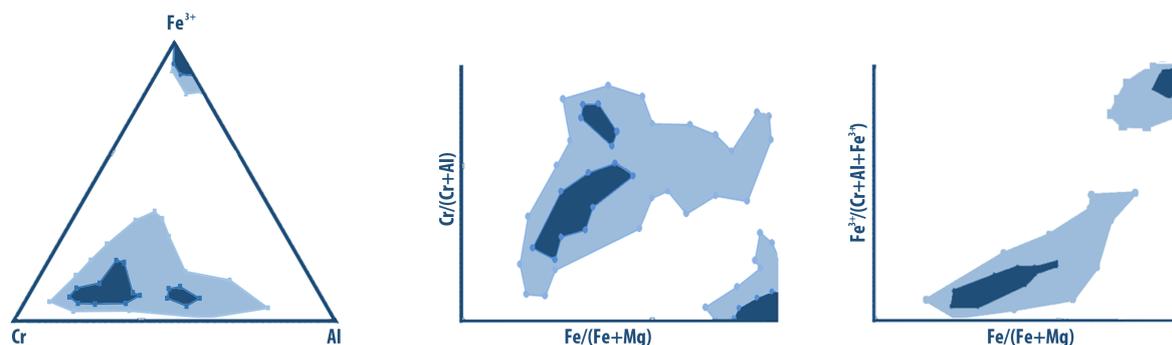


Figura 6.4: Contornos de Barnes y Roeder para el Campo Composicional de los Basaltos. En el ejemplo, el contorno está diagramado en las 3 proyecciones más representativas del Prisma de la Magnetita [GGF⁺15]. Los contornos coloreados de azul claro corresponden a los contornos de percentil 50 y los coloreados de azul oscuro corresponden a los de percentil 90.

Para que si una muestra de espinelo corresponde a un campo composicional, entonces al ubicarla en el espacio prismático, y proyectarla sobre sus caras, debe posicionarse dentro del contorno correspondiente. Este proceso tradicionalmente involucra una comparación manual exhaustiva de las proyecciones en el Prisma de los Espinelos de los datos a categorizar con los contornos de Barnes y Roeder. El objetivo de esta comparación es encontrar aquellos contornos que contengan (o contengan parcialmente) los puntos correspondientes al conjunto de datos a categorizar.

La gran cantidad de datos a visualizar, la complejidad del espacio involucrado y el extenso conjunto de contornos disponible transforman el análisis de los minerales del grupo de los espinelos en una tarea sumamente tediosa y propensa a errores para ser realizada manualmente. Por otro lado, el diagramado en distintos planos que podría llevarse a cabo en distintas vistas 2D, deja a cargo del usuario la integración en el espacio prismático 3D. Esto sugiere que un prisma 3D con el que se pueda interactuar adecuadamente es una alternativa invaluable.

6.2.1. Flujo de Trabajo Tradicional de Análisis de Minerales del Grupo de los Espinelos

En la figura 6.5 se ilustra el flujo de trabajo tradicional de análisis de minerales del grupo de los espinelos. El primer paso consiste en la recolección del conjunto de

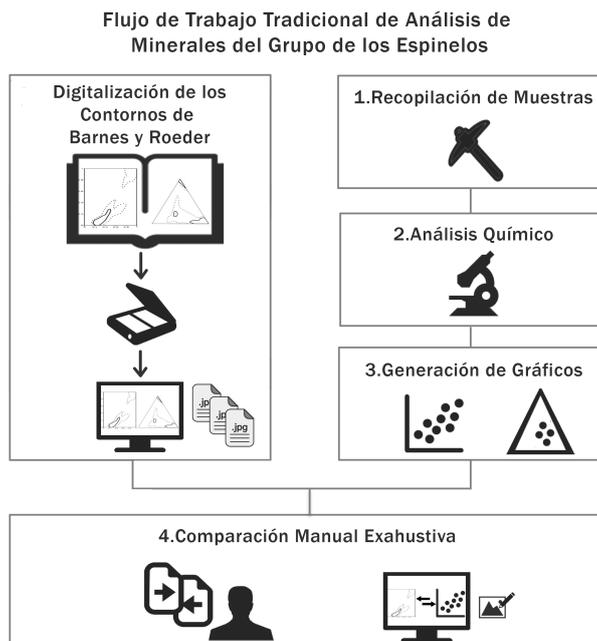


Figura 6.5: Flujo de Trabajo Tradicional de Análisis de Minerales del Grupo de los Espinelos.

muestras a analizar. En esta etapa los geólogos toman *in situ*¹ muestras que consideran representativas al problema que desean resolver. Las muestras, una vez recolectadas, son analizadas químicamente mediante un microscopio petrográfico con el fin de obtener la composición química de los minerales de interés. La composición química de una muestra está conformada por su composición en términos de los *elementos mayoritarios* (Silicio (*Si*), Aluminio (*Al*), Hierro (*Fe*), Calcio (*Ca*), Sodio (*Na*), Magnesio (*Mg*), etc.), de los *óxidos* de los *elementos mayoritarios* (Sílice (SiO_2), Alúmina (Al_2O_3), Óxido Férrico (Fe_2O_3), Cal (CaO), Óxido de Magnesio (MgO), etc.) y de los *miembros finales* (*Magnetita* (Fe_3O_4), *Magnesioferrita* ($MgFe_2O_4$), *Espinelo* ($MgAl_2O_4$), *Hercinita* ($FeAl_2O_4$), *Cromita* ($FeCr_2O_4$), *Magnesiocromita* ($MgCr_2O_4$), etc.). En adición a la composición química, a cada muestra se le asigna un identificador con el objetivo de poder diferenciarla durante el proceso de exploración.

En el tercer paso, y a partir de los análisis químicos obtenidos en el paso anterior, los expertos construyen diferentes diagramas de dispersión (*scatteplots*) y diagramas triangulares (*triplots*) que representan las proyecciones del Prisma de los Espinelos. Éstos les permiten estudiar las relaciones entre los elementos químicos que constituyen cada mineral de espinelo. En estos diagramas, los geólogos también pueden discriminar las

¹*In situ*: en el sitio o en el lugar.

variaciones en el conjunto de puntos y reconocer los grupos químicos y subgrupos. Eventualmente, después de que los diagramas han sido creados, se comparan con los contornos de Barnes y Roeder, que en general están disponibles como copias impresas o imágenes escaneadas.

El último paso del flujo de trabajo tradicional consiste en una comparación manual exhaustiva de todos los diagramas de contorno de Barnes y Roeder con los gráficos generados en el tercer paso. Los geólogos, o bien colocan sus gráficos al lado de los contornos de Barnes y Roeder o bien los superponen con los mismos. Éste se considera el método más avanzado en el campo. Finalmente, los geólogos deciden (visualmente) cuáles de los contornos digitalizados se corresponde mejor con sus propios datos. El que mejor se ajuste corresponderá al campo composicional más probable para su conjunto de datos. Si el conjunto de datos coincide con más de un campo composicional, los geólogos deben apelar a sus conocimientos sobre el contexto geológico del área de estudio para seleccionar el campo composicional más adecuado.

Por lo general, los geólogos necesitan comparar manualmente los contornos de Barnes y Roeder para todos los ambientes tectónicos (y para todas las proyecciones del prisma) con los gráficos generados en el tercer paso, tarea que puede tomar varios días.

6.3. Trabajo Relacionado

Se han desarrollado varios proyectos con el fin de proveer herramientas que permitan visualizar datos provenientes de las Ciencias Geológicas en general ([BGB⁺00], [MRY⁺04], [AMB⁺03], etc.), sin embargo, en lo que al grupo de minerales de los espinelos se refiere, no hay herramientas específicas de visualización. En 1991 Williams *et al.* publicaron un paquete integral de 10 aplicaciones Macintosh para la producción de información referente a petrología ígnea. Este paquete incluía herramientas para calcular diagramas ternarios, diagramas multi-elemento, cálculos isotópicos, etc. Además, contenía dos programas llamados SPINEL y SPINELTAB, cuya funcionalidad consistía en el diagramado de análisis de espinelos en los prismas composicionales Prisma de la Magnetita Oxidizado y Prisma Reducido (Ulvöespinelo) [Wil90]. Las funcionalidades más elaboradas de estos programas eran las que permitían agrandar y achicar la vista, elegir el ángulo de visión y distinguir entre distintos conjuntos de datos (máximo 5) utilizando distintos símbolos.

De esto se desprende claramente que estos sistemas están orientados básicamente sólo a la presentación de conjuntos de espinelos y no a la visualización de este tipo de minerales, lo que explica la falta de un conjunto de interacciones para la exploración del conjunto de datos. Al no contar con aplicaciones tridimensionales de visualización de minerales que utilicen el Prisma de los Espinelos los científicos se ven restringidos al uso de diversas herramientas de diagramado triangular y análisis de minerales que trabajan de forma separada, siendo inexistentes las herramientas que integran todas las tareas necesarias para el análisis de este tipo de datos.

Las aplicaciones de software más populares para el análisis de conjuntos de datos y diagramado petrológico son IGPet [Car02], MinPet [Ric95], MinCalc [Ber07] [MK92], GDCKit [EJFŠ03] y diversos Plug-Ins de Microsoft Excel. Por otra parte, para todo lo referente a diagramado triangular, los científicos utilizan herramientas como Tri-Plot [GM00], Tern-Plot [Mar96], etc.

En lo referente a visualización de espinelos, ninguna de las aplicaciones mencionadas alcanza la funcionalidad completa de la aplicación aquí presentada ni logra integrar todas las tareas mencionadas sin necesidad de utilizar herramientas alternativas de análisis o diagramado. Para lograrlo fue necesario el desarrollo de herramientas que integrasen múltiples vistas e interacciones *ad hoc*.

La técnica de múltiples vistas coordinadas [Rob07] se emplean a menudo como un concepto consolidado en el análisis visual. La idea principal de las múltiples vistas coordinadas es describir varias dimensiones de los datos utilizando múltiples vistas y permitir al usuario seleccionar interactivamente subconjuntos de los datos en una de ellas, haciendo efectiva la selección en todas las vistas restantes.

En este contexto se desarrollaron el *SpinelViz* [GCF⁺12] y el *Spinel Explorer* [GFG⁺14, GGF⁺15, GFG⁺17]. El *SpinelViz* brinda al usuario una vista en 3 dimensiones de uno o más conjuntos de datos cohabitando en él y el conjunto de interacciones necesarias para explorar este espacio y posibilitar un análisis efectivo de los datos. El *SpinelViz* explota el concepto de múltiples vistas utilizando dos vistas correlacionadas para visualizar el Prisma de los Espinelos y sus proyecciones permitiendo la visualización de varios conjuntos de datos a la vez.

Para completar otro paso del flujo de trabajo del análisis de minerales del grupo de los espinelos, se planteó la necesidad de comparar los datos del usuario con los contornos

de Barnes y Roeder. Con este objetivo en mente, en 2014 presentamos la primera versión del *Spinel Explorer*, un sistema semi-automático e interactivo de detección de campos composicionales para un conjunto arbitrario de minerales del grupo de los espinelos utilizando los contornos publicados por Barnes y Roeder. Para este sistema se desarrollaron nuevas técnicas de interacción que asisten al experto en el proceso de categorización de minerales. La nueva categorización interactiva es el resultado de un arduo estudio de las tareas de los geólogos, identificadas durante numerosas sesiones con expertos del dominio. Todos los contornos de Barnes y Roeder han sido integrados en el sistema y se pueden representar en tres proyecciones del Prisma de los Espinelos: dos rectangulares (representadas con *scatterplots*) y una cara triangular (representada con un *triplot*). Con este sistema se integraron, por primera vez, los gráficos generados por los expertos con los contornos de Barnes y Roeder para comparaciones simultáneas. De acuerdo a lo mejor de nuestro conocimiento, no existe una herramienta previa que asista a los expertos de forma interactiva y semi-automática en el proceso de categorización de minerales del grupo de los espinelos.

6.4. *SpinelViz*

El *SpinelViz* [GCF⁺12] es una aplicación de visualización de datos geológicos en 3 dimensiones que permite al usuario observar, explorar e interactuar con el Prisma de los Espinelos, brindando la posibilidad de cargar diferentes conjuntos de datos e interactuar con ellos en simultáneo. Adicionalmente, dado que los geólogos acostumbran a trabajar con la representación en 2 dimensiones de las proyecciones del espacio prismático, el *SpinelViz* provee una vista en 2 dimensiones, que permite explorar y visualizar los datos proyectados sobre las caras del Prisma de los Espinelos (ver figura 6.6). La herramienta soporta la representación de los dos espacios prismáticos usualmente utilizados en el campo de aplicación: el Prisma de la Magnetita y el Prisma Reducido (Ulvöespinelo) (ver figura 6.1). Esto permite que el usuario trabaje en una representación o en la otra dependiendo de los miembros finales predominantes de los cuales disponga.

Uno de los objetivos más importantes de esta herramienta consistió en lograr una representación adecuada de los conjuntos de composiciones minerales del grupo de los espinelos que permitiese comparar intuitivamente conjuntos de muestras entre sí, o mues-

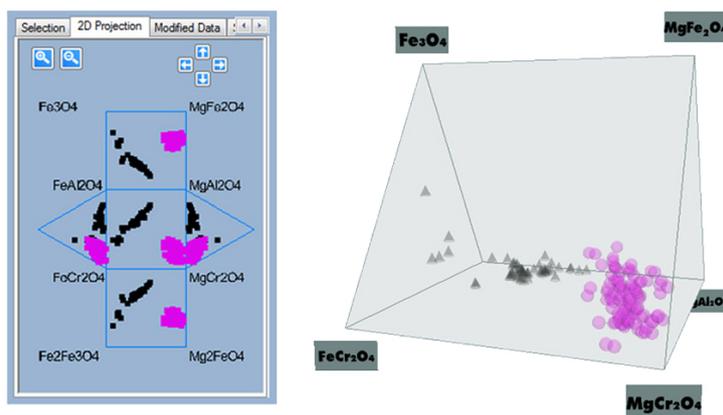


Figura 6.6: *SpinelViz*. Vistas correlacionadas en dos y tres dimensiones.

tras individuales contra un patrón de referencia dado, con el objetivo de determinar su origen tectónico. Para esto es imprescindible utilizar un mapeo visual efectivo y expresivo. Cada muestra de espinelo es representada con una marca dentro del espacio prismático (una esfera, un cono o un cilindro). El usuario puede seleccionar con qué marca se representarán sus datos y modificarla en cualquier momento (ver figura 6.7). Adicionalmente, las distintas muestras pueden dibujarse con distintos colores, predeterminados o definidos por el usuario, permitiendo una diferenciación intuitiva entre las distintas muestras, los distintos conjuntos de muestras y los distintos patrones de referencia. Por otra parte, para representar cada muestra de mineral se utiliza la propiedad de transparencia sobre la figura geométrica asociada, de modo tal que un conjunto de muestras posicionadas en ubicaciones cercanas se fusionan dando la sensación de formar un volumen dentro del Prisma (ver Figura 6.8(a)). Los canales de las marcas asociadas a cada muestra pueden editarse bajo demanda, asistiendo al usuario en el entendimiento de la composición mineralógica representada en la vista. El *SpinelViz* permite cargar hasta 10 patrones de referencia (o conjunto de muestras minerales) en el mismo espacio prismático, representando cada conjunto de datos con un color, tamaño o forma distintos (ver Figura 6.8(b)). Finalmente, con el objetivo de que la herramienta de visualización sea efectiva, se provee un conjunto de interacciones cuya definición, funcionalidad e implementación en el contexto del MUV se abordará en la sección 6.4.2.

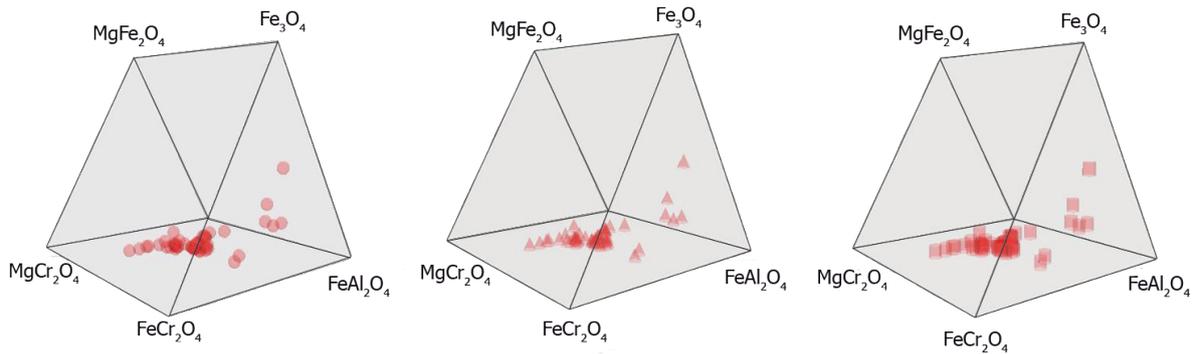


Figura 6.7: *SpinelViz*. Un conjunto de muestras de composiciones minerales representado con las tres marcas que soporta la aplicación.

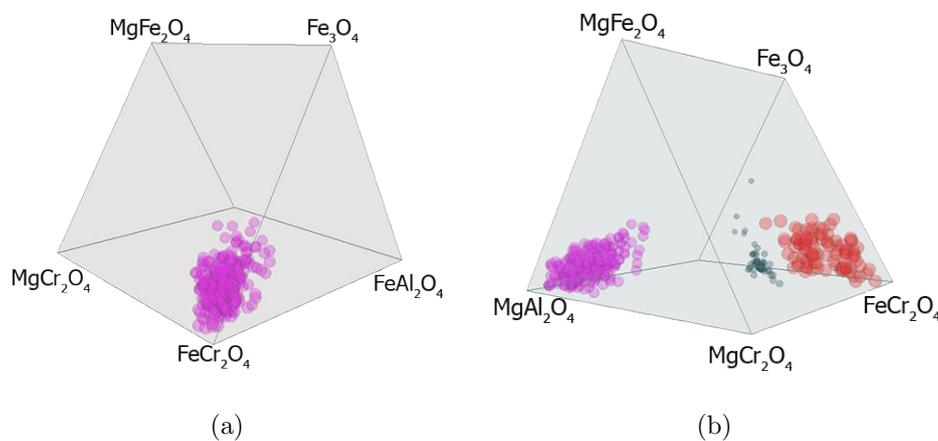


Figura 6.8: *SpinelViz*. (a) Un conjunto de datos representado con esferas y (b) tres patrones distintos cohabitando el mismo espacio prismático representados con tres marcas y colores diferentes.

