



Tema 4:

Técnicas de Distorsión

3.1 Introducción

3.2 Distorsión basada en funciones

3.2.1 Polyfocal Display

3.2.2 Bifocal Display

3.2.3 Fisheye View (Furnas)

3.2.4 Fisheye View (Sarkar and Brown)

3.2.5 Perspective Wall

3.2.6 Table lens

3.2.7 Tipo de función

3.2.8 Implementación

3.3 Otras técnicas

3.3.1 Hyperbolic Trees

3.3.2 Bundles

3.3.3 Document Lens

3.3.4 Hyperbox

3.4 Conclusiones

Máster Universitario en Ingeniería Informática

Pablo.Bermejo@uclm.es

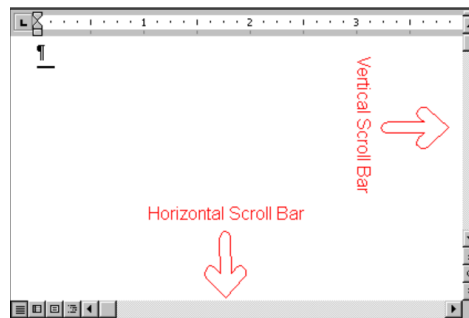
3.1 Introducción

- En ocasiones la información a mostrar requiere más espacio del disponible en pantalla.
- Esto ocasiona 3 problemas:
 - Es difícil localizar una pieza de información
 - *Subsección dentro de un texto*
 - Es difícil relacionarla con el contexto
 - *¿Qué más subsecciones comparten ese capítulo?*
 - Y por lo tanto es difícil interpretar dicha información
 - *¿A qué se refiere este texto?*
- Soluciones comunes que no distorsionan la imagen:
 - Paginación



3.1 Introducción

- Scrolling



- Acceso jerárquico

ÍNDICE:

1. La literatura y el cine. Interrelaciones y divergencias ..	3
1.1. La literatura en el cine	4
1.2. El cine en la literatura	5
1.3. El lenguaje literario y el lenguaje fílmico	6
2. Las adaptaciones cinematográficas	8
2.1. Adaptación de las novelas al cine	9
2.1.1. Adaptación como ilustración	9
2.1.2. Adaptación como transposición	10
2.1.3. Adaptación como interpretación	11
2.1.4. Adaptación libre	12
2.2. Adaptación de obras teatrales al cine	13

- Aunque estas soluciones de *no distorsión* alivian los problemas citados, todavía resulta costoso poder navegar y relacionar grandes cantidades de información:
 - Índice: hay que volver al índice para ver cómo se relaciona un ítem de información con otro
 - Paginación: es necesario explorar todas las páginas para hacerse una idea del contexto