6.4.1. *SpinelViz* en el MUV

En esta sección se abordará el *SpinelViz* en el contexto del MUV, detallando cómo se representan los conjuntos de datos subyacentes y qué acciones se aplican a los mismos en cada etapa y transformación del modelo. Como se mencionó anteriormente, el *SpinelViz* provee dos vistas distintas de los conjuntos de datos cargados, una vista en tres dimensiones que representa el espacio prismático y una vista en dos dimensiones que representa las proyecciones del prisma. Estas dos vistas están relacionadas y muestran los mismos datos de forma diferente. En el MUV en ejecución, a pesar de existir un único conjunto de datos crudos, existirán dos conjuntos de datos visualizados distintos, producto de una ramificación del *pipeline* en algún punto del proceso. Dado que las dos visualizaciones

muestran los mismos datos pero aplicando un mapeo visual y una técnica de visualización diferente, las ramas del pipeline subyacentes compartirán los estados de DC, DA y DaV, generándose la ramificación en la *Transformación de Mapeo Visual* (ver figura 6.9).

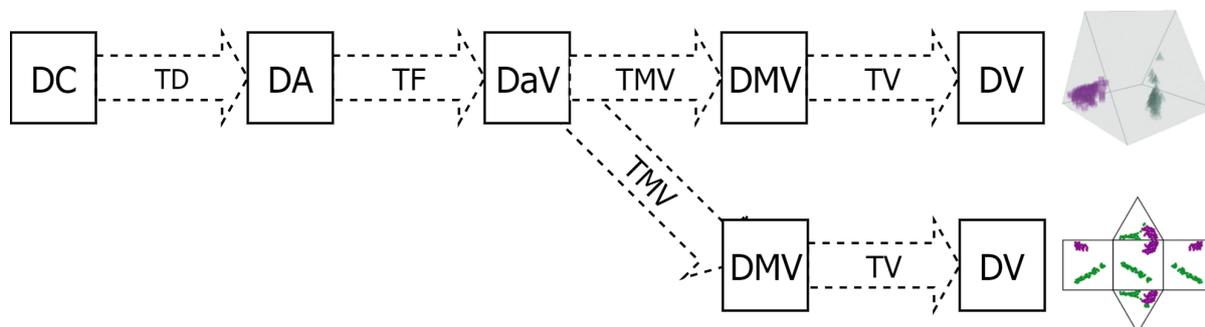


Figura 6.9: *SpinelViz* en el MUV. Ramificaciones del modelo en ejecución para representar las vistas 3D y 2D provistas por el *SpinelViz* .

Datos Crudos

El resultado del proceso descrito en la sección 6.2.1 corresponde a los datos crudos en el *SpinelViz*. Se trata de un archivo compuesto por tantas filas como muestras se hayan tomado *in situ*, y tantas columnas como la suma de la cantidad de elementos mayoritarios, de óxidos y de miembros finales que conforman su composición química. Además, cada muestra tiene asignado un identificador representativo y si el conjunto de DC contiene más de un patrón, éstos deben estar diferenciados de alguna manera para identificar qué muestra corresponde a cada patrón.

Transformación de Datos

En la Transformación de Datos, los datos crudos son convertidos a un formato interno al sistema, analizando la semántica y el tipo de los atributos y filtrando aquellos ítems y/o atributos que deseamos conservar para el resto del proceso interactivo de visualización. En este caso en particular, la semántica de los datos es definida por el experto geólogo, incluida en los datos crudos e interpretada por el sistema.

La representación resultante será una colección de *datasets* como los definidos en el capítulo 4. Para este ejemplo en particular, se instanciará el *dataset* con una Tabla como la definida en la sección 4.5 por cada patrón de referencia a representar, donde cada muestra será considerada un ítem de dato, y cada columna un atributo

de los ítems de datos. Siguiendo la clasificación para los atributos presentada en la sección 4.4, aquellos atributos que conforman la composición química de cada muestra se corresponden con atributos ordinales cuantitativos. El identificador de cada muestra, en cambio, se corresponde con un atributo categórico.

Como se detalló en la sección 6.2, la posición de un ítem de dato en el espacio prismático está dada por su composición química en término de 6 de sus 22 miembros finales. Es decir que, para representar un ítem de dato dentro de los dos espacios prismáticos soportados basta conservar, para cada ítem de dato, los 6 miembros finales que conforman el Prisma de la Magnetita y los 6 miembros finales que conforman el Prisma Reducido (Ulvöespinelo). Dado que ambos prismas comparten su base, son necesarios sólo ocho miembros finales para soportar ambas representaciones. Es por esto que en esta transformación se efectuará un filtrado de atributos, dejando fuera del conjunto de los DA todos los miembros finales que no sean necesarios para soportar la representación prismática (ver figura 6.10).

Cabe aclarar que los miembros finales pueden obtenerse como una combinación de los elementos mayoritarios y sus óxidos. Puede darse la situación de que los miembros finales no se encuentren presentes en los datos crudos; en este caso, éstos se calcularán en la *Transformación de Datos*. Entonces, los miembros finales no serían parte de los datos crudos y serían considerados *metadatos* generados en esta transformación.

Datos Abstractos

El conjunto de Datos Abstractos se compone del conjunto de datos potencialmente visualizables almacenados en un formato interno al sistema. Dado que el *SpinelViz* soporta la exploración y manipulación de varios conjuntos de datos (o patrones) a la vez, el conjunto de datos abstractos consta de una colección de tablas de m filas x n columnas, donde m corresponde a la cantidad de muestras para un patrón determinado y por cada muestra se conservan los n atributos seleccionados en la transformación anterior (ver figura 6.11).

Transformación de Filtrado

En esta transformación se seleccionarán aquellos datos que efectivamente van a estar presentes en la visualización. En función de qué espacio prismático se desee

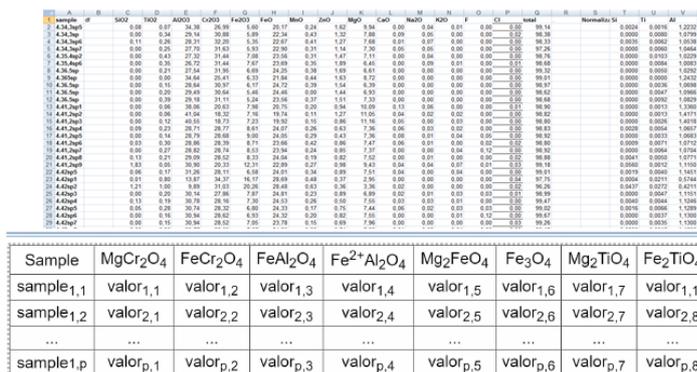


Figura 6.10: Filtrado de Atributos en la Transformación de Datos. De los cerca de 60 atributos que conforman la composición química de cada muestra se seleccionan únicamente los 8 miembros finales correspondientes a los vértices de los dos espacios prismáticos soportados, los óxidos y los elementos mayoritarios de cada muestra.

representar, se seleccionarán aquellos seis miembros finales correspondientes a su composición, descartando (tal vez momentáneamente) los dos restantes. En el caso de que la representación seleccionada sea el Prisma de la Magnetita, en esta transformación se seleccionarán los miembros finales $MgAl_2O_4$, $FeAl_2O_4$, $FeCr_2O_4$, $MgCr_2O_4$, Fe_3O_4 y $MgFeO_4$. En cambio, si la representación seleccionada es el Prisma Reducido (Ulvöespinelo), se seleccionarán los miembros finales $MgAl_2O_4$, $FeAl_2O_4$, $FeCr_2O_4$, $MgCr_2O_4$, Mg_2TiO_4 y Fe_2TiO_4 (ver figura 6.12).

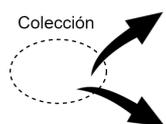
Nuevamente se lleva a cabo un filtrado de atributos.

Datos a Visualizar

El conjunto de datos a visualizar está compuesto por aquellos datos que efectivamente estarán presentes en la visualización. En el caso del *SpinelViz*, el conjunto de Datos a Visualizar respeta la misma estructura que el conjunto de datos abstractos, conservando únicamente seis de los ocho miembros finales, los óxidos y los elementos mayoritarios por ítem de dato (ver figura 6.13)

Transformación de Mapeo Visual

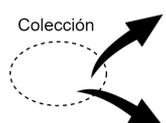
En la Transformación de Mapeo Visual se asocia a los datos a visualizar la información necesaria para su representación en pantalla, enriqueciéndolos con una estructura visual. En esta transformación se generará una representación *mapeo visual* como la descrita en la sección 4.6 del capítulo 4. El *mapeo visual* estará compuesto



Sample	MgCr ₂ O ₄	FeCr ₂ O ₄	FeAl ₂ O ₄	Fe ²⁺ Al ₂ O ₄	Mg ₂ FeO ₄	Fe ₃ O ₄	Mg ₂ TiO ₄	Fe ₂ TiO ₄
sample _{1,1}	valor _{1,1}	valor _{1,2}	valor _{1,3}	valor _{1,4}	valor _{1,5}	valor _{1,6}	valor _{1,7}	valor _{1,1}
sample _{1,2}	valor _{2,1}	valor _{2,2}	valor _{2,3}	valor _{2,4}	valor _{2,5}	valor _{2,6}	valor _{2,7}	valor _{2,8}
...
sample _{1,p}	valor _{p,1}	valor _{p,2}	valor _{p,3}	valor _{p,4}	valor _{p,5}	valor _{p,6}	valor _{p,7}	valor _{p,8}

Sample	MgCr ₂ O ₄	FeCr ₂ O ₄	FeAl ₂ O ₄	Fe ²⁺ Al ₂ O ₄	Mg ₂ FeO ₄	Fe ₃ O ₄	Mg ₂ TiO ₄	Fe ₂ TiO ₄
sample _{2,1}	valor _{1,1}	valor _{1,2}	valor _{1,3}	valor _{1,4}	valor _{1,5}	valor _{1,6}	valor _{1,7}	valor _{1,1}
sample _{2,2}	valor _{2,1}	valor _{2,2}	valor _{2,3}	valor _{2,4}	valor _{2,5}	valor _{2,6}	valor _{2,7}	valor _{2,8}
...
sample _{2,x}	valor _{x,1}	valor _{x,2}	valor _{x,3}	valor _{x,4}	valor _{x,5}	valor _{x,6}	valor _{x,7}	valor _{x,8}

Figura 6.11: Datos Abstractos en el *SpinelViz*: el conjunto de DA consta de una colección de tablas de m filas x n columnas, donde m corresponde a la cantidad de muestras para un patrón de referencia en particular, y por cada muestra se conservan n atributos seleccionados en la transformación anterior.



Sample	MgCr ₂ O ₄	FeCr ₂ O ₄	FeAl ₂ O ₄	Fe ²⁺ Al ₂ O ₄	Mg ₂ FeO ₄	Fe ₃ O ₄	Mg ₂ TiO ₄	Fe ₂ TiO ₄
sample _{1,1}	valor _{1,1}	valor _{1,2}	valor _{1,3}	valor _{1,4}	valor _{1,5}	valor _{1,6}	valor _{1,7}	valor _{1,1}
sample _{1,2}	valor _{2,1}	valor _{2,2}	valor _{2,3}	valor _{2,4}	valor _{2,5}	valor _{2,6}	valor _{2,7}	valor _{2,8}
...
sample _{1,p}	valor _{p,1}	valor _{p,2}	valor _{p,3}	valor _{p,4}	valor _{p,5}	valor _{p,6}	valor _{p,7}	valor _{p,8}

Sample	MgCr ₂ O ₄	FeCr ₂ O ₄	FeAl ₂ O ₄	Fe ²⁺ Al ₂ O ₄	Mg ₂ FeO ₄	Fe ₃ O ₄	Mg ₂ TiO ₄	Fe ₂ TiO ₄
sample ₁	valor _{1,1}	valor _{1,2}	valor _{1,3}	valor _{1,4}	valor _{1,5}	valor _{1,6}	valor _{1,7}	valor _{1,1}
sample ₂	valor _{2,1}	valor _{2,2}	valor _{2,3}	valor _{2,4}	valor _{2,5}	valor _{2,6}	valor _{2,7}	valor _{2,8}
...
sample _p	valor _{p,1}	valor _{p,2}	valor _{p,3}	valor _{p,4}	valor _{p,5}	valor _{p,6}	valor _{p,7}	valor _{p,8}

Figura 6.12: Transformación de Filtrado en el *SpinelViz*. En este caso, se seleccionaron los seis miembros finales necesarios para la representación de los datos en el Prisma de la Magnetita, junto con los valores de los óxidos y de los elementos mayoritarios para cada muestra.

por el conjunto de *datasets*, el sustrato espacial y una colección de asociaciones entre datos e ítems visuales.

Dado que el *SpinelViz* soporta dos vistas, una en tres dimensiones del espacio prismático, y otra en dos dimensiones de las proyecciones del espacio prismático, debemos considerar la Transformación de Mapeo Visual de forma independiente para cada vista ya que, aunque el sustrato espacial será similar, las marcas difieren para ambos casos. A raíz de esto último, a partir de esta transformación se generarán dos conjuntos distintos de datos mapeados visualmente, uno para cada vista, y es a partir de este punto que se ramificará el pipeline de visualización (ver figura 6.9).

Sample	MgCr ₂ O ₄	FeCr ₂ O ₄	FeAl ₂ O ₄	Fe ²⁺ Al ₂ O ₄	Mg ₂ FeO ₄	Fe ₃ O ₄
sample _{1,1}	valor _{1,1}	valor _{1,2}	valor _{1,3}	valor _{1,4}	valor _{1,5}	valor _{1,6}
sample _{1,2}	valor _{2,1}	valor _{2,2}	valor _{2,3}	valor _{2,4}	valor _{2,5}	valor _{2,6}
...
sample _{1,p}	valor _{p,1}	valor _{p,2}	valor _{p,3}	valor _{p,4}	valor _{p,5}	valor _{p,6}

Sample	MgCr ₂ O ₄	FeCr ₂ O ₄	FeAl ₂ O ₄	Fe ²⁺ Al ₂ O ₄	Mg ₂ FeO ₄	Fe ₃ O ₄
sample _{2,1}	valor _{1,1}	valor _{1,2}	valor _{1,3}	valor _{1,4}	valor _{1,5}	valor _{1,6}
sample _{2,2}	valor _{2,1}	valor _{2,2}	valor _{2,3}	valor _{2,4}	valor _{2,5}	valor _{2,6}
...
sample _{2,x}	valor _{x,1}	valor _{x,2}	valor _{x,3}	valor _{x,4}	valor _{x,5}	valor _{x,6}

Figura 6.13: Datos a Visualizar en el *SpinelViz*. El conjunto de DaV respeta la misma estructura que el conjunto de DA, conservando únicamente seis de los ocho miembros finales, los óxidos y los elementos mayoritarios por cada muestra. Los miembros finales seleccionados corresponden a la representación del Prisma de la Magnetita.

- Transformación de Mapeo Visual para la Vista 3D
 - Sustrato Espacial: Ejes: no estructurados. Cantidad de Ejes: 0.
 - Marcas: Todas las muestras correspondientes a un patrón se representarán con la misma marca. Con el objetivo de proveer la capacidad de diferenciar entre diferentes patrones, se utilizarán distintas marcas para representar patrones distintos. El *SpinelViz* permite representar las muestras como esferas, conos o cilindros. En el caso del ejemplo, las muestras correspondientes al primer patrón se representarán con esferas y las correspondientes al segundo patrón se representarán con conos.
 - Canales: Los canales que se utilizarán serán el tamaño y el color de cada elemento visual, que estarán determinados por el patrón al que pertenecen. Todas las muestras de un mismo patrón tendrán inicialmente el mismo color, tamaño y forma. En el caso del ejemplo, las muestras correspondientes al primer patrón se representarán con color violeta y las correspondientes al segundo patrón se representarán con color verde. Inicialmente, el tamaño de todas las marcas será asignado con un valor por defecto.
- Transformación de Mapeo Visual para la Vista 2D
 - Sustrato Espacial: Ejes: no estructurados. Cantidad de Ejes: 0.
 - Marcas: Todos los ítems de dato se asociarán a la misma marca (punto).

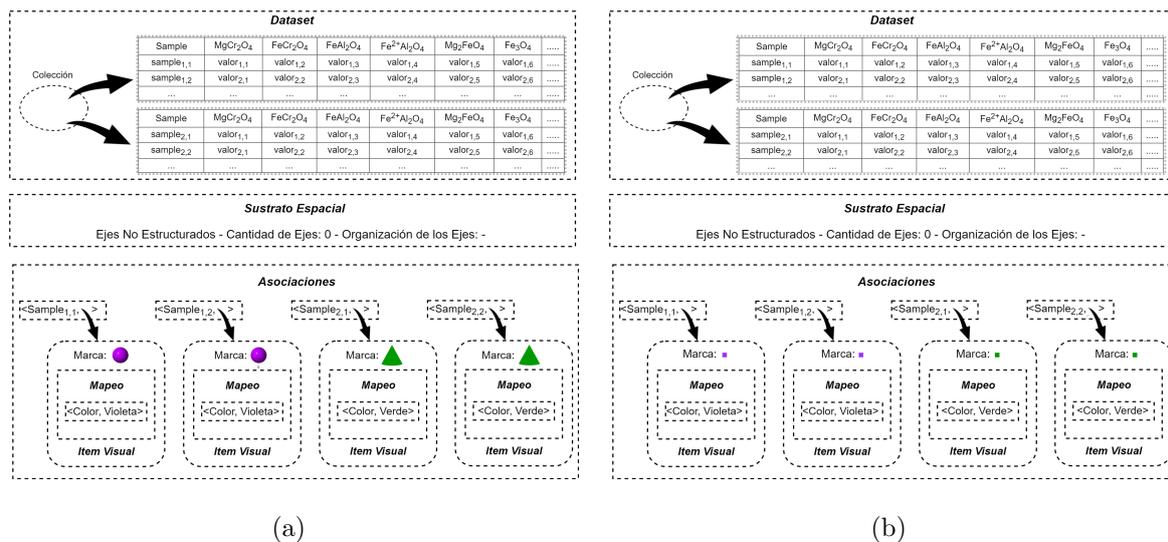


Figura 6.14: (a) Datos Mapeados Visualmente para la Vista 3D y (b) para la vista 2D. A pesar de que los DaV y el sustrato espacial son iguales para ambas representaciones, las marcas difieren. El color asociado a las marcas depende del patrón al que correspondan.

- **Canales:** El canal que se utilizará será el color de cada elemento visual, que estará determinado por el patrón al que pertenecen. Inicialmente, todas las muestras correspondientes a un mismo patrón se asociarán al mismo color. En el caso del ejemplo, las marcas asociadas al primer patrón se representarán con color violeta y las asociadas al segundo patrón se representarán con color verde, colores que coinciden con los asignados en la Vista 3D.

Datos Mapeados Visualmente

En la figura 6.14 se ilustran los dos conjuntos de datos mapeados visualmente generados en la *Transformación de Mapeo Visual*.

Transformación de Visualización

En esta transformación se aplica la técnica de visualización. La técnica aplicada será distinta para la vista 3D que para la vista 2D. Para la vista 3D se aplica una técnica de visualización llamada *Prisma de los Espinelos*, en tanto que para la vista 2D se aplica una técnica de visualización en 2 dimensiones denominada *Prisma Proyectado*.

- Transformación de Visualización para la Vista 3D

Para la vista 3D no se aplica una técnica convencional de visualización, sino una técnica en particular diseñada a medida. Esta técnica implica el diagramado del espacio prismático, el cálculo de la posición de cada muestra dentro del espacio y finalmente se representa cada ítem de dato en la posición previamente calculada con el color, tamaño y forma determinados en la transformación de mapeo visual (ver figura 6.15)

En el algoritmo 1 se detalla la técnica de visualización *Prisma de los Espinelos*. Este algoritmo requiere como dato de entrada el conjunto de ítems a visualizar, que será una colección de ítems de datos, cada uno compuesto por 6 valores: $(mf_1, mf_2, mf_3, mf_4, mf_5, mf_6)$, siendo mf_i la composición del i -ésimo miembro final para el ítem de dato correspondiente. Para definir el prisma en el espacio 3D se utiliza una estructura P que almacenará cada uno de los vértices del prisma. Cabe aclarar que para cada ítem de dato, mf_i corresponderá al miembro final asociado al vértice P_i del Prisma de Espinelos. La primitiva “CalcularCoordenadasVérticesPrisma” calcula las posiciones correspondientes a los vértices del prisma en tres dimensiones.

- Transformación de Visualización para la Vista 2D

La técnica de visualización aplicada a la vista 2D implica el diagramado del prisma proyectado en 2 dimensiones, el cálculo de la posición de cada muestra proyectada dentro del espacio y finalmente la representación de cada ítem de dato en la posición previamente calculada con el color determinado en la transformación de mapeo visual (ver figura 6.15)

Esta técnica de visualización en particular se denomina *Prisma Proyectado* y se detalla en el algoritmo a continuación. Este algoritmo requiere como dato de entrada el conjunto de ítems a visualizar, que será una colección de ítems de datos y las coordenadas de los vértices del prisma. Cada ítem de dato estará compuesto por 6 valores: $(mf_1, mf_2, mf_3, mf_4, mf_5, mf_6)$, siendo mf_i la composición del i -ésimo miembro final para el ítem de dato correspondiente. Los vértices del prisma estarán definidos en la estructura P . La primitiva “CalcularCoordenadasProyectadasDelPrisma” calcula las posiciones correspondientes a las coordenadas del prisma proyectadas en dos dimensiones. La primitiva

Algoritmo 1 Prisma de los Espinelos

Técnica de Visualización: Prisma de los Espinelos

Entrada: DS : Conjunto de ítems de datos $(mf_1, mf_2, mf_3, mf_4, mf_5, mf_6)$

definir: $P[]$: Colección de coordenadas de los 6 vértices del Prisma (x, y, z)

definir: $Coordenadas[]$: Colección de coordenadas finales de los ítems (x, y, z)

definir: $coord_x, coord_y, coord_z$ reales

$Prisma \leftarrow$ **CalcularCoordenadasVérticesPrisma** ()

para todo ítem de dato d en el conjunto DS **realizar**

$$coord_x \leftarrow D_{MF_1} \times P[1]_x + (D_{MF_2} \times P[2]_x) + (D_{MF_3} \times P[3]_x) + (D_{MF_4} \times P[4]_x) \\ + (D_{MF_5} \times P[5]_x) + (D_{MF_6} \times P[6]_x)$$

$$coord_y \leftarrow D_{MF_1} \times P[1]_y + (D_{MF_2} \times P[2]_y) + (D_{MF_3} \times P[3]_y) + (D_{MF_4} \times P[4]_y) \\ + (D_{MF_5} \times P[5]_y) + (D_{MF_6} \times P[6]_y)$$

$$coord_z \leftarrow D_{MF_1} \times P[1]_z + (D_{MF_2} \times P[2]_z) + (D_{MF_3} \times P[3]_z) + (D_{MF_4} \times P[4]_z) \\ + (D_{MF_5} \times P[5]_z) + (D_{MF_6} \times P[6]_z)$$

$Coordenadas \leftarrow$ **insertar** $(coord_x, coord_y, coord_z)$

fin para

DibujarPrisma (P)

para todo coordenada c en el conjunto $Coordenadas$ **realizar**

DibujarÍtemDeDato $(c, color, forma, tamaño)$

fin para

fin

‘CalcularProyección’ calcula la posición de una muestra proyectada en dos dimensiones.

Algoritmo 2 Prisma Proyectado

Técnica de Visualización: Prisma Proyectado

Entrada: DS : Conjunto de ítems de datos ($mf_1, mf_2, mf_3, mf_4, mf_5, mf_6$)

P : Coordenadas de los vértices del prisma en 3D

definir: $PtosP[]$: Coordenadas de los vértices del Prisma proyectados en 2D(x, y)

definir: $CoordProy[]$: Colección de coordenadas de los ítems proyectados(x, y)

definir: $pto.proyectado$: Punto en 2D (x, y)

$PtosP \leftarrow$ **CalcularCoordenadasProyectadasDelPrisma** (P)

para todo ítem de dato d en el conjunto DS **realizar**

para todo cara c del *Prisma* **realizar**

$pto.proyectado \leftarrow$ **CalcularProyección**($c, d, PtosP$)

$CoordProy \leftarrow$ **insertar**($pto.proyectado$)

fin para

fin para

DibujarPrismaProyectado($PtosP$)

para todo coordenada $coord$ en el conjunto $CoordProy$ **realizar**

DibujarÍtemDeDato ($coord, color$)

fin para

fin

Datos Visualizados

El conjunto de datos visualizados es el producto final del proceso. En el caso particular del *SpinelViz* contamos con dos conjuntos de datos visualizados, uno correspondiente a la vista 3D y el otro correspondiente a las proyecciones en la vista 2D. Ambos conjuntos de datos constituyen el punto de partida para el proceso de exploración que realizará el usuario. En la figura 6.15 se ilustra el proceso completo del *SpinelViz* en el MUV.

6.4.2. Interacciones

La necesidad de lograr una visualización interactiva nos impulsa a poner énfasis en el estudio y el análisis de qué interacciones resultan importantes en función del campo de aplicación. Entre las interacciones provistas por el *SpinelViz* se cuenta con interacciones básicas que permiten la navegación de la escena 3D y 2D y la exploración de los datos

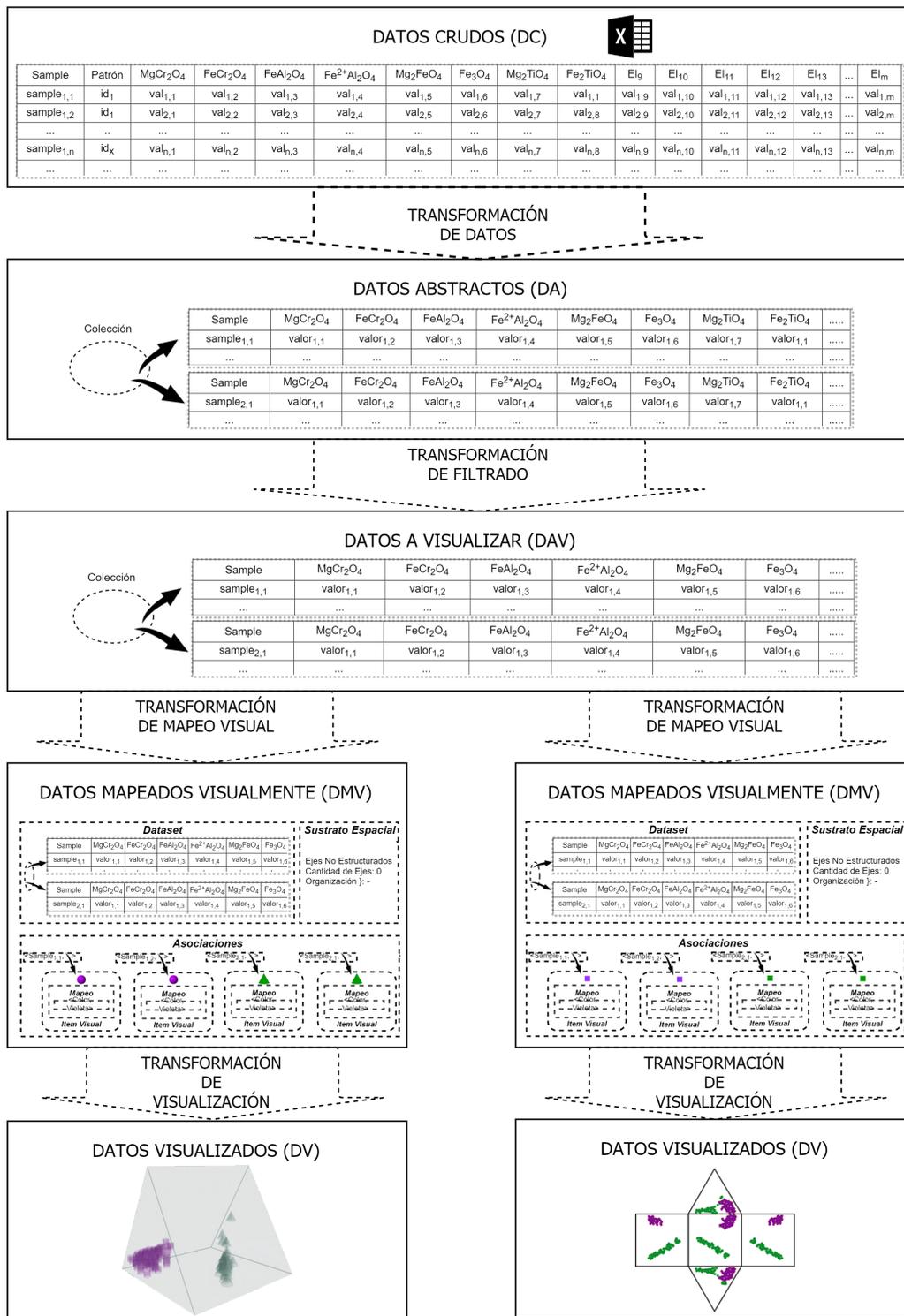


Figura 6.15: SpinelViz en el MUV.

desde diferentes puntos de vista, tales como las de rotación, zoom geométrico y traslación de la escena y con interacciones que involucran tareas específicas del campo de aplicación y no son comunes a otras aplicaciones de visualización (ver figura 6.16). A continuación se describen las interacciones mas importantes provistas por el SpinelViz :

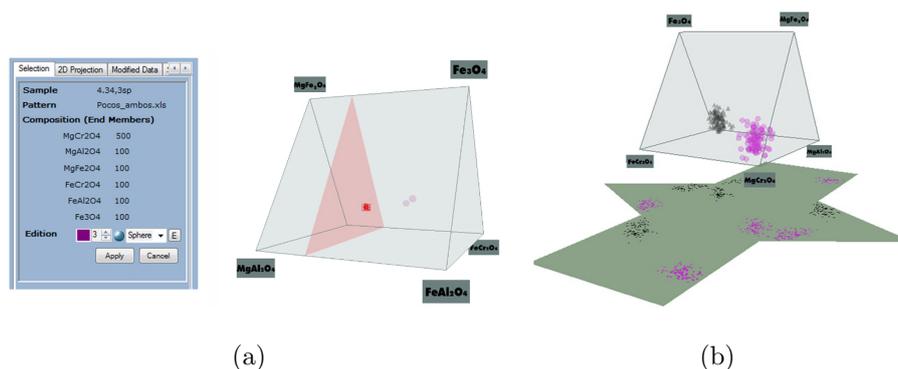


Figura 6.16: Interacciones en el *SpinelViz*. (a) *Zoom Semántico*. (b) *Proyecciones 2D integradas en el espacio 3D*.

Interacciones sobre la vista 2D

- Navegación:** El *SpinelViz* provee interacciones básicas de escalado y traslación para su vista en dos dimensiones. A nivel de las interacciones de usuario definidas en la sección 5.4 del capítulo 5, estas interacciones se traducen en *Zooming* y *Panning*, ambas interacciones a nivel de usuario sobre la vista. Estas interacciones afectan directamente a la representación de la vista y no tienen influencia sobre las etapas ni transformaciones anteriores del pipeline de visualización. Las interacciones de navegación sobre la vista 2D se resuelven en el estado *Datos Visualizados*.

Interacciones sobre la vista 3D

- Navegación:** El *SpinelViz* provee interacciones básicas de escalado, rotación y traslación para su vista en tres dimensiones. A nivel de las interacciones de usuario definidas en la sección 5.4 del capítulo 5, estas interacciones se traducen en *Zooming*, *Rotación* y *Panning*, todas interacciones a nivel de usuario sobre la vista. Estas interacciones afectan directamente a la representación de la vista, no tienen influencia sobre las etapas ni transformaciones anteriores del pipeline de visualización. Las interacciones de navegación sobre la vista 3D se resuelven en el estado *Datos Visualizados*.
- Selección:** El *SpinelViz* provee esta interacción para resaltar momentáneamente una muestra mediante la selección con el *mouse*. El resaltado se logra

modificando el color y el tamaño de la muestra y dibujando un triángulo característico (ver figura 6.16(a)). A nivel de las interacciones de usuario definidas en la sección 5.4 del capítulo 5, estas interacciones se traducen en *Brushing*, interacción compuesta a nivel de usuario. Estas interacción se resuelve en la *Transformación de Mapeo Visual*.

- **Zoom Semántico:** Esta interacción permite seleccionar una muestra para obtener más información sobre la misma (valores de los óxidos, etc.) Estos valores se muestran en una sección apartada de la vista 3D (ver figura 6.16(a)). Esta interacción se descompone en una selección en la vista, y un nuevo filtrado de atributos en la transformación de filtrado para insertar en los DaV los nuevos atributos que se desea mostrar. A nivel de las interacciones de usuario definidas en la sección 5.4 del capítulo 5, esta interacción se traduce en *Zoom Semántico*, interacción compuesta a nivel de usuario. Esta interacción se resuelve en la *Transformación de Filtrado*.
- **Mostrar/Ocultar Proyecciones 2D en el espacio 3D:** Estas interacciones permiten integrar en la vista 3D las proyecciones del Prisma de los Espinelos en 2D. El *SpinelViz* permite al usuario mostrar por debajo del Prisma de los Espinelos una proyección del mismo (ver figura 6.16(b)). Estas interacciones afectan directamente la implementación de la técnica de visualización, que en función de la demanda del usuario, mostrará u ocultará las proyecciones del prisma sobre la vista 3D. A nivel de las interacciones de usuario definidas en la sección 5.4 del capítulo 5, esta interacción se traduce en *SeleccionarTécnica*, que se resuelve en la *Transformación de Visualización*.

Interacciones Generales

- **Cargar un conjunto de datos:** Esta interacción permite cargar un nuevo conjunto de datos (o patrón) al sistema de visualización. Este patrón se integrará con los ya existentes, modificando en una primera instancia el conjunto de DC. A nivel de las interacciones de usuario definidas en la sección 5.4 del capítulo 5, esta interacción se traduce en *AbrirConjuntoDatos*, que se resuelve en el estado DC.
- **Eliminar un conjunto de datos:** Esta interacción permite eliminar un

patrón del conjunto de datos; esta operación no puede deshacerse sin volver a cargar el conjunto de datos al sistema. A nivel de las interacciones de usuario definidas en la sección 5.4 del capítulo 5, esta interacción se traduce en *EliminarDatos*, que se resuelve en el estado DA.

- Ocultar/Mostrar un conjunto de datos** Estas interacciones permiten ocultar o mostrar (tal vez momentáneamente) un conjunto de datos. Al ocultarlo, éste no es eliminado del conjunto de DA, simplemente es eliminado del conjunto de DaV. A nivel de las interacciones de usuario definidas en la sección 5.4 del capítulo 5, ocultar se traduce en *OcultarDatos*, y mostrar en *MostrarDatos*, ambas se resuelven en la *Transformación de Filtrado*.
- Codificación:** El *SpinelViz* provee interacciones que permiten modificar la marca asociada a un conjunto de datos o a un ítem de dato en particular, así como también su tamaño o su color. A nivel de las interacciones de usuario definidas en la sección 5.4 del capítulo 5, estas interacciones se traducen en *AplicarMarca* y *AsociarValorACanal*, ambas interacciones a nivel de usuario sobre el mapeo visual. Estas interacciones se resuelven en la *Transformación de Mapeo Visual*.

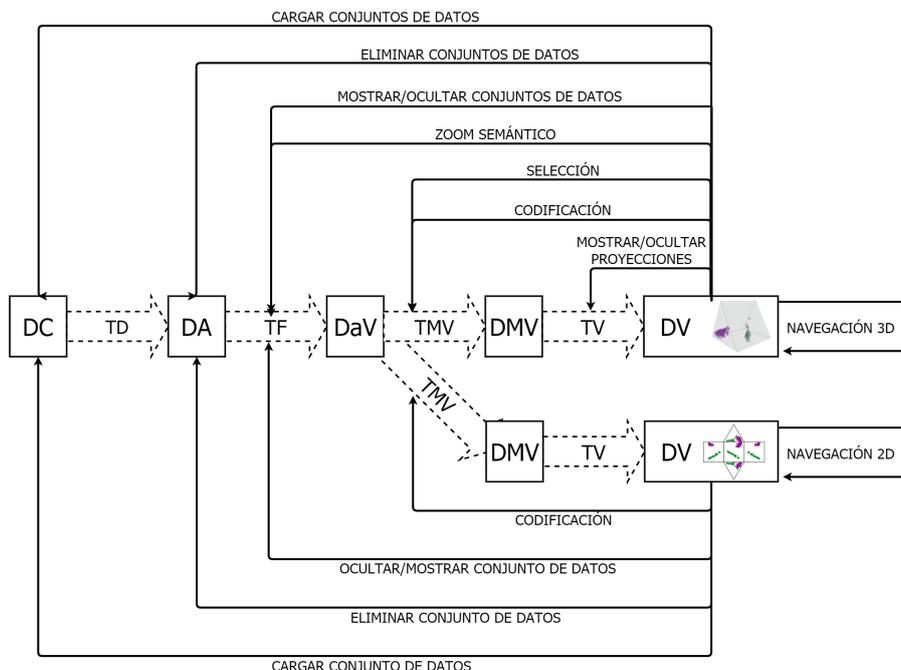


Figura 6.17: *SpinelViz* en el MUV. Interacciones.

6.5. *Spinel Explorer*

El *Spinel Explorer* es un entorno unificado de análisis visual para datos geológicos. El sistema utiliza vistas múltiples y coordinadas dando soporte para la mayoría de los gráficos que los geólogos utilizan en el contexto de descrito en la sección 6.2. Al igual que el *SpinelViz*, el *Spinel Explorer* fue diseñado para el análisis y exploración de minerales del grupo de los espinelos, pero a diferencia de éste, provee más de dos vistas coordinadas e incorpora técnicas de visualización como coordinadas paralelas, diagramas *triplots*, *scatterplots*, histogramas, etc, y provee además un proceso de categorización semi-interactivo de minerales del grupo de los espinelos. El *Spinel Explorer* integra los contornos de Barnes y Roeder utilizando una representación poligonal y permite superponer las proyecciones de los conjuntos de datos efectuadas sobre las distintas caras del Prisma de los Espinelos para ofrecer una comparación visual. Provee, además, una categorización semi-automática basada en la cantidad de puntos ubicado dentro de los contornos. En la figura 6.18 se puede observar una captura del *Spinel Explorer* en ejecución. En la misma, se muestran cuatro de sus vistas más significativas: un *triplot* (usualmente utilizado para representar las proyecciones de las muestras sobre las caras triangulares del Prisma de los Espinelos), un *scatterplot* (usualmente utilizado para representar las proyecciones de las muestras sobre la base del Prisma de los Espinelos), coordinadas paralelas y una vista 3D, que representa el Prisma de los Espinelos y sus proyecciones.

Al introducir este sistema proponemos un nuevo flujo de trabajo para la categorización de minerales del grupo de los espinelos [GFG⁺17]. La integración de los contornos de Barnes y Roeder implican su transformación en polígonos, que se almacenarán en una base de datos de contornos. Los contornos de Barnes y Roeder se podrán mostrar en los *scatterplots* y *triplots* durante todo el proceso de exploración de los datos. De este modo, se cuenta con una base de datos de contornos almacenada en el sistema, lo que permite a los expertos visualizar los mismos sin la necesidad de utilizar herramientas adicionales que involucran la colección predeterminada de imágenes de los contornos y procesos tediosos de alineación.

El análisis comienza cuando el experto carga un conjunto de muestras en el *Spinel Explorer*. Con el objetivo de estudiar las variaciones composicionales, grupos químicos y subgrupos, y de clasificar los minerales en función de su campo composicional, el *Spinel Explorer* permite a los expertos la construcción de los diagramas correspondientes a las

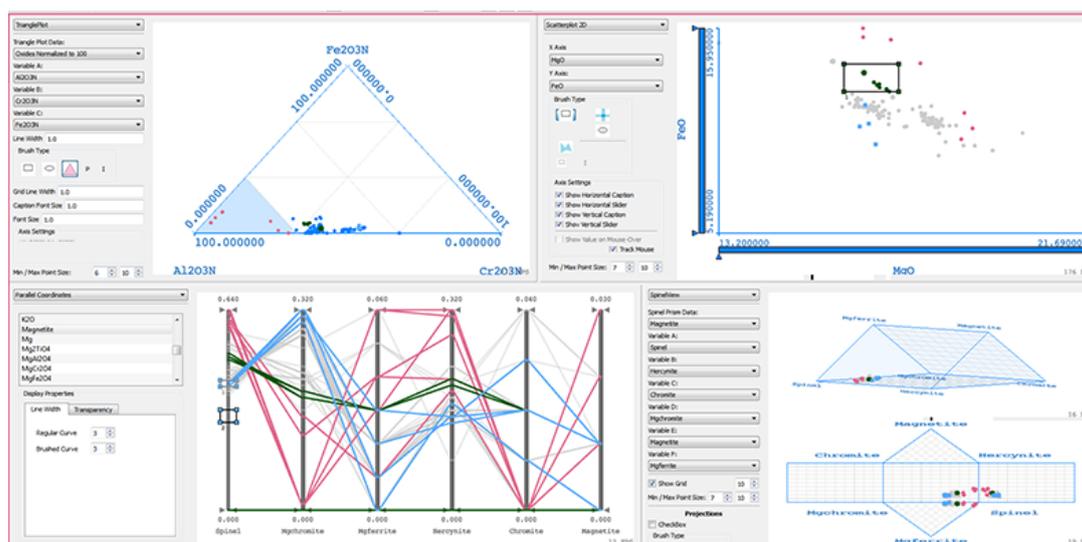


Figura 6.18: *Spinel Explorer*. Cuatro de sus vistas más significativas, un *Triplot* (arriba-izquierda) un *Scatterplot* (arriba-derecha), una vista de coordenadas paralelas (abajo-izquierda) y la vista 3D, que representa el Prisma de los Espinelos con sus proyecciones (abajo-derecha).

proyecciones del Prisma de los Espinelos, además de muchos otros diagramas, como por ejemplo coordenadas paralelas o histogramas.

A continuación, se comparan los diagramas generados con los contornos de Barnes y Roeder almacenados en la base de datos de contornos. Para esto, el usuario puede seleccionar qué contorno desea solapar con sus datos (si los expertos conjeturan un cierto campo composicional), o el sistema puede buscar los contornos que contienen la mayoría de las muestras del conjunto de puntos. El *Spinel Explorer* clasifica automáticamente los contornos de Barnes y Roeder basado en el número de muestras que incluyen. El usuario puede proporcionar dos umbrales, uno para el percentil 50, y el otro para los contornos del percentil 90. Todos los contornos para los cuales el número de muestras que caen en su interior es mayor que el umbral proporcionado por el usuario, se enumeran como posibles campos composicionales. Estas listas de categorización (posibles campos composicionales) se proporcionan para cada una de las cuatro proyecciones del Prisma de los Espinelos por separado. Todos los posibles contornos se muestran en una vista previa debajo de la visualización correspondiente. La vista previa se muestra como una tira de pequeños contornos ordenados por el número de muestras que encierra (ver figura 6.19). En estos pequeños contornos, se utiliza el canal de transparencia para reflejar el nivel de superposición entre el conjunto de puntos y el contorno representado. Ahora, el usuario puede seleccionar un contorno de la vista previa para solaparlo con el conjunto de muestras

visualizado. La información cuantitativa provista sobre el número de muestras dentro de los contornos y sobre la velocidad del proceso de comparación, remarca que este proceso de categorización es considerablemente superior al utilizado en el flujo de trabajo tradicional presentado en la sección 6.2.1.

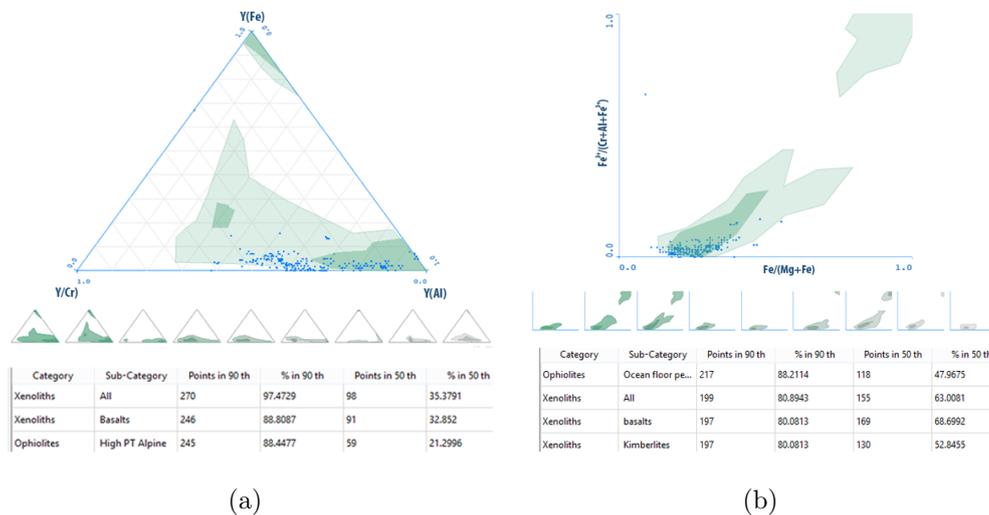


Figura 6.19: Categorización Semi-Automática en el *Spinel Explorer*. (a) Listas de categorización y vistas previas para la vista triplot y (b) *scatterplot*. El conjunto de muestras corresponde a Xenolitos en Basaltos. Observar que si bien las listas de categorización son diferentes para las dos vistas, el campo composicional de Xenolitos en Basaltos aparece en ambas. Adicionalmente, los contornos de Barnes y Roeder para este campo composicional en particular se muestran solapados con el conjunto de datos proporcionado por el usuario.

Adicionalmente, el *Spinel Explorer* provee la creación de contornos sobre el conjunto de datos del usuario. Esta nueva interacción permite a los expertos la creación de contornos de diferentes percentiles. El usuario tiene el control total de los parámetros de creación de contornos y puede crear libremente contornos de percentil en el rango de 1 a 100. Esto permite la visualización de contornos de diferentes percentiles generados a partir del conjunto de muestras del usuario, lo que ayuda a los expertos a obtener una visión más profunda de sus datos.

El *Spinel Explorer* proporciona la creación de contornos para todas las proyecciones del prisma. Una vez creados los contornos, éstos se pueden superponer y comparar con los contornos de Barnes y Roeder. La creación de contornos está basada en densidad, y sigue la estrategia utilizada por Barnes y Roeder para generar los propios. En la figura 6.20 se ilustra la creación de contornos sobre los datos del usuario para las tres proyecciones del

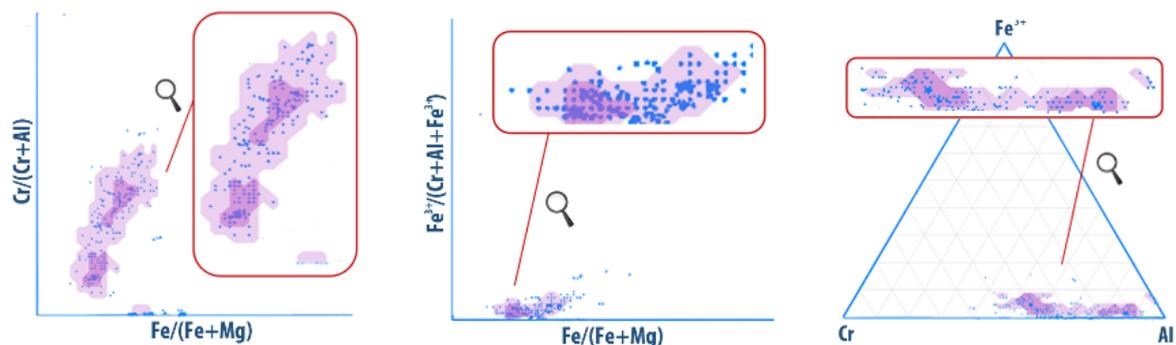


Figura 6.20: Creación de Contornos en el *Spinel Explorer*. Creación de contornos sobre los datos del usuario para las tres proyecciones del prisma. Los contornos de percentil 50 se dibujan en color púrpura oscuro y los contornos de percentil 90 en púrpura claro. El usuario puede seleccionar interactivamente el percentil del contorno a generar.

prisma. Los contornos de percentil 50 se dibujan en color púrpura oscuro y los contornos de percentil 90 en púrpura claro. El usuario puede seleccionar interactivamente el percentil del contorno a generar. En el algoritmo 3 se detalla el proceso de generación de contornos, que a continuación describimos de manera conceptual. Comenzamos con un conjunto de muestras (puntos). El área de la visualización *tripplot* o *scatterplot* se subdivide en celdas triangulares o rectangulares respectivamente formando una grilla. El tamaño de la celda es una décima parte del ancho del espacio de mostrado de la gráfica correspondiente. Para cada celda se cuenta el número de muestras incluidas y a éste se lo denomina densidad de la celda (ver algoritmo 4). Las celdas se ordenan de acuerdo a sus densidades en una lista en orden descendente. Para un valor percentil específico primero determinamos las celdas de la grilla que pertenecen al interior del contorno. Se recorre la lista ordenada de celdas y se suman sus valores de densidad hasta alcanzar el valor del percentil especificado. Las celdas que se posicionan más abajo en la lista quedarán fuera del contorno. Finalmente, el contorno se crea automáticamente aplicando una simplificación del algoritmo *Marching Squares* [HWC⁺05] sobre las celdas que caen dentro del mismo.

Finalmente y con el objetivo de favorecer la exploración de los datos, se incorporaron al *Spinel Explorer* dos técnicas novedosas y creadas *ad hoc* de *Brushing* y *Linking*: el *brushing* por percentil y el *brushing* por contorno. El **brushing por percentil** permite seleccionar, mediante un simple clic, todos los puntos pertenecientes a un contorno dado. Dependiendo de la configuración, sólo los puntos pertenecientes al contorno de percentil 90 ó 50 son seleccionados. Esta técnica se soporta no sólo para los contornos de Barnes

Algoritmo 3 Creación de Contornos

CreaciónDeContornos(LISTADEMUESTRAS, PERCENTIL)*cont* ← *size(listaDeMuestras)**listaDeTuplas* ← *CalcularDensidad(listaDeMuestras)**cantPuntosEnNuevoContorno* ← $(cont/100)*percentil$ **para todo** *index:=0:cantPuntosEnNuevoContorno* **realizar***insertar listaDeTuplas[index] in PuntosEnNuevoContorno***fin para****marchingSquares**(*PuntosEnNuevoContorno*)**fin**

Algoritmo 4 Cálculo de Densidad

CalcularDensidad(LISTADEMUESTRAS)**para todo** *muestra en listaDeMuestras* **realizar***kernel* ← *calcularKernel(muestra)**densidad* ← *cantPuntosDentroKernel(kernel)**tupla*[0] ← *muestra[x]**tupla*[1] ← *muestra[y]**tupla*[2] ← *densidad**insertar tupla en listaDeTuplas***fin para****ordenar** *listaDeTuplas según densidad***retornar** *listaDeTuplas***fin**

y Roeder, sino también para los contornos creados por el usuario. Si se aplica a un contorno de Barnes y Roeder, esta técnica permite al usuario la identificación de los miembros del conjunto de puntos que caen dentro del contorno. Si este *brushing* se aplica a los contornos generados por el usuario permite identificar, con un simple clic, todos los puntos con densidades más altas que el percentil del contorno generado (ver figura 6.21).

En el ***brushing por contorno***, el usuario selecciona un conjunto de datos en una vista arbitraria y los puntos correspondientes, en los *triplots* y *scatterplots* se resaltan y se utilizan para crear un contorno. Se visualiza entonces un contorno cóncavo alrededor de los puntos seleccionados y es posible calcular las intersecciones con todos los demás contornos. De este modo también se pueden comparar las áreas de los contornos además del número de puntos en el contorno. La figura 6.21 muestra un ejemplo de un *brushing por percentil* para un contorno generado por el usuario, a la izquierda, y un *brushing por contorno* que encierra los puntos seleccionados, a la derecha.

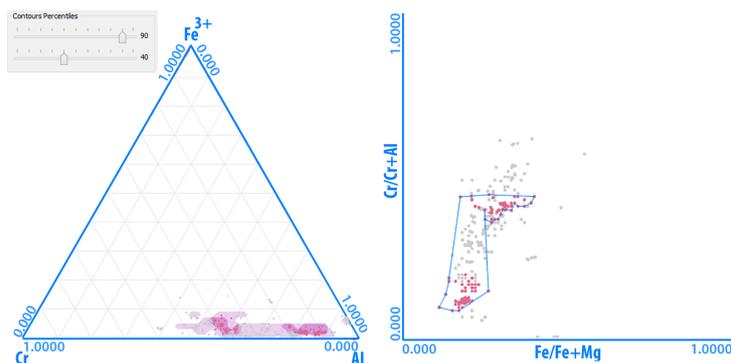


Figura 6.21: *Brushing* en el *Spinel Explorer*. Ejemplo de un *brushing por percentil* para un contorno generado por el usuario (a la izquierda) y un *brushing por contorno* que encierra los puntos seleccionados (a la derecha).

6.5.1. *Spinel Explorer* en el MUV

Dado que tanto en el *Spinel Explorer* como en el *SpinelViz* se exploran conjuntos de datos del mismo tipo (análisis químicos de minerales del grupo de los espinelos), las representaciones correspondientes y los procesos que atraviesan en el MUV en sus primeras etapas (desde DA hasta los DA) son similares para ambos prototipos (ver sección 6.4.1). La *Transformación de Filtrado*, sin embargo, será diferente en función de qué datos se desean mostrar efectivamente en la vista. El mapeo visual y la técnica de visualización,

por supuesto, también será diferente.

En la figura 6.18 se pueden observar cuatro de las vistas más significativas del *Spinel Explorer*. Para cada una de estas vistas, se generará una nueva rama en el *pipeline*, como se ilustra en la figura 6.22.

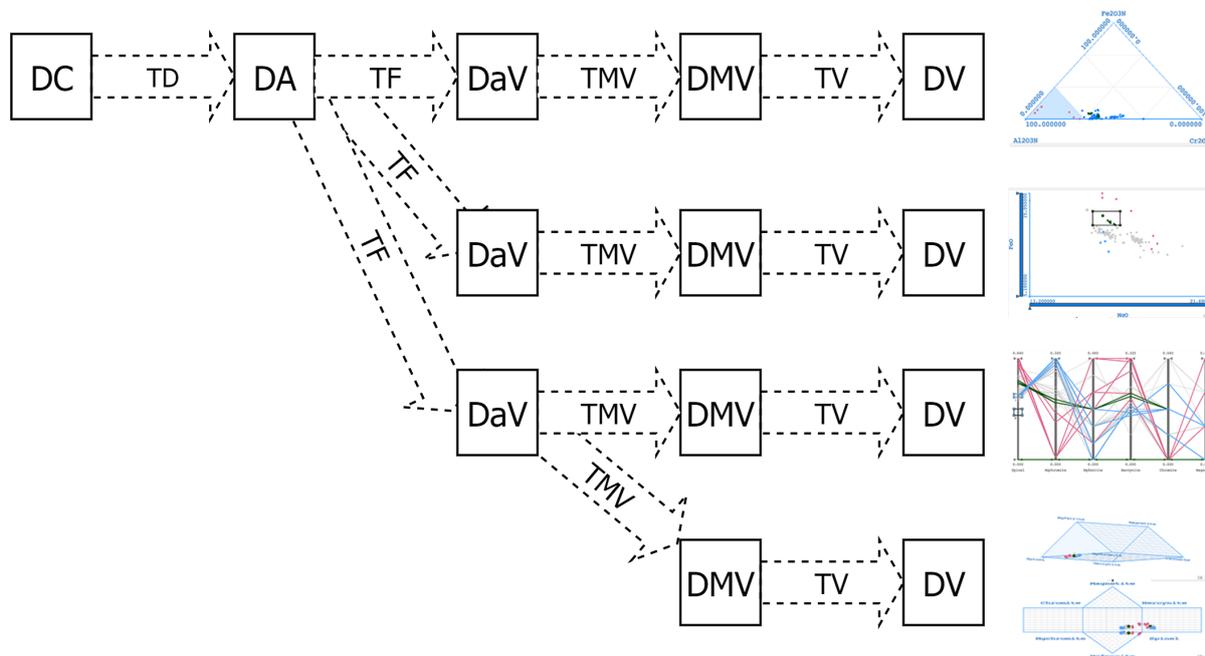


Figura 6.22: *Spinel Explorer* en el MUV: Una posible configuración del proceso generando cuatro vistas correlacionadas: un *tripplot*, un *scatterplot*, una vista de coordenadas paralelas y una vista 3D del Prisma de los Espinelos con sus proyecciones.

A continuación, se detallará el *Spinel Explorer* en el MUV para cada una de las vistas ilustradas en la figura 6.22 a partir de la *Transformación de Filtrado*.

Transformación de Filtrado

En esta transformación se seleccionarán aquellos datos que efectivamente van a estar presentes en la visualización. En cada una de las visualizaciones que se muestran en la figura 6.22, la selección de atributos es diferente.

- Triplot:** Para lograr esta visualización, se seleccionan de los DA únicamente tres atributos cuantitativos que pasarán a formar parte de los DaV, descartando todos los restantes. El filtrado es por atributos. En el caso del ejemplo, se han seleccionado los atributos correspondientes a los elementos mayoritarios: Aluminio (Al_2O_3), Cromo (Cr_2O_3) y Óxido Férrico (Fe_2O_3).

- **Scatterplot:** Para lograr esta visualización, se seleccionan de los DA únicamente dos atributos cuantitativos que pasarán a formar parte de los DaV, descartando todos los restantes. El filtrado es por atributos. En el caso del ejemplo, se han seleccionado los atributos correspondientes a los elementos mayoritarios: Magnesio (MgO) y Óxido Férrico (Fe_2O_3).
- **Coordenadas Paralelas:** Para lograr esta visualización, se seleccionan de los DA únicamente seis atributos cuantitativos que pasarán a formar parte de los DaV, descartando todos los restantes. El filtrado es por atributos. En el caso del ejemplo, se han seleccionado los atributos correspondientes a los miembros finales de la representación del Prisma de la Magnetita: Magnetita (Magnetita (Fe_3O_4)), Magnesioferrita (Magnesioferrita ($MgFe_2O_4$)), Espinela (Espinelo ($MgAl_2O_4$)), Hercinita (Hercinita ($FeAl_2O_4$)), Cromita Ferrocromita (Cromita ($FeCr_2O_4$)) y Magnesiocromita ($MgCr_2O_4$)).
- **Vista 3D:** Para lograr esta visualización, se seleccionan de los DA únicamente seis atributos cuantitativos que pasarán a formar parte de los DaV, descartando todos los restantes. El filtrado es por atributos. En el caso del ejemplo, al igual que para la visualización de coordenadas paralelas, se han seleccionado los atributos correspondientes a los miembros finales de la representación del Prisma de la Magnetita.

Transformación de Mapeo Visual

En la transformación de mapeo visual se generará una representación *mapeo visual* como la descrita en la sección 4.6 del capítulo 4. El *mapeo visual* estará compuesto por el conjunto de *datasets*, el sustrato espacial y una colección de asociaciones entre datos e ítems visuales. En cada una de las visualizaciones de la figura 6.22, el *mapeo visual* será diferente.

- **Transformación de Mapeo Visual para la Vista *Triplot***
 - Sustrato Espacial: Se tendrán de 3 ejes, organizados por composición, con orientación libre. Los tres ejes serán de tipo cuantitativo y con un rango establecido de 0 a 1.
 - Marcas: Todos los ítems de dato se representarán con la misma marca, un punto.

- Canales: Los canales que se utilizarán serán el tamaño y el color de cada elemento visual, que inicialmente se asociarán a valores por defecto. En la figura 6.23 se ilustra el *mapeo visual* resultante de esta transformación para la vista de *tripplot*.

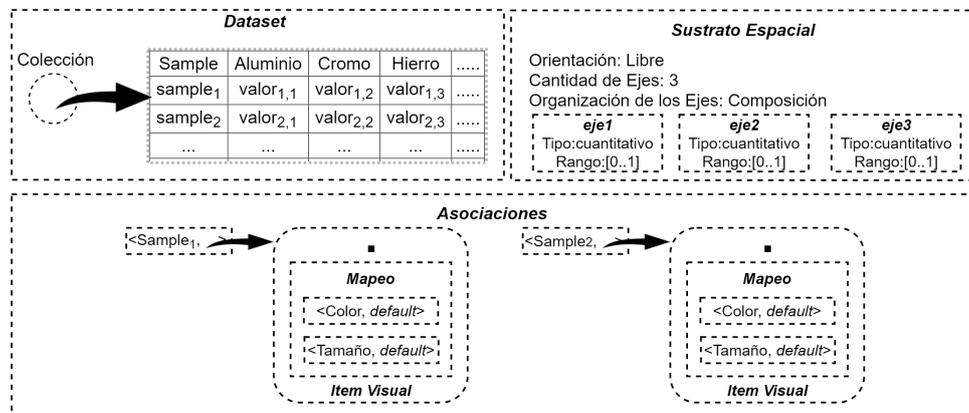


Figura 6.23: *Spinel Explorer* en el MUV. Mapeo Visual para la Vista de *Triplot*.

■ Transformación de Mapeo Visual para la Vista *Scatterplot*

- Sustrato Espacial: Se tendrán 2 ejes, organizados por composición, con orientación rectilínea. Los dos ejes serán de tipo cuantitativo y con un rango establecido de 0 a 1.
- Marcas: Todos los ítems de dato se representarán con la misma marca, un punto.
- Canales: Los canales que se utilizarán serán el tamaño y el color de cada elemento visual, que inicialmente se asociarán a valores por defecto. En la figura 6.24 se ilustra el *mapeo visual* resultante de esta transformación para la vista *scatterplot*.

■ Transformación de Mapeo Visual para la Vista Coordenadas Paralelas

- Sustrato Espacial: Se tendrán 6 ejes, organizados por alineación, con orientación paralela. Todos los ejes serán de tipo cuantitativo y con un rango establecido de 0 a 1.
- Marcas: Todos los ítems de dato se representarán con la misma marca, una polilínea.

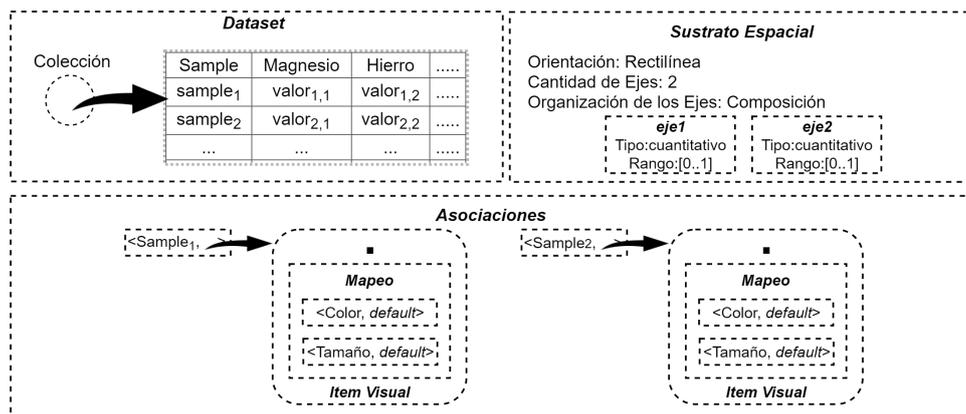


Figura 6.24: *Spinel Explorer* en el MUV. Mapeo Visual para la Vista de *Scatterplot*.

- **Canales:** El canal que se utilizará será el color de las marcas, que inicialmente se asociará a un valor por defecto.

En la figura 6.25 se ilustra el *mapeo visual* resultante de esta transformación para la vista de coordenadas paralelas.

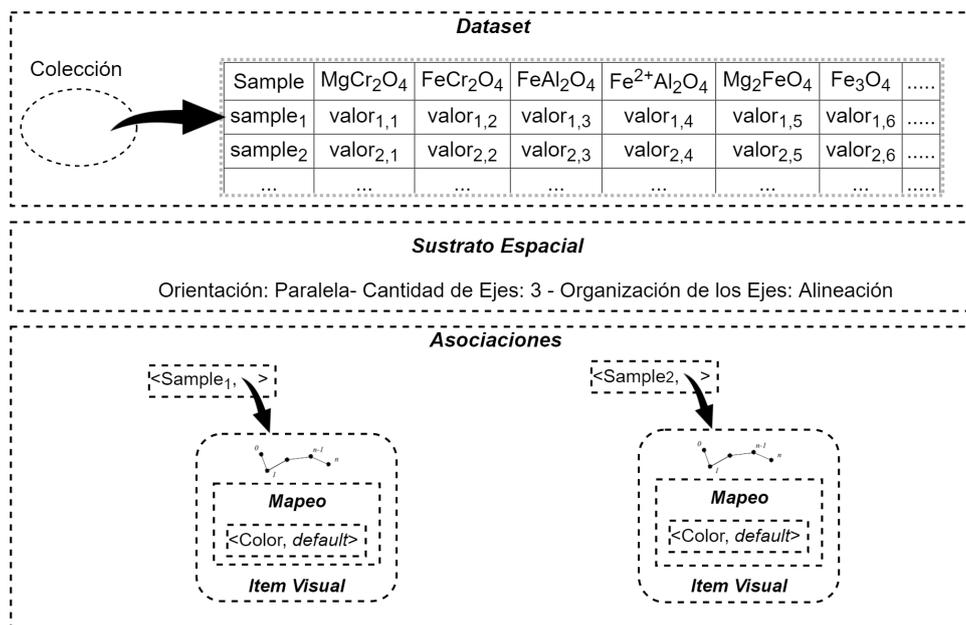


Figura 6.25: *Spinel Explorer* en el MUV: Mapeo Visual para la Vista de *Coordenadas Paralelas*.

■ **Transformación de Mapeo Visual para la 3D**

- **Sustrato Espacial:** Se tendrá una organización de ejes no estructurados.
- **Marcas:** Todos los ítems de dato se representarán con la misma marca,

una esfera.

- Canales: Los canales que se utilizarán serán el tamaño y el color de las marcas, que inicialmente se asociarán a valores por defecto.

En la figura 6.26 se ilustra el *mapeo visual* resultante de esta transformación para la vista 3D.

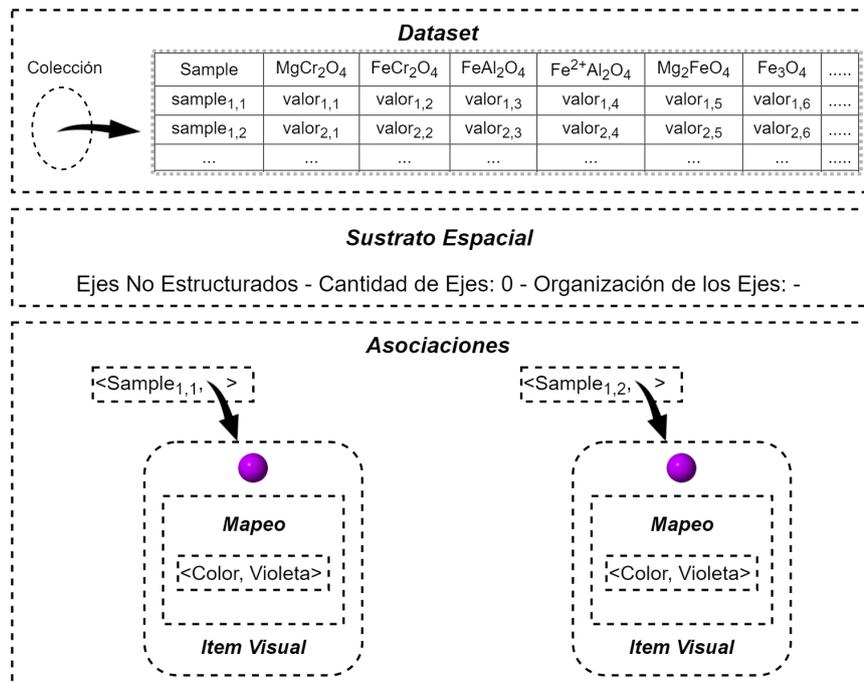


Figura 6.26: *Spinel Explorer* en el MUV. Mapeo Visual para la Vista 3D.

Datos Mapeados Visualmente

El conjunto de DMV generado en la Transformación de Mapeo Visual para cada una de las visualizaciones generadas se ilustra en las figuras 6.23 (*tripplot*), 6.24 (*scatterplot*), 6.25 (coordenadas paralelas) y 6.26 (vista 3D).

Transformación de Visualización

En esta transformación se aplica la técnica de visualización. La técnica aplicada dependerá de la visualización que se desea generar. Las técnicas de visualización de *Tripplot*, *Scatterplot* y Coordenadas Paralelas son técnicas estándar de visualización, que están soportadas por el *Spinel Explorer* y no consideramos necesario detallar su algoritmo. Para obtener la visualización del Prisma de los Espinelos y sus proyecciones, se aplicará una técnica similar a la descrita para el *SpinelViz* que se detalla en el algoritmo 1.

Datos Visualizados

El conjunto de datos visualizados es el producto final del proceso. En el caso particular del *Spinel Explorer* contamos con múltiples vistas correlacionadas. En este ejemplo en particular hemos seleccionado cuatro de sus vistas más significativas, cuyos Datos Visualizados se ilustran en la figura 6.27. Todas las vistas constituyen un punto de partida para el proceso de exploración que realizará el usuario.

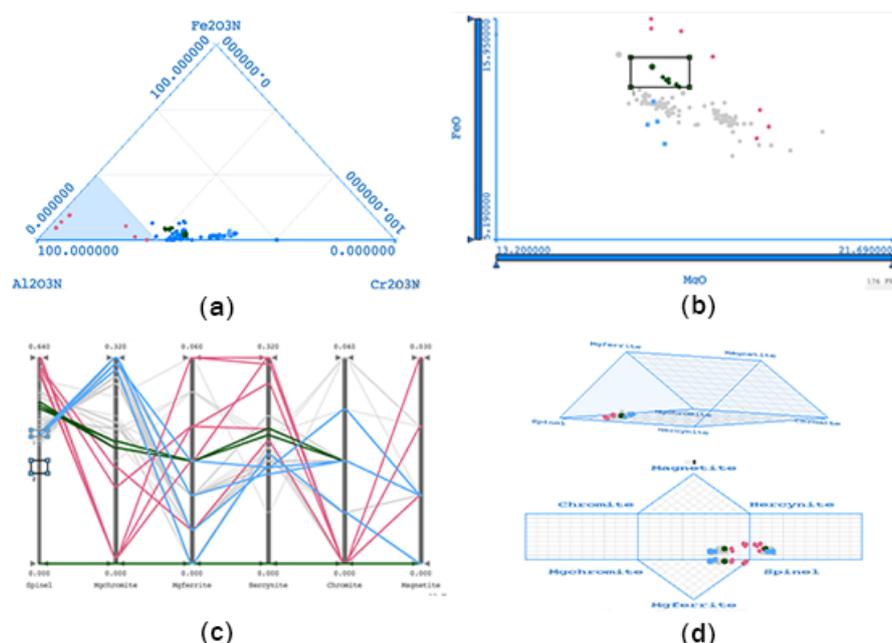


Figura 6.27: *Spinel Explorer* en el MUV. Datos Visualizados para cuatro de sus vistas más significativas: (a) *tripplot*, (b) *scatterplot*, (c) coordenadas paralelas y (d) vista 3D.

6.5.2. Interacciones

El *Spinel Explorer* provee un conjunto de interacciones generales e interacciones particulares para cada una de las vistas que soporta. Al igual que en el *SpinelViz* se cuenta con interacciones básicas que permiten manipular los conjuntos de datos que se desea visualizar, navegar sobre las vistas y configurar el mapeo visual. Adicionalmente, para el *Spinel Explorer* se han diseñado tanto un conjunto de interacciones de *brushing* novedosas como interacciones específicas del campo de aplicación para la categorización de espinelos para las vistas *scatterplot* y *tripplot*. A continuación se detalla un subconjunto de las interacciones provistas por el *Spinel Explorer* que consideramos más significativas en el contexto de esta tesis.

Interacciones generales para todas las vistas

- **Brushing y Linking:** El *Spinel Explorer* provee para todas sus vistas la interacción *Brushing y Linking* que permite seleccionar un subconjunto de los datos visualizados y resaltarlos en todas las vistas coordinadas. En particular, este *brushing* se puede realizar seleccionando un área cuadrada o circular (ver figura 6.18). El resaltado se logra modificando el color de la muestra. A nivel de las interacciones de usuario definidas en la sección 5.4 del capítulo 5, estas interacciones se traducen en *Brushing y Linking*, interacción compuesta a nivel de usuario. Estas interacción, si bien se inicia en la vista efectuando una selección del área a resaltar, se resuelve en la *Transformación de Mapeo Visual* pero es necesario volver al estado DA para poder resaltar los datos en las vistas correlacionadas (ver *Brushing y Linking* en la sección 5.4.4 del capítulo 5).
- **Codificar Marcas:** Esta interacción permite modificar el color y el tamaño asociados a todos los elementos mostrados en una vista en particular. A nivel de las interacciones de usuario definidas en la sección 5.4 del capítulo 5, esta interacción se traduce en *ConfigurarCanal*, interacción a nivel de usuario en el mapeo visual. Esta interacción se resuelve en la *Transformación de Mapeo Visual*.
- **Codificar Elementos Resaltados:** Esta interacción permite modificar el color asociado a los ítems resaltados mediante la interacción *Brushing y Linking* en todas las vistas coordinadas. El usuario puede seleccionar en todo momento con qué color desea resaltar los elementos seleccionados. Si no desea elegir un color, el sistema elige un color por defecto. A nivel de las interacciones de usuario definidas en la sección 5.4 del capítulo 5, esta interacción se traduce en *ConfigurarCanal*, interacción a nivel de usuario en el mapeo visual. Estas interacción, se resuelve en la *Transformación de Mapeo Visual*.
- **Cargar Conjunto de Datos:** Esta interacción permite cargar un nuevo conjunto de datos al *Spinel Explorer*. El conjunto de datos podrá reemplazar al existente, o integrarse con el mismo, modificando en cualquier caso el conjunto de DC. A nivel de las interacciones de usuario definidas en la sección 5.4 del capítulo 5, esta interacción se traduce en *AbrirConjuntoDatos*, interacción a

nivel de usuario sobre los conjuntos de datos que se resuelve en el estado DC.

- **Crear Nueva Vista:** Esta interacción permite incorporar una nueva vista al sistema de visualización. Esta interacción generará una ramificación del *pipeline* a partir del estado *Datos Abstractos*. A nivel de las interacciones de usuario definidas en la sección 5.4 del capítulo 5, esta interacción se traduce en *NuevaConfiguración()*, interacción de usuario sobre el mapeo visual que se resuelve en la *Transformación de Filtrado*.
- **Cerrar Vista:** Esta interacción permite cerrar una de las vistas abiertas en el sistema de visualización. Esta interacción eliminará una rama completa del *pipeline*. A nivel de las interacciones de usuario definidas en la sección 5.4 del capítulo 5, esta interacción se traduce en la interacción equivalente a *EliminarRama()*, que se resuelve en la transformación donde se ha generado la bifurcación de la rama eliminada.
- **Modificar Atributos Mostrados:** Esta interacción permite modificar los atributos asociados a los ejes para cualquiera de las vistas del *Spinel Explorer*. Esta interacción realizará un nuevo filtrado en la Transformación de Filtrado, incorporando al conjunto de DaV los nuevos atributos que se desea mostrar, y eliminando del mismo los restantes. A nivel de las interacciones de usuario definidas en la sección 5.4 del capítulo 5, esta interacción se traduce en *OcultarAtributos* y *MostrarAtributos*, interacciones a nivel de usuario sobre los conjuntos de datos que se resuelven en la *Transformación de Filtrado*.

Interacciones de categorización

Estas interacciones se aplican únicamente a las vistas *tripplot* y *scatterplot*, que se utilizarán para visualizar las proyecciones de los datos cargados por el usuario en el Prisma de los Espinelos y los contornos de Barnes y Roeder.

- **Mostrar/Ocultar Lista de Categorización:** Esta interacción permite mostrar u ocultar la lista de categorización. Esto involucra un proceso de selección de aquellos contornos de Barnes y Roeder que mejor se corresponden con los datos cargados por el usuario para la vista generada. Este proceso se realiza contando, para cada contorno de Barnes y Roeder, cuántos puntos del conjunto de datos visualizados se localizan en su interior y generando a partir de esto,

una lista de contornos ordenados, en función de esta cantidad de puntos, de mayor a menor. Este proceso se resuelve en la Transformada de Visualización, que ejecutará la técnica correspondiente a la vista (*tripplot* o *scatterplot*) incorporando la lista de categorización generada. A nivel de las interacciones de usuario, esta interacción se traduce en *SeleccionarTécnica*, interacción a nivel de usuario sobre la Transformación de Visualización.

- **Cargar Preview de Categorización:** Esta interacción permite mostrar u ocultar una vista previa de los contornos de Barnes y Roeder que mejor se corresponden con los datos. Esta vista previa se muestra debajo de la visualización correspondiente (*tripplot* o *scatterplot*) como una tira de pequeñas visualizaciones (*thumbnails*). Esto involucra un proceso similar al de la interacción *Mostrar Lista de Categorización* en el cual se genera la lista de categorización pero la misma no se muestra en la visualización, sino que se utiliza para seleccionar los contornos a partir de los cuales generar la vista previa. Este proceso se resuelve en la Transformada de Visualización, que ejecutará la técnica correspondiente a la vista (*tripplot* o *scatterplot*) incorporando una vista previa de los mejores contornos y mostrándolos debajo del *tripplot* o del *scatterplot* según corresponda. A nivel de las interacciones de usuario, esta interacción se traduce en *Seleccionar Técnica*, interacción a nivel de usuario sobre la Transformación de Visualización.
- **Solapar/Ocultar Contornos de Barnes y Roeder:** Esta interacción permite mostrar u ocultar un contorno de Barnes y Roeder en particular, solapándolo con los datos del usuario mostrados en la vista. Esta interacción se resuelve en la Técnica de Visualización, que ejecutará la técnica correspondiente a la vista (*tripplot* o *scatterplot*) incorporando el contorno seleccionado. A nivel de las interacciones de usuario, esta interacción se traduce en *SeleccionarTécnica*, interacción a nivel de usuario sobre la Transformación de Visualización.
- **Generar Contornos:** Esta interacción permite generar y mostrar un contorno de un cierto percentil generado a partir de los datos visualizados. La generación de este contorno se resuelve en el estado *Datos Abstractos*, donde se procesarán los mismos calculando el nuevo contorno. Este nuevo contorno se

almacenará como un nuevo *dataset* con formato *Geometría* (ver sección 4.5). A partir de la generación de este nuevo contorno, se modificará el mapeo visual asociando el nuevo contorno a las marcas correspondientes y configurando sus canales para que al aplicar la técnica de visualización el contorno se visualice correctamente. Este proceso se resuelve en el estado *Datos Abstractos*. A nivel de las interacciones de usuario, esta interacción se traduce en *AbrirConjunto-Datos*, interacción a nivel de usuario sobre el estado *Datos Abstractos*.

- **Brushing y Linking por Percentil:** Esta interacción permite hacer una selección, sobre la vista *triplot* o *scatterplot*, de todos aquellos datos que se encuentren dentro de un determinado percentil de uno de los contornos de Barnes y Roeder o de los contornos generados por el usuario sobre sus propios datos. Esta selección se hace efectiva en todas las vistas correlacionadas. Este proceso se resuelve, como cualquier *brushing y linking*, en los *Datos Abstractos*, estado al que hay que retornar para detectar todas las ramificaciones que dependan de los mismos datos. Sin embargo, el proceso de selección es más complejo que el *brushing y linking* convencional, ya que es necesario realizar un procesamiento de los datos, detectando todos aquéllos que se ubican dentro de un determinado percentil. Si la interacción se aplica sobre los contornos de Barnes y Roeder este proceso se lleva a cabo en la *Transformación de Visualización*. Si la interacción se aplica sobre un contorno generado por el usuario, entonces este proceso se lleva a cabo en el estado *Datos Abstractos*.
- **Brushing y Linking de Contorno:** Esta interacción permite hacer una selección, sobre la vista *triplot* o *scatterplot*, de todos aquellos datos que se encuentren dentro de un determinado percentil de uno de los contornos definidos por el usuario. Esta selección se hace efectiva en todas las vistas correlacionadas. Este proceso se resuelve, como cualquier *brushing y linking*, en los *Datos Abstractos*, estado al que hay que retornar para detectar todas las ramificaciones que dependan de los mismos datos. Sin embargo, para el proceso de selección es necesario realizar un procesamiento de los datos, detectando todos aquéllos que se ubican dentro de un determinado percentil. Este proceso se lleva a cabo en el estado *Datos Abstractos*, ya que los contornos definidos por el usuario serán parte del conjunto de DA.

Interacciones de la vista 3D

- **Navegación:** El *Spinel Explorer* provee interacciones básicas de escalado, rotación y traslación para su vista 3D. A nivel de usuario, estas interacciones se traducen en *Zoomming*, *Rotación* y *Panning*, todas interacciones a nivel de usuario sobre la vista. Estas interacciones afectan directamente a la representación de la vista y no tienen influencia sobre las etapas ni transformaciones anteriores del pipeline de visualización. Las interacciones de navegación sobre la vista 3D se resuelven en el estado *Datos Visualizados*.

En la figura 6.28 se ilustran las interacciones generales para todas las vistas del *Spinel Explorer*, indicando en qué etapa del pipeline se resuelven.

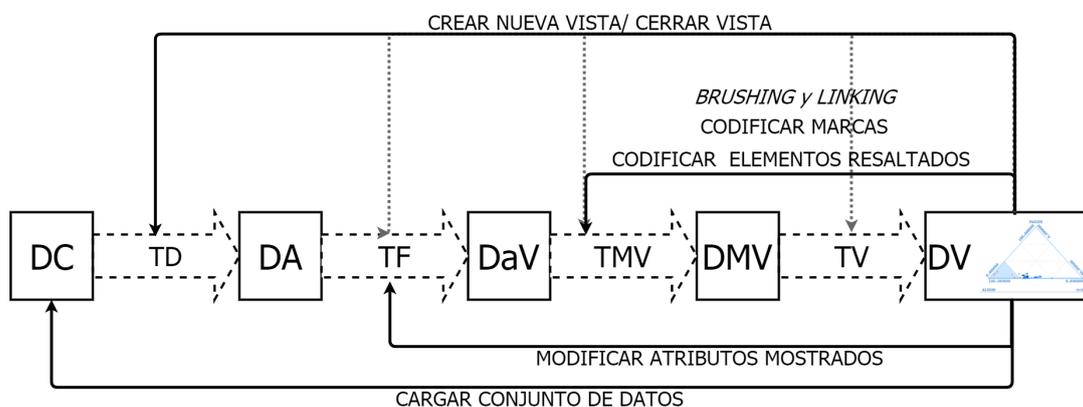


Figura 6.28: *Spinel Explorer* en el MUV. Interacciones generales para todas las vistas.

En la figura 6.29 se ilustran las interacciones de categorización del *Spinel Explorer* que se aplican a las vistas *scatterplot* y *tripplot*.

6.6. Conclusión

En este capítulo se han presentado el *SpinelViz* y el *Spinel Explorer*, dos prototipos de visualización de datos geológicos que sirvieron de base para el estudio y exploración de los conceptos introducidos en esta tesis. Inicialmente se describió el campo de aplicación, para luego situar ambos prototipos en el marco del Modelo Unificado de Visualización (MUV), describiendo su funcionalidad.

Para cada prototipo, se detalló la representación de los datos asumidos en cada estado y transformación del proceso de visualización, instanciando la representación para los

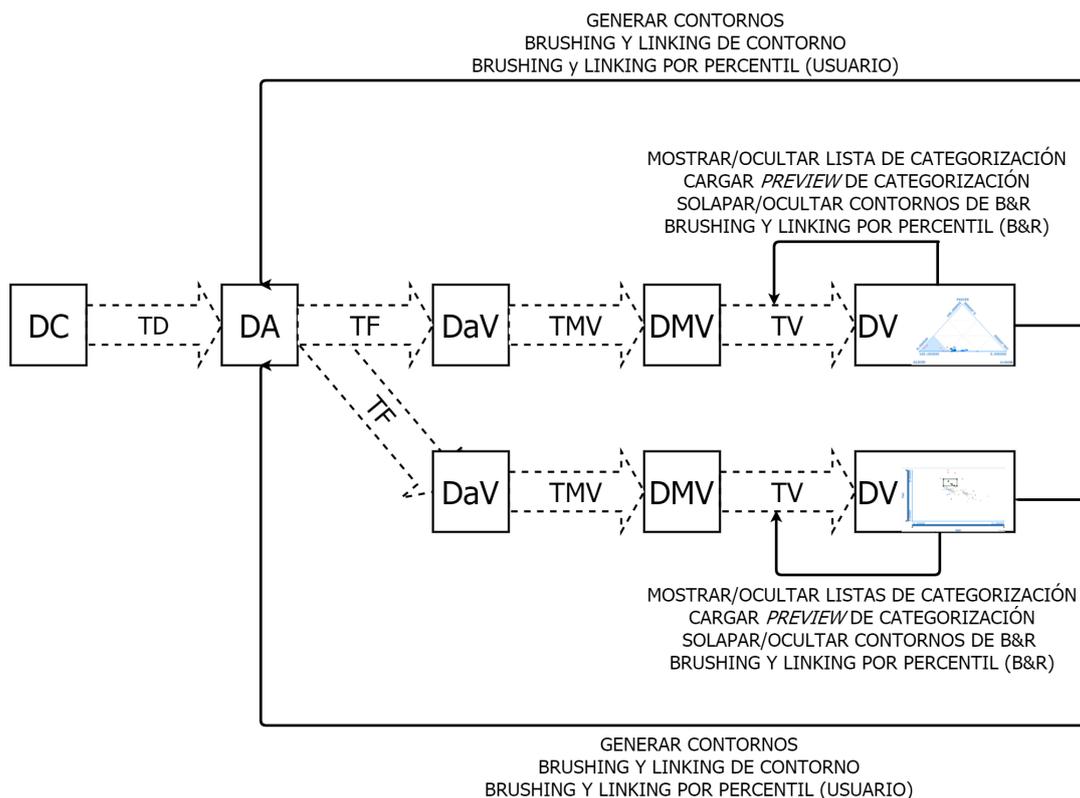


Figura 6.29: *Spinel Explorer* en el MUV. Interacciones de categorización.

datos, los atributos, los conjuntos de datos y el mapeo visual presentados en el capítulo 4.

Finalmente se detalló el conjunto de interacciones especialmente diseñadas para cada prototipo, detallando para cada una de éstas en qué estado o transformación del *pipeline* se resuelve y su implementación en términos de la clasificación de interacciones para visualización presentada en el capítulo 5.

Las interacciones incorporadas en ambos prototipos contribuyeron directamente en un avance significativo en el flujo de trabajo de los geólogos expertos. En el *SpinelViz*, la incorporación de interacciones para manipular la vista 3D permitieron integrar, por primera vez, la representación tridimensional del Prisma de los Espinelos con las proyecciones en dos dimensiones que los usuarios acostumbran a explorar, cómo así también enriquecieron el proceso de exploración incorporando *zoom* semántico directamente en el espacio prismático. En el *Spinel Explorer*, las interacciones diseñadas enriquecen considerablemente el proceso de clasificación de muestras de minerales. En este sistema, el *brushing y linking*, las interacciones específicas de clasificación y las de generación y manipulación de contornos reducen considerablemente el tiempo destinado al proceso de clasificación y permiten lograr mejores clasificaciones.

Principalmente, además se muestra cómo la clasificación de las interacciones y las operaciones definidas permiten ordenar y facilitar el desarrollo de un sistema de visualización en un determinado campo de aplicación.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 7

Conclusiones y Trabajo a Futuro

“It was easier to know it than to explain why I know it.”

— Sherlock Holmes, Sir Arthur
Conan Doyle

El objetivo común de los distintos procesos de visualización es la obtención de representaciones visuales interactivas con el propósito de ampliar la adquisición y el uso del conocimiento. Según la naturaleza y las características de la información a visualizar podemos hablar de dos grandes áreas de Visualización: Visualización Científica y Visualización de Información. Es claro que los requerimientos impuestos por cada dominio pueden llegar a ser dramáticamente diferentes; sin embargo, como expresáramos previamente, de un análisis cuidadoso de los mismos podemos encontrar denominadores comunes. En función de lo expuesto se definió el Modelo Unificado de Visualización (MUV), un modelo de visualización consistente para todas las áreas de visualización y válido para los diferentes dominios de aplicación.

La definición de este modelo tiene como objetivo orientar a los usuarios en cada una de las etapas del proceso de visualización para que éste pueda interactuar con la visualización de manera efectiva. Una herramienta de visualización debe ser lo suficientemente general y flexible como para soportar las necesidades individuales, ofreciendo un esquema de interacción consistente y válido independientemente del dominio de aplicación. Esta característica habilita a los usuarios a transferir el conocimiento del uso de la herramienta entre los distintos dominios basándose en un modelo conceptual consistente.

Esto se logra en parte con la elección de una metáfora adecuada (*representación*

visual + interacciones disponibles) encuadrada en un marco conceptual del proceso que le permita saber en cada momento en qué punto del proceso se encuentra y hacia dónde puede dirigirse.

A partir de lo expuesto surge nuestra propuesta de definición de un esquema de interacciones para el MUV, con el objetivo de dar soporte a la definición de las interacciones que se aplicarán en las distintas transformaciones y estados del proceso de visualización. Subyacente a esto, surge también la necesidad de definir una representación para los conjuntos de datos lo suficientemente flexible como para soportar las distintas clasificaciones de datos, atributos, conjuntos de datos y mapeos visuales presentes en la literatura.

A partir de esto, en esta tesis se han propuesto:

- **Una especificación de una clasificación para los datos, atributos y conjuntos de datos en el contexto de visualización.** Presentamos una clasificación y una representación para los datos, los conjuntos de datos y los atributos. El proceso de desarrollo de esta representación involucró un estudio profundo de los principales aportes en esta materia. La representación obtenida es general y considera las definiciones propuestas por los autores más relevantes en el área, logrando de este modo una integración que sienta las bases de un proceso de visualización independiente del campo de aplicación y del área de visualización. Tanto la clasificación como las representaciones obtenidas sirvieron de base para definir las interacciones que se llevan a cabo en cada estado y transformación del MUV.
- **Una especificación de una representación para el mapeo visual.** Se introduce una especificación para el mapeo visual que contempla tanto el almacenamiento del conjunto de datos a representar, como el de la estructura visual a ser utilizada para representar esos datos en pantalla. Como se mencionó anteriormente, la estructura visual está compuesta por el sustrato espacial y el sustrato gráfico. Se logra entonces definir una especificación para el mapeo visual que integre tanto la estructura visual como los datos a representar, necesaria para la definición de interacciones sobre éste. La especificación lograda integra y completa los conceptos presentes en los aportes más relevantes en el área.
- **Una clasificación multi-nivel de interacciones para visualización en el contexto del MUV.** Encontramos en el Modelo Unificado de Visualización (MUV)

un cuerpo troncal de interacciones operacionalmente similares entre los distintos dominios de aplicación contemplando las características distintivas de los distintos dominios y/o técnicas particulares y que constituye un modelo conceptual de interacción. La clasificación presentada se compone de dos clases principales, en función del usuario al que están destinadas: las interacciones a nivel del programador y las interacciones a nivel del usuario. Las interacciones a nivel de programador, a su vez, se clasifican en dos grupos: las interacciones de bajo nivel del programador y las interacciones de alto nivel del programador. Las interacciones de bajo nivel del programador se definieron en función de las representaciones para los conjuntos de datos y *mapeo visual* presentadas en el capítulo 4. Las interacciones de alto nivel del programador se definieron en función de las interacciones a bajo nivel del programador. Las interacciones a nivel del usuario se clasificaron en 4 clases principales, en función de la etapa del proceso de visualización que afectan: las interacciones a nivel del usuario sobre los conjuntos de datos, sobre el mapeo visual, sobre la vista y composiciones de estas tres. Las interacciones a nivel del usuario se definieron en base de las interacciones a nivel del programador, especificando para cada una en qué estado o transformación del proceso de visualización se resuelven.

- **El diseño e implementación de dos prototipos de visualización de datos geológicos y las interacciones asociadas.** Se diseñó e implementó el *SpinelViz* y el *Spinel Explorer*, dos prototipos de visualización de datos geológicos que sirvieron de base para el estudio y la exploración de los conceptos introducidos en esta tesis. Para cada prototipo se diseñó un conjunto de interacciones, detallando para cada una de éstas en qué estado o transformación del *pipeline* se resuelve, y su implementación en términos de la clasificación de interacciones para visualización presentada en el capítulo 5. Las interacciones incorporadas en ambos prototipos contribuyeron directamente en un avance significativo en el flujo de trabajo de los geólogos expertos. En el *SpinelViz*, la incorporación de interacciones para manipular la vista 3D, permitieron integrar, por primera vez, la representación tridimensional del Prisma de los Espinelos con las proyecciones en dos dimensiones que los usuarios acostumbran a explorar, como así también enriquecieron el proceso de exploración incorporando *zoom* semántico directamente en el espacio prismático. En el *Spinel Explorer*, las interacciones diseñadas enriquecen considerablemente el proceso de

clasificación de muestras de minerales. En este sistema, el *brushing y linking*, las interacciones específicas de clasificación y las de generación y manipulación de contornos reducen considerablemente el tiempo destinado al proceso de clasificación y permiten lograr mejores clasificaciones.

Gracias a los resultados obtenidos, hemos logrado definir un esquema de interacciones válida para el Modelo Unificado de Visualización, que establece tanto para el usuario como para el programador las interacciones disponibles en cada estado o transformación del proceso de visualización. De este modo, tanto el usuario como el programador conocen en todo momento qué es lo que pueden hacer, cómo hacerlo y cuáles serán las consecuencias de sus acciones sobre la representación visual y a nivel de los datos subyacentes, controlando de este modo el proceso de exploración de sus datos.

7.1. Publicaciones

A continuación se detallan los trabajos científicos más relevantes publicados durante el proceso de desarrollo de este trabajo de investigación:

1. Publicaciones en Revistas Indexadas

- 2012 *An interactive 3D application for visualizing spinel group minerals*. Computer and Geosciences, Elsevier. **M. L. Ganuza**, S. M. Castro, G. Ferracutti, E. A. Bjerg y S. R. Martig. Volumen 48, páginas 50–56. Noviembre 2012.
- 2014 *The Spinel Explorer - Interactive Visual Analysis of Spinel Group Minerals*. IEEE Transactions on Visualization and Computer Graphics, IEEE Computer Society. **M. L. Ganuza**, G. Ferracutti, M. F. Gargiulo, S. M. Castro, E. Bjerg, E. Gröller y K. Matković. Volumen 20, tomo 12, páginas 1912–1922. Diciembre 2014.
- 2015 *Determination of the Spinel Group End-Members Based On Electron Microprobe Analyses*. Mineralogy and Petrology, Springer. G. Ferracutti, M. F. Gargiulo, **M. L. Ganuza**, E. A. Bjerg y S. M. Castro. Volumen 109, tomo 2, páginas 153-160. Abril 2015.

2. Publicaciones en Conferencias Internacionales

- 2015 *Interactive Semi-Automatic Categorization for Spinel Group Minerals*. IEEE Conference on Visual Analytics Science and Technology (IEEE VAST 2015). **M. L. Ganuza**, M. F. Gargiulo, G. Ferracutti, S. M. Castro, E. Bjerg, E. Gröller y K. Matković. Chicago, Illinois, Estados Unidos. Octubre de 2015.
- 2017 *Interactive Visual Categorization of Spinel-Group Minerals*. 33rd Spring Conference on Computer Graphics (SCCG 2017). **M. L. Ganuza**, G. Ferracutti, M. F. Gargiulo, S. M. Castro, E. Bjerg, E. Gröller y K. Matković. Mikulov, República Checa. Mayo de 2017.
- 2017 *Immersive Analytics for Geology: Field Sketch-Like Visualization to assist geological structure analysis during fieldwork*. IEEE VIS - Workshop on Immersive Analytics. N. F. Gazcón, J. M. Trippel, J. I. Larregui, **M. L. Ganuza**, E. A. Bjerg y S. M. Castro. Phoenix, USA. Septiembre de 2017.

3. Publicaciones en Congresos Nacionales

- 2009 *Mineral Compositions Visualization Implementing the Spinel Prism*. XV Congreso Argentino de Ciencias de la Computación (CACIC 2009). **M. L. Ganuza**, S. M. Castro, S. R. Martig, G. Ferracutti y E. Bjerg. Jujuy, Argentina. Octubre de 2009.
- 2014 *Visualización de la composición de los minerales del grupo del espinelo en 3D: nueva herramienta para su aplicación como trazadores tectónicos*. XIX Congreso Geológico Argentino. G. Ferracutti, M. F. Gargiulo, **M. L. Ganuza**, S. M. Castro y E. A. Bjerg. Córdoba, Argentina. Junio de 2014.
- 2015 *Análisis Visual en Geología*. XVII Workshop de Investigadores en Ciencias de la Computación (WICC 2015). **M. L. Ganuza**, G. Ferracutti, M. F. Gargiulo, S. M. Castro, K. Matković, E. Bjerg y E. Gröller. Salta, Argentina. Abril de 2015.
- 2017 *Visualización y Realidad Aumentada en el campo de las Ciencias Geológicas*. XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017). **M. L. Ganuza**, J. M. Trippel, N. F. Gazcón, S. M. Castro, E. Bjerg, F.

Gargiulo, G. Ferracutti, K. Matković y E. Gröller. Buenos Aires, Argentina.
Abril de 2017.

7.2. Direcciones Futuras de Investigación

Este trabajo es el punto de inicio para múltiples líneas de investigación. A continuación describiremos las que consideramos más relevantes:

■ Definición de una Taxonomía de Tareas en el Contexto del MUV

Si bien este trabajo de investigación está enfocado en la definición de una clasificación de interacciones en el contexto de visualización, consideramos que una posible futura extensión de la clasificación presentada debe incluir una taxonomía de tareas de usuario. En el contexto de visualización, las tareas de usuario pueden definirse como las actividades que se llevan a cabo interactivamente sobre una representación visual con un objetivo en particular. En la literatura existen varias clasificaciones, taxonomías y *frameworks* para describirlas [ZF98, AES05, SNHS13, BM13]; sin embargo, éstas son disjuntas, y están pensadas específicamente para Visualización de Información. Un espacio de diseño para las tareas del usuario, que permita descripciones concretas de tareas y secuencias de tareas, facilitaría la comparación y enriquecería la expresión idiomática de la visualización a través de todos los dominios de aplicación. La inclusión de una taxonomía para las tareas de usuario en el contexto del MUV implicará un estudio profundo de las distintas sub-áreas de la visualización para encontrar un conjunto de tareas comunes a los distintos dominios de aplicación. Estas tareas se definirán en función de las interacciones a nivel de usuario y por encima de la clasificación de interacciones presentada en este trabajo de tesis.

■ Extensión del MUV para Datos Temporales

Los datos temporales son aquéllos en los que los cambios a lo largo del tiempo o los aspectos temporales juegan un rol central o son de interés. Los datos temporales son complicados de manejar dado que naturalmente pensamos el tiempo como una estructura jerárquica y potencialmente periódica. Muchas técnicas de visualización tratan al tiempo simplemente como un dato cuantitativo; sin embargo, para crear representaciones visuales eficientes y que permitan el razonamiento sobre el tiempo,

es preciso que los métodos de visualización tengan en cuenta las características particulares de tiempo.

Si bien el MUV soporta el procesamiento de datos temporales en su estado actual, no está orientado para tratar los mismos en todo su potencial. En trabajos relativamente recientes se han comenzado a formalizar las operaciones y las tareas relacionadas con este tipo de datos [AMST11]. Estos trabajos consideran, por ejemplo, la estructura multiescala del tiempo, donde la escala de interés puede variar desde nanosegundos a horas, de horas a décadas y de décadas a miles de años. Las tareas de análisis temporal a menudo involucran la búsqueda o verificación de la periodicidad, de un elemento de dato en un determinado momento, de ubicación temporal, de textura temporal, de velocidad de cambio, etc. Dadas su estructura jerárquica compleja y las tareas específicas asociadas al mismo, la incorporación de los datos temporales al MUV es un desafío importante.

Se plantea como trabajo a futuro dar soporte a los datos temporales tomando en cuenta sus características especiales. Se plantea extender la representación de los datos con el objetivo de poder soportar su estructura jerárquica específica, diseñando interacciones tanto a nivel del programador como a nivel del usuario que soporten las tareas definidas para este tipo de datos.

- **Extensión del MUV para Soportar Un Historial de Interacciones** El resultado de una visualización no es solamente la vista o vistas obtenidas, sino todas y cada una de las interacciones realizadas para su generación. Esto determina la necesidad e importancia de registrar la secuencia de interacciones sobre los distintos elementos del proceso, información que podemos denominar “Historial de Interacciones”. Este registro debe mantener toda la información necesaria para posibilitar la obtención de los mismos resultados, independientemente del estado del proceso, en el momento en que se requiera. Es decir, se debe mantener la información necesaria para poder reconstruir cualquiera de los estados del proceso.

Se plantea entonces, como trabajo futuro, la extensión del MUV para soportar un historial de interacciones. Como el proceso de visualización es interactivo e iterativo, para cada operación registrada se deberá almacenar el estado sobre el cual se aplica y con qué parámetros. Resulta claro que el orden de aplicación es clave, y que las

secuencias de interacciones deberían tratarse como datos temporales y definirse las interacciones necesarias para su exploración.

Bibliografía

- [AES05] AMAR, R., EAGAN, J., AND STASKO, J. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* (2005), IEEE, pp. 111–117.
- [AMB⁺03] ARTIMO, A., MÄKINEN, J., BERG, C. R., ABERT, C. C., AND SALONEN, V.-P. Three-dimensional geologic modeling and visualization of the virttaankangas aquifer, southwestern finland. *Hydrogeology Journal* 11, 3 (2003), 378–386.
- [AMST11] AIGNER, W., MIKSCH, S., SCHUMANN, H., AND TOMINSKI, C. *Visualization of time-oriented data.* Springer Science & Business Media, 2011.
- [AS94a] AHLBERG, C., AND SHNEIDERMAN, B. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1994), ACM, pp. 313–317.
- [AS94b] AHLBERG, C., AND SHNEIDERMAN, B. Visual information seeking using the filmfinder. In *Conference companion on Human factors in computing systems* (1994), ACM, pp. 433–434.
- [Asi85] ASIMOV, D. The grand tour: a tool for viewing multidimensional data. *SIAM journal on scientific and statistical computing* 6, 1 (1985), 128–143.
- [AT95] ABRAM, G., AND TREINISH, L. An extended data-flow architecture for data analysis and visualization. In *Proceedings of the 6th Conference on Visualization '95* (Washington, DC, USA, 1995), VIS '95, IEEE Computer Society, pp. 263–.

- [AW90] ABRAM, G. D., AND WHITTED, T. Building block shaders. *SIGGRAPH Comput. Graph.* 24, 4 (September 1990), 283–288.
- [BCC⁺05] BAVOIL, L., CALLAHAN, S. P., CROSSNO, P. J., FREIRE, J., SCHEIDEGGER, C. E., SILVA, C. T., AND VO, H. T. Vistrails: Enabling interactive multiple-view visualizations. In *IEEE Visualization '05* (2005), pp. 135–142.
- [BCS96] BUJA, A., COOK, D., AND SWAYNE, D. F. Interactive high-dimensional data visualization. *Journal of computational and graphical statistics* 5, 1 (1996), 78–99.
- [BCW87] BECKER, R. A., CLEVELAND, W. S., AND WILKS, A. R. Dynamic graphics for data analysis. *Statistical Science* (1987), 355–383.
- [BE96] BALL, T., AND EICK, S. G. Software visualization in the large. *Computer* 29, 4 (1996), 33–43.
- [Ber77] BERTIN, J. *La graphique et le traitement graphique de l'information*. No. 91 (084.21) BER. 1977.
- [Ber83] BERTIN, J. *Semiology of graphics: diagrams, networks, maps*.
- [Ber07] BERNHARDT, H.-J. Mincalc-v5, a software tool for mineral analyses data processing. *Acta Microscopica* 16, 1-2 (2007), 43–44.
- [BGB⁺00] BURFORD, D. J., GER, L., BLAKE, E. H., WIT, M. J. D., DOUCOURÉ, C. M., AND HART, R. J. A seismic modelling environment as a research and teaching tool for 3-d subsurface modelling. *International Journal of Applied Earth Observation and Geoinformation* 2 (2000), 69–77.
- [BHP⁺12] BLOCK, F., HORN, M. S., PHILLIPS, B. C., DIAMOND, J., EVANS, E. M., AND SHEN, C. The deeptree exhibit: Visualizing the tree of life to facilitate informal learning. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2789–2798.

- [BKH05] BENDIX, F., KOSARA, R., AND HAUSER, H. Parallel sets: visual analysis of categorical data. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on* (2005), IEEE, pp. 133–140.
- [BLoA91] BANERJEE, S. K., LINDSLEY, D. H., AND OF AMERICA, M. S. *Oxide minerals : petrologic and magnetic significance*, 1991. Replaces Reviews in mineralogy, v. 3, and Mineralogical Society of America short course notes, v. 3.
- [BM13] BREHMER, M., AND MUNZNER, T. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2376–2385.
- [BR01] BARNES, S. J., AND ROEDER, P. L. The range of spinel compositions in terrestrial mafic and ultramafic rocks. *Journal of Petrology* 42, 12 (2001), 2279–2302.
- [BSP⁺93] BIER, E. A., STONE, M. C., PIER, K., BUXTON, W., AND DEROSE, T. D. Toolglass and magic lenses: the see-through interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques* (1993), ACM, pp. 73–80.
- [Car02] CARR, M. Igpert for windows. *Terra Softa Inc., Somerset, NJ* (2002).
- [CDD⁺97] CASTRO, S. M., DANZI, M., DELRIEUX, C., LARREA, M., AND SILVETTI, A. Low-cost volume visualization. *Proceedings International Conference on Imaging Science, Systems, and Technology* (1997), 486–493.
- [Cle93] CLEVELAND, W. S. *Visualizing Data*. Hobart Press, 1993.
- [CM97] CARD, S. K., AND MACKINLAY, J. The structure of the information visualization design space. In *Information Visualization, 1997. Proceedings., IEEE Symposium on* (1997), IEEE, pp. 92–99.
- [CMS99] CARD, S. K., MACKINLAY, J. D., AND SHNEIDERMAN, B. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

- [CNM83] CARD, S. K., NEWELL, A., AND MORAN, T. P. The psychology of human-computer interaction.
- [Coo84] COOK, R. L. Shade trees. *SIGGRAPH Comput. Graph.* 18, 3 (January 1984), 223–231.
- [CR96] CHUAH, M. C., AND ROTH, S. F. On the semantics of interactive visualizations. In *Information Visualization'96, Proceedings IEEE Symposium on* (1996), IEEE, pp. 29–36.
- [CR98] CHI, E. H.-H., AND RIEDL, J. T. An operator interaction framework for visualization systems. In *Information Visualization, 1998. Proceedings. IEEE Symposium on* (1998), IEEE, pp. 63–70.
- [CRMK95] CHUAH, M. C., ROTH, S. F., MATTIS, J., AND KOLOJEJCHICK, J. Sdm: Selective dynamic manipulation of visualizations. In *Proceedings of the 8th annual ACM symposium on User interface and software technology* (1995), ACM, pp. 61–70.
- [CWL96] CARR, D., WEGMAN, E., AND LUO, Q. Explorn: Design considerations past and present. *Center for Computational Statistics, George Mason University*, 129 (1996).
- [DE98] DIX, A., AND ELLIS, G. Starting simple: adding value to static visualisation through simple interaction. In *Proceedings of the working conference on Advanced visual interfaces* (1998), ACM, pp. 124–134.
- [Dix09] DIX, A. *Human-computer interaction*. Springer, 2009.
- [DRRD12] DÖRK, M., RICHE, N. H., RAMOS, G., AND DUMAIS, S. Pivotpaths: Strolling through faceted information spaces. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2709–2718.
- [Duk01] DUKE, D. Modular techniques in information visualization. In *Proceedings of the 2001 Asia-Pacific symposium on Information visualisation-Volume 9* (2001), Australian Computer Society, Inc., pp. 11–18.

- [Dye90] DYER, D. S. A dataflow toolkit for visualization. *IEEE Computer Graphics and Applications* 10, 4 (July 1990), 60–69.
- [Eic94] EICK, S. G. Data visualization sliders. In *Proceedings of the 7th annual ACM symposium on User interface software and technology* (1994), ACM, pp. 119–120.
- [EJFŠ03] ERBAN, V., JANOUSEK, V., FARROW, C., AND ŠMÍD, J. Geochemical data toolkit (gcdkit): a key for magmatic geochemists to the treasury of data analysis, statistics and graphics in r. *Geolines* 16 (2003), 25–26.
- [ESS92] EICK, S., STEFFEN, J. L., AND SUMNER, E. E. Seesoft—a tool for visualizing line oriented software statistics. *IEEE Transactions on Software Engineering* 18, 11 (1992), 957–968.
- [F⁺90] FOLEY, J. D., ET AL. *Computer Graphics: Principles and Practice*. The Systems Programming Series. Addison-Wesley Publishing Company, USA, 1990.
- [FB90] FEINER, S. K., AND BESHERS, C. Worlds within worlds: Metaphors for exploring n-dimensional virtual worlds. In *Proceedings of the 3rd annual ACM SIGGRAPH symposium on User interface software and technology* (1990), ACM, pp. 76–83.
- [Fou95] FOULSER, D. Iris explorer: A framework for investigation. *SIGGRAPH Comput. Graph.* 29, 2 (May 1995), 13–16.
- [FS95] FISHKIN, K., AND STONE, M. C. Enhanced dynamic queries via movable filters. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1995), ACM Press/Addison-Wesley Publishing Co., pp. 415–420.
- [Fur86] FURNAS, G. W. *Generalized fisheye views*, vol. 17. ACM, 1986.
- [FWR00] FUA, Y.-H., WARD, M. O., AND RUNDENSTEINER, E. A. Structure-based brushes: A mechanism for navigating hierarchically organized data and information spaces. *IEEE Transactions on visualization and computer graphics* 6, 2 (2000), 150–159.

- [GCF⁺12] GANUZA, M. L., CASTRO, S. M., FERRACUTTI, G., BJERG, E. A., AND MARTIG, S. Spinelviz: An interactive 3d application for visualizing spinel group minerals. *Computers & Geosciences* 48 (2012), 50–56.
- [GFG⁺14] GANUZA, M. L., FERRACUTTI, G., GARGIULO, M. F., CASTRO, S. M., BJERG, E. A., GRÖLLER, E., AND MATKOVIC, K. The spinel explorer - interactive visual analysis of spinel group minerals. *IEEE Trans. Vis. Comput. Graph.* 20, 12 (2014), 1913–1922.
- [GFG⁺17] GANUZA, M. L., FERRACUTTI, G., GARGIULO, M. F., CASTRO, S. M., BJERG, E. A., GRÖLLER, E., AND MATKOVIC, K. Interactive visual categorization of spinel-group minerals. *Proceedings of the Spring Conference on Computer Graphics* (2017).
- [GGF⁺15] GANUZA, M. L., GARGIULO, M. F., FERRACUTTI, G., CASTRO, S. M., BJERG, E. A., GRÖLLER, E., AND MATKOVIC, K. Interactive semi-automatic categorization for spinel group minerals. 197–198.
- [Gil16] GILLIES, M. What is movement interaction in virtual reality for? In *Proceedings of the 3rd International Symposium on Movement and Computing* (2016), ACM, p. 31.
- [GM00] GRAHAM, D. J., AND MIDGLEY, N. G. Technical communication-graphical representation of particle shape using triangular diagrams: An excel spreadsheet method. *Earth Surface Processes and Landforms* 25, 13 (2000), 1473–1478.
- [GR94] GOLDSTEIN, J., AND ROTH, S. F. Using aggregation and dynamic queries for exploring large data sets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (1994), ACM, pp. 23–29.
- [GSF⁺97] GAINES, R. V., SKINNER, H. C. W., FOORD, E. E., MASON, B., AND ROSENZWEIG, M. *Dana's New Mineralogy*. John Wiley & Sons Inc, 1997.
- [HA04] HENDERSON, A., AND AHRENS, J. *The Paraview guide : a parallel visualization application*. Kitware, Inc., New York, 2004.

- [Hae88] HAEBERLI, P. E. Conman: A visual programming language for interactive graphics. *SIGGRAPH Comput. Graph.* 22, 4 (June 1988), 103–111.
- [Hag76] HAGGERTY, S. Opaque mineral oxides in terrestrial igneous rocks. *Oxide Minerals. Mineralogical Society of America, Washington, DC* (1976), 101–277.
- [HD12] HEALEY, C. G., AND DENNIS, B. M. Interest driven navigation in visualization. *IEEE transactions on visualization and computer graphics* 18, 10 (2012), 1744–1756.
- [Hil91] HILS, D. D. Datavis: A visual programming language for scientific visualization. In *Proceedings of the 19th Annual Conference on Computer Science* (New York, NY, USA, 1991), CSC '91, ACM, pp. 439–448.
- [HS12] HEER, J., AND SHNEIDERMAN, B. Interactive dynamics for visual analysis. *Queue* 10, 2 (2012), 30.
- [HWC⁺05] HO, C., WU, F.-C., CHEN, B.-Y., CHUANG, Y.-Y., OUHYOUNG, M., ET AL. Cubical marching squares: Adaptive feature preserving surface extraction from volume data. In *Computer graphics forum* (2005), vol. 24, Wiley Online Library, pp. 537–545.
- [IFP⁺12] ISENBERG, P., FISHER, D., PAUL, S. A., MORRIS, M. R., INKPEN, K., AND CZERWINSKI, M. Co-located collaborative visual analytics around a tabletop display. *IEEE Transactions on visualization and Computer Graphics* 18, 5 (2012), 689–702.
- [Irv65] IRVINE, T. Chromian spinel as a petrogenetic indicator: Part 1. theory. *Canadian Journal of Earth Sciences* 2, 6 (1965), 648–672.
- [Jac67] JACQUES, B. Sémiologie graphique: les diagrammes, les réseaux, les cartes. *Paris, École des hautes études en sciences sociales/Mouton-Gauthier-Villars* (1967).
- [JGH⁺08] JACOB, R. J., GIROUARD, A., HIRSHFIELD, L. M., HORN, M. S., SHAER, O., SOLOVEY, E. T., AND ZIGELBAUM, J. Reality-based interaction: a

- framework for post-wimp interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2008), ACM, pp. 201–210.
- [JMIntISC] JOHNSON, H. J., MCCORMICK, M. M., IBÁÑEZ, L., AND THE INSIGHT SOFTWARE CONSORTIUM. *The ITK Software Guide*, 4 ed. Updated for ITK version 4.9.
- [JS91] JOHNSON, B., AND SHNEIDERMAN, B. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the 2nd conference on Visualization'91* (1991), IEEE Computer Society Press, pp. 284–291.
- [Kas92] KASS, M. Condor: Constraint-based dataflow. *SIGGRAPH Comput. Graph.* 26, 2 (July 1992), 321–330.
- [Kei02] KEIM, D. A. Information visualization and visual data mining. *IEEE transactions on Visualization and Computer Graphics* 8, 1 (2002), 1–8.
- [KHG03] KOSARA, R., HAUSER, H., AND GRESH, D. L. An interaction view on information visualization. *State-of-the-Art Report. Proceedings of EUROGRAPHICS* (2003), 123–137.
- [KKDF97] KIPLING, R., KIPLING, J. L., DRAKE, W., AND FRENZENY, P. *The jungle book*. Heinemann and Balestier, 1897.
- [KLM⁺04] KWAN, M.-P., LEE, J., MICHAEL, I., GOODCHILD, F., AND JANELLE, D. G. Geovisualization of human activity patterns using 3d gis: A time-geographic approach. In *Spatially Integrated Social Science* (2004), University Press, pp. 48–66.
- [KLS00] KREUSELER, M., LOPEZ, N., AND SCHUMANN, H. A scalable framework for information visualization. In *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on* (2000), IEEE, pp. 27–36.
- [KS94] KOELMA, D., AND SMEULDERS, A. A visual programming interface for an image processing environment. *Pattern Recognition Letters* 15, 11 (1994), 1099 – 1109.

- [KS12] KERREN, A., AND SCHREIBER, F. Toward the role of interaction in visual analytics. In *Proceedings of the Winter Simulation Conference (2012)*, Winter Simulation Conference, p. 420.
- [LAC⁺92] LUCAS, B., ABRAM, G. D., COLLINS, N. S., EPSTEIN, D. A., GRESH, D. L., AND MCAULIFFE, K. P. An architecture for a scientific visualization system. In *Proceedings of the 3rd Conference on Visualization '92 (Los Alamitos, CA, USA, 1992)*, VIS '92, IEEE Computer Society Press, pp. 107–114.
- [LCWL14] LIU, S., CUI, W., WU, Y., AND LIU, M. A survey on information visualization: recent advances and challenges. *The Visual Computer* 30, 12 (2014), 1373–1393.
- [LG95] LEE, J. P., AND GRINSTEIN, G. An architecture for retaining and analyzing visual explorations of databases. In *Proceedings of the 6th conference on Visualization'95 (1995)*, IEEE Computer Society, p. 101.
- [LGH⁺96] LIVINGSTON, M. A., GARRETT, W. F., HIROTA, G., WHITTON, M. C., PISANO, E. D., FUCHS, H., ET AL. Technologies for augmented reality systems: Realizing ultrasound-guided needle biopsies. In *Proceedings of the 23rd annual conference on computer graphics and interactive techniques (1996)*, ACM, pp. 439–446.
- [LRP95] LAMPING, J., RAO, R., AND PIROLI, P. A focus+ context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the SIGCHI conference on Human factors in computing systems (1995)*, ACM Press/Addison-Wesley Publishing Co., pp. 401–408.
- [LSH⁺15] LEIGH, S.-W., SCHOESSLER, P., HEIBECK, F., MAES, P., AND ISHII, H. Thaw: tangible interaction with see-through augmentation for smartphones on computer screens. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction (2015)*, ACM, pp. 89–96.
- [Lun96] LUNZER, A. *Reconnaissance: a widely applicable approach encouraging well-informed choices in computer-based tasks*. PhD thesis, Citeseer, 1996.

- [LZP⁺09] LIU, S., ZHOU, M. X., PAN, S., QIAN, W., CAI, W., AND LIAN, X. Interactive, topic-based visual text summarization and analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management (2009)*, ACM, pp. 543–552.
- [LZP⁺12] LIU, S., ZHOU, M. X., PAN, S., SONG, Y., QIAN, W., CAI, W., AND LIAN, X. Tiara: Interactive, topic-based visual text summarization and analysis. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 2 (2012), 25.
- [Mac95] MACEACHREN, A. M. *How maps work: representation, visualization, and design*. Guilford Press, 1995.
- [Mar96] MARSHALL, D. Ternplot: An excel spreadsheet for ternary diagrams. *Computers & Geosciences* 22, 6 (1996), 697–699.
- [MB95] MUNZNER, T., AND BURCHARD, P. Visualizing the structure of the world wide web in 3d hyperbolic space. In *Proceedings of the first symposium on Virtual reality modeling language (1995)*, ACM, pp. 33–38.
- [McC88] MCCORMICK, B. H. Visualization in scientific computing. *SIGBIO Newsl.* 10, 1 (March 1988), 15–21.
- [MCFE03] MARTIG, S. R., CASTRO, S. M., FILLOTTRANI, P. R., AND ESTEVEZ, E. C. Un modelo unificado de visualización. In *IX Congreso Argentino de Ciencias de la Computación (2003)*.
- [MFGH08] MATKOVIC, K., FREILER, W., GRACANIN, D., AND HAUSER, H. Comvis: A coordinated multiple views system for prototyping new visualization technology. In *2008 12th International Conference Information Visualisation (2008)*, IEEE, pp. 215–220.
- [MFM03] MARCUS, A., FENG, L., AND MALETIC, J. I. 3d representations for software visualization. In *Proceedings of the 2003 ACM symposium on Software visualization (2003)*, ACM, pp. 27–ff.

- [MGT⁺03] MUNZNER, T., GUIMBRETIERE, F., TASIRAN, S., ZHANG, L., AND ZHOU, Y. Treejuxtaposer: Scalable tree comparison using focus+context with guaranteed visibility. *ACM Trans. Graph.* 22, 3 (July 2003), 453–462.
- [MK92] MELÍN, M., AND KUNST, M. Mincalc development kit 2.1. *Geologický ústav Akademie věd, Praha* (1992).
- [Mor13] MORELAND, K. A survey of visualization pipelines. *IEEE Transactions on Visualization and Computer Graphics* 19, 3 (March 2013), 367–378.
- [MRC91] MACKINLAY, J. D., ROBERTSON, G. G., AND CARD, S. K. The perspective wall: Detail and context smoothly integrated. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1991), ACM, pp. 173–176.
- [MRY⁺04] MASUMOTO, S., RAGHAVAN, V., YONEZAWA, G., NEMOTO, T., AND SHIONO, K. Construction and visualization of a three dimensional geologic model using GRASS GIS. *Trans. GIS* 8, 2 (2004), 211–223.
- [MSS12] MULLONI, A., SEICHTER, H., AND SCHMALSTIEG, D. Indoor navigation with mixed reality world-in-miniature views and sparse localization on mobile devices. In *Proceedings of the International Working Conference on Advanced Visual Interfaces* (2012), ACM, pp. 212–215.
- [Mun97] MUNZNER, T. H3: Laying out large directed graphs in 3d hyperbolic space. In *Information Visualization, 1997. Proceedings., IEEE Symposium on* (1997), IEEE, pp. 2–10.
- [Mun09] MUNZNER, T. A nested model for visualization design and validation. *IEEE transactions on visualization and computer graphics* 15, 6 (2009), 921–928.
- [Mun14] MUNZNER, T. *Visualization Analysis and Design*. A.K. Peters visualization series. A K Peters, 2014.
- [MW95] MARTIN, A. R., AND WARD, M. O. High dimensional brushing for interactive exploration of multivariate data. In *Proceedings of the 6th Conference on Visualization'95* (1995), IEEE Computer Society, p. 271.

- [New94] NEWELL, A. *Unified theories of cognition*. Harvard University Press, 1994.
- [Nor88] NORMAN, D. The design of everyday things (originally published: The psychology of everyday things). *The Design of Everyday Things (Originally published: The psychology of everyday things)* (1988).
- [Nor98] NORTH, C. A taxonomy of information visualization user-interfaces. <http://www.cs.umd.edu/~north/infviz.html> (1998).
- [Per85] PERLIN, K. An image synthesizer. *SIGGRAPH Comput. Graph.* 19, 3 (July 1985), 287–296.
- [PHK04] PIEPER, S., HALLE, M., AND KIKINIS, R. 3d slicer. 632–5.
- [PJ95] PARKER, S. G., AND JOHNSON, C. R. Scirun: A scientific programming environment for computational steering. In *Supercomputing, 1995. Proceedings of the IEEE/ACM SC95 Conference* (1995), pp. 52–52.
- [PMS⁺98] PLAISANT, C., MUSHLIN, R., SNYDER, A., LI, J., HELLER, D., AND SHNEIDERMAN, B. Lifelines: using visualization to enhance navigation and analysis of patient records. In *Proceedings of the AMIA Symposium* (1998), American Medical Informatics Association, p. 76.
- [RC94] RAO, R., AND CARD, S. K. The table lens: merging graphical and symbolic representations in an interactive focus+ context visualization for tabular information. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1994), ACM, pp. 318–322.
- [RFF⁺08] ROBERTSON, G., FERNANDEZ, R., FISHER, D., LEE, B., AND STASKO, J. Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (2008), 1325–1332.
- [Ric95] RICHARD, L. Minpet: Mineralogical and petrological data processing system, version 2.02. *MinPet Geological Software, Québec, Canada* (1995).
- [Rob07] ROBERTS, J. C. State of the art: Coordinated & multiple views in exploratory visualization. In *Coordinated and Multiple Views in Exploratory Vi-*

- sualization, 2007. CMV'07. Fifth International Conference on* (2007), IEEE, pp. 61–71.
- [Roe94] ROEDER, P. L. Chromite; from the fiery rain of chondrules to the kilauea iki lava lake. *The Canadian Mineralogist* 32, 4 (1994), 729–746.
- [RSR04] ROSSET, A., SPADOLA, L., AND RATIB, O. Osirix: An open-source software for navigating in multidimensional dicom images. *J. Digital Imaging* 17, 3 (2004), 205–216.
- [RV11] RAMACHANDRAN, P., AND VAROQUAUX, G. Mayavi: 3d visualization of scientific data. *Computing in Science Engineering* 13, 2 (March 2011), 40–51.
- [SA82] SPENCE, R., AND APPERLEY, M. Data base navigation: an office environment for the professional. *Behaviour & Information Technology* 1, 1 (1982), 43–54.
- [SCB92] SWAYNE, D. F., COOK, D., AND BUJA, A. User's manual for xgobi: A dynamic graphics program for data analysis. *Bellcore, Morristown, NJ* (1992).
- [SH16] SCHMALSTIEG, D., AND HOLLERER, T. *Augmented reality: principles and practice*. Addison-Wesley Professional, 2016.
- [Shn94] SHNEIDERMAN, B. Dynamic queries for visual information seeking. *IEEE software* 11, 6 (1994), 70–77.
- [Shn96] SHNEIDERMAN, B. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on* (1996), IEEE, pp. 336–343.
- [SLCZ09] SPENCER, B., LARAMEE, R. S., CHEN, G., AND ZHANG, E. Evenly spaced streamlines for surfaces: An image-based approach. *Computer Graphics Forum* (2009).
- [SLM04] SCHROEDER, W. J., LORENSEN, B., AND MARTIN, K. *The visualization toolkit*. Kitware, 2004.

- [SLMV92] SCHROEDER, W. J., LORENSEN, W. E., MONTANARO, G. D., AND VOLPE, C. R. Visage: an object-oriented scientific visualization system. In *Visualization, 1992. Visualization '92, Proceedings., IEEE Conference on* (Oct 1992), pp. 219–226.
- [SML96] SCHROEDER, W. J., MARTIN, K. M., AND LORENSEN, W. E. The design and implementation of an object-oriented toolkit for 3d graphics and visualization. In *Proceedings of the 7th conference on Visualization'96* (1996), IEEE Computer Society Press, pp. 93–ff.
- [SML⁺06] SCHROEDER, W. J., MARTIN, K., LORENSEN, B., SOBIERAJSKI AVILA, L. C., AVILA, R. C., AND LAW, C. C. C. *The visualization toolkit : an object-oriented approach to 3D graphics*. Kitware, New York, 2006. CD rom Toolkit : version 5.0.2.
- [SMW94] STARK, M., MÜLLER, H., AND WELSCH, U. Variations of the splitting box scheme for adaptive generation of contour surfaces in volume data. In *IN SCIENTIFIC VISUALIZATION: OVERVIEWS, METHODOLOGIES AND TECHNIQUES, IEEE COMPUTER* (1994), Society Press, pp. 337–356.
- [SNHS13] SCHULZ, H.-J., NOCKE, T., HEITZLER, M., AND SCHUMANN, H. A design space of visualization tasks. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2366–2375.
- [Spe07] SPENCE, R. *Information Visualization: Design for Interaction*. 2007.
- [SSTR93] SARKAR, M., SNIBBE, S. S., TVERSKY, O. J., AND REISS, S. P. Stretching the rubber sheet: a metaphor for viewing large layouts on small screens. In *Proceedings of the 6th annual ACM symposium on User interface software and technology* (1993), ACM, pp. 81–91.
- [Ste46] STEVENS, S. S. On the theory of scales of measurement, 1946.
- [STH02] STOLTE, C., TANG, D., AND HANRAHAN, P. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics* 8, 1 (2002), 52–65.

- [TC05] THOMAS, J. J., AND COOK, K. A. *Illuminating the path: the research and development agenda for visual analytics*. IEEE Computer Society, 2005.
- [Tel14] TELEA, A. C. *Data Visualization: Principles and Practice, Second Edition*, 2nd ed. A. K. Peters, Ltd., Natick, MA, USA, 2014.
- [TFJ12] TOMINSKI, C., FORSELL, C., AND JOHANSSON, J. Interaction support for visual comparison inspired by natural behavior. *IEEE transactions on visualization and computer graphics* 18, 12 (2012), 2719–2728.
- [TJ00] TANRIVERDI, V., AND JACOB, R. J. Interacting with eye movements in virtual environments. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (2000), ACM, pp. 265–272.
- [TS08] TU, Y., AND SHEN, H.-W. Balloon focus: a seamless multi-focus+ context method for treemaps. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (2008), 1157–1164.
- [Twe97] TWEEDIE, L. Characterizing interactive externalizations. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems* (1997), ACM, pp. 375–382.
- [UCM13a] URRIBARRI, D., CASTRO, S. M., AND MARTIG, S. Gyrolayout: A hyperbolic level-of-detail tree layout. *J. UCS* 19, 1 (2013), 132–156.
- [UCM13b] URRIBARRI, D. K., CASTRO, S. M., AND MARTIG, S. R. Gyrolayout: A hyperbolic level-of-detail tree layout. *Journal of Universal Computer Science* 19, 1 (2013), 132–156.
- [UFK⁺89] UPSON, C., FAULHABER, T. A., KAMINS, D., LAIDLAW, D., SCHLEGEL, D., VROOM, J., GURWITZ, R., AND VAN DAM, A. The application visualization system: a computational environment for scientific visualization. *IEEE Computer Graphics and Applications* 9, 4 (July 1989), 30–42.
- [vis05] Visit user’s manual. Tech. Rep. UCRL-SM-220449, Lawrence Livermore Nat’l Laboratory, 2005.

- [War04] WARE, C. *Information Visualization: Perception for Design*. Morgan Kaufmann, 2004.
- [WKG10] WARD, M. O., GRINSTEIN, G., AND KEIM, D. *Interactive data visualization: foundations, techniques, and applications*. CRC Press, 2010.
- [Wil90] WILLIAMS, K. L.; ROCK, N. M. S. C. G. W. Spinel and spineltab: Machintosh programs to plot spinel analyses in the three-dimensional oxidized (magnetite) and reduced (ulvospinel) prism. *American Mineralogist* 74 (1990), 1428–1430.
- [Wil06] WILKINSON, L. *The grammar of graphics*. Springer Science & Business Media, 2006.
- [WLJ⁺12] WALNY, J., LEE, B., JOHNS, P., RICHE, N. H., AND CARPENDALE, S. Understanding pen and touch interaction for data exploration on interactive whiteboards. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2779–2788.
- [WR90] WILLIAMS, C. S., AND RASURE, J. R. A visual language for image processing. In *Visual Languages, 1990., Proceedings of the 1990 IEEE Workshop on* (Oct 1990), pp. 86–91.
- [WWL⁺10] WU, Y., WEI, F., LIU, S., AU, N., CUI, W., ZHOU, H., AND QU, H. Opinionseer: interactive visualization of hotel customer feedback. *IEEE transactions on visualization and computer graphics* 16, 6 (2010), 1109–1118.
- [WY04] WARD, M. O., AND YANG, J. Interaction spaces in data and information visualization. In *VisSym* (2004), pp. 137–145.
- [YaKSJ07] YI, J. S., AH KANG, Y., STASKO, J., AND JACKO, J. Toward a deeper understanding of the role of interaction in information visualization. *IEEE transactions on visualization and computer graphics* 13, 6 (2007), 1224–1231.
- [YWR02] YANG, J., WARD, M. O., AND RUNDENSTEINER, E. A. Interring: An interactive tool for visually navigating and manipulating hierarchical structures. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium On* (2002), IEEE, pp. 77–84.

- [ZF98] ZHOU, M. X., AND FEINER, S. K. Visual task characterization for automated visual discourse synthesis. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1998), ACM Press/Addison-Wesley Publishing Co., pp. 392–399.