3.1 Introducción

3.2 Distorsión basada en funciones

3.2.1 Polyfocal Display

3.2.2 Bifocal Display

3.2.3 Fisheye View (Furnas)

3.2.4 Fisheye View (Sarkar and Brown)

3.2.5 Perspective Wall

3.2.6 Table lens

3.2.7 Tipo de función

3.2.8 Implementación

3.3 Otras técnicas

3.3.1 Hyperbolic Trees

3.3.2 Bundles

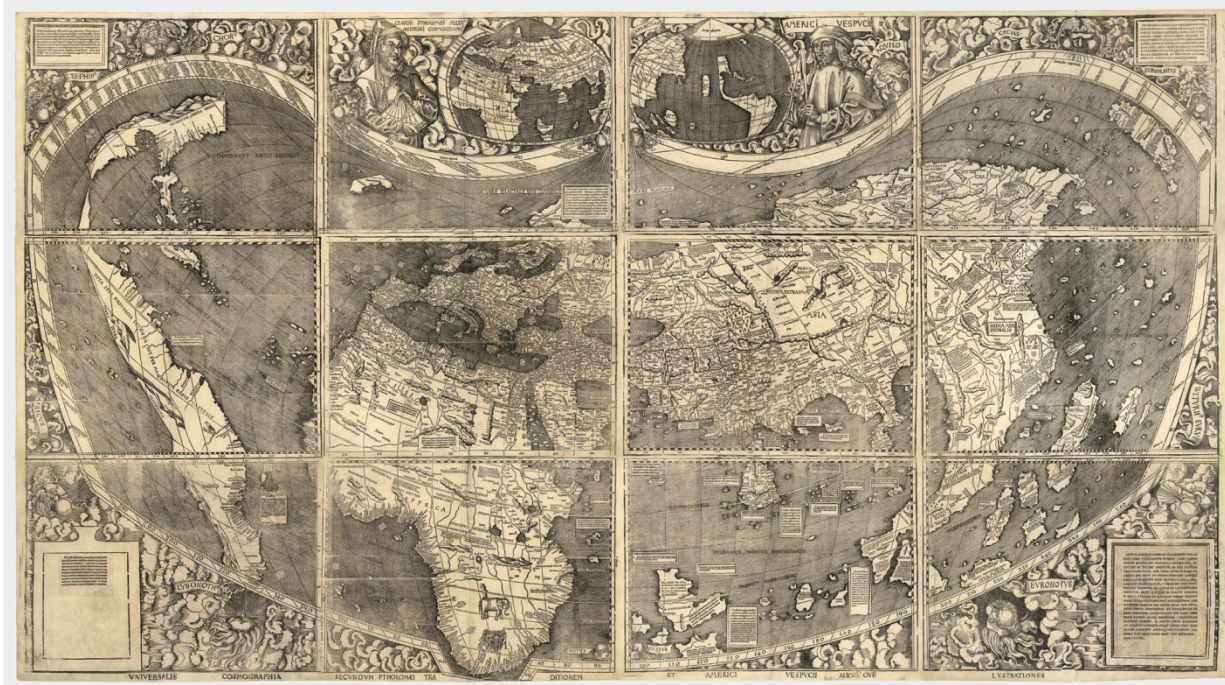
3.3.3 Document Lens

3.3.4 Hyperbox

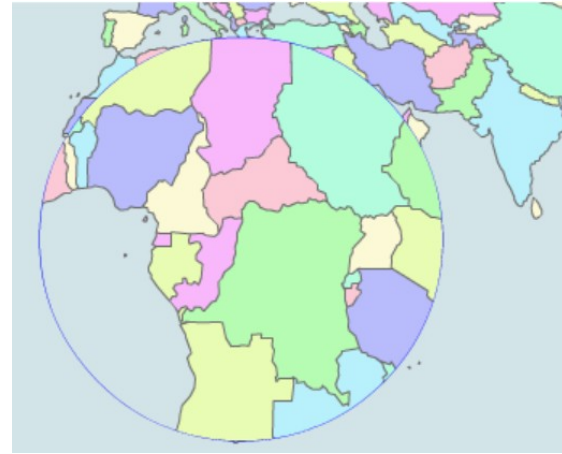
3.4 Conclusiones

3.2 Distorsión basada en funciones

- Las **técnicas de distorsión** tienen como objetivo ayudar al observador a encontrar la información necesaria sin llegar a perder una imagen global de toda la información
- La distorsión de la información existe hace varios siglos por cartógrafos:



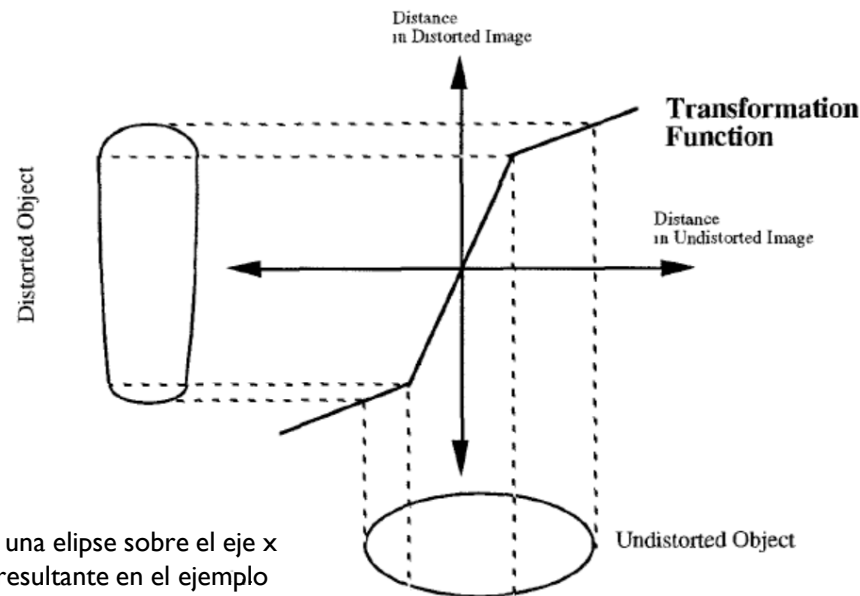
3.2 Distorsión basada en funciones



- Hace relativamente poco tiempo se han comenzado a aplicar *técnicas de distorsión* a la información para su representación en pantalla, que en general se basan en:
 - Seleccionar un foco de la imagen y magnificarla
 - Desmagnificar el resto

3.2 Distorsión basada en funciones

- Una técnica de distorsión es definida formalmente por 2 funciones matemáticas: [Leung and Apperley 1994]
 - **Función de Transformación $T(x)$:**
 - Define cómo se mapea la imagen original a la imagen destino.
 - Se define una función de transformación para cada dimensión



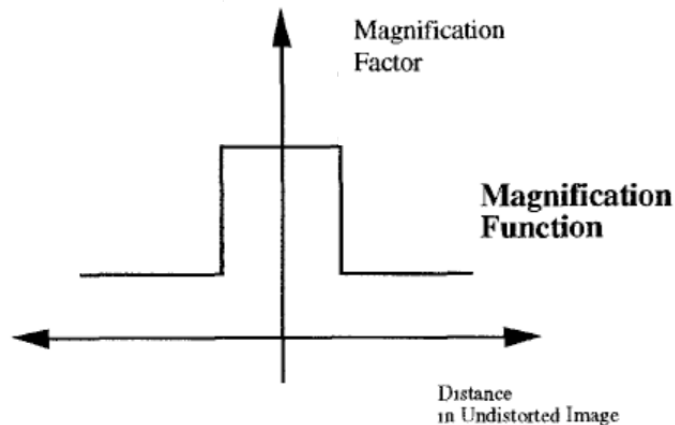
Ejemplo de distorsión de una elipse sobre el eje x (el objeto distorsionado resultante en el ejemplo hay que rotarlo 90 grados!)

3.2 Distorsión basada en funciones

- **Función de Magnificación $M(x)$**

- Es la derivada de primer orden de la función de transformación
 - $M(x) = T'(x)$
- Representa el grado de magnificación o desmagnificación de todo el área de la imagen

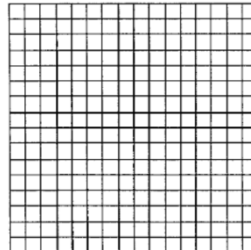
Función de magnificación para el ejemplo anterior:
Los puntos centrales de la elipse se magnifican



- Si toma valores $= 0$, el punto correspondiente de la imagen desaparece
- Si toma valores < 0 , varios puntos cercanos se superponen entre sí.

3.2 Distorsión basada en funciones

- Es de esperar que un sistema de distorsión permita al usuario interacción para **poder cambiar el foco** de la distorsión, lo cual provoca:
 - Nuevo cálculo de la función de transformación para cada punto
 - Actualizar la representación
- El tiempo de respuesta del sistema dependerá de:
 - La complejidad matemática de la función de transformación
 - El tamaño de la imagen
 - Recursos del sistema
- En este tema veremos distintas funciones de transformación que dan lugar a distintas Técnicas de Distorsión, aplicada a la siguiente imagen de origen:

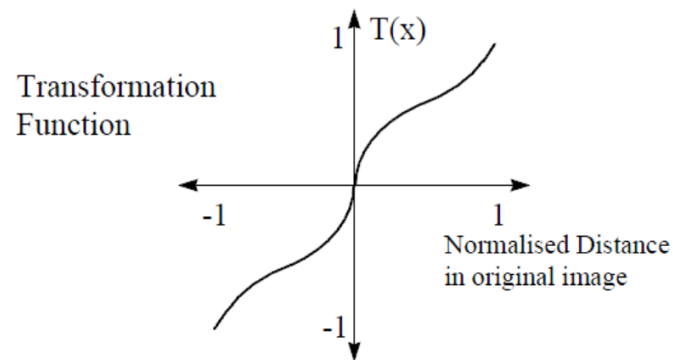


3.3.1 Polyfocal Display

- Primera técnica definida [Kadmon and Shlomi, 1978]
- Magnifica el foco a cambio de desmagnificar el resto.
- Su **función de Transformación** es:

$$T_{polyfocal}(x) = x + \frac{A \cdot x}{(1 + C \cdot x^2)}$$

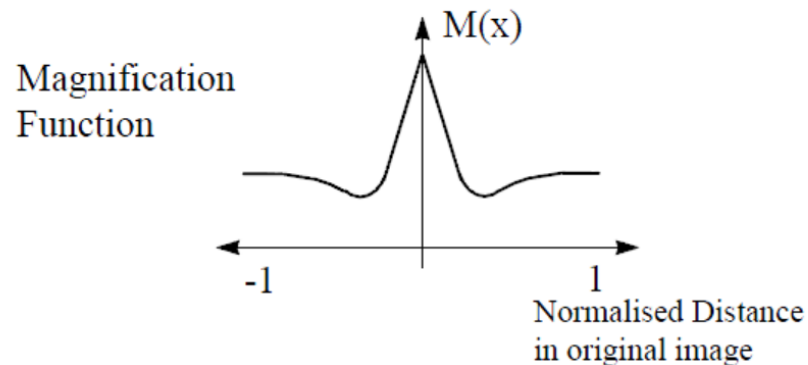
- A y C son constantes.



3.2.1 Polyfocal Display

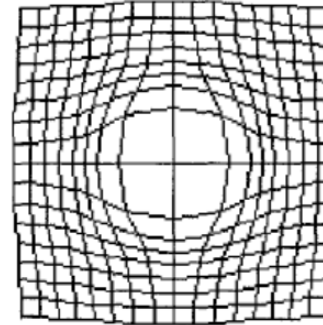
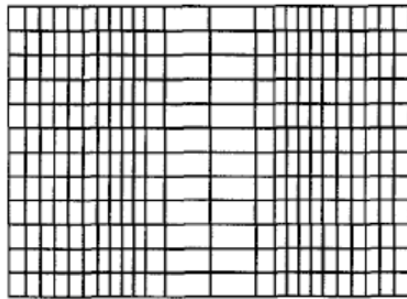
- La función de **Magnificación** $M(x)$ se obtiene derivando $T(x)$

$$\begin{aligned}M_{polyfocal}(x) &= \frac{d}{dx}(T_{polyfocal}(x)) \\&= 1 + \frac{A.(1 + C.x^2) - A.x.2.C.x}{(1 + C.x^2)^2} \\&= 1 + \frac{A - A.C.x^2}{(1 + C.x^2)^2} \\M_{polyfocal}(x) &= 1 + \frac{A.(1 - C.x^2)}{(1 + C.x^2)^2}\end{aligned}$$



3.2.1 Polyfocal Display

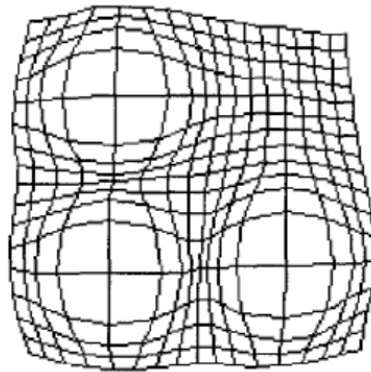
- Resultado al distorsionar en 1 (izquierda) y 2 (derecha) dimensiones.



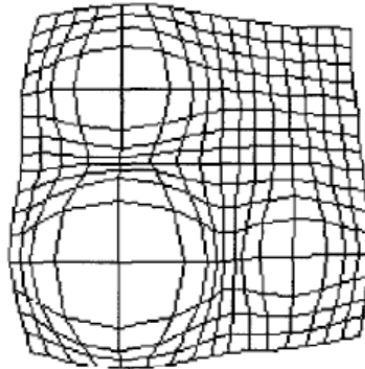
- Esta técnica presenta el problema de que el resultado **distorsiona los bordes del continente**.

3.2.1 Polyfocal Display

- Una variante, **Multifocal Polyfocal Display**, selecciona distintos puntos como focos.
 - Puede utilizarse la misma función de transformación en los focos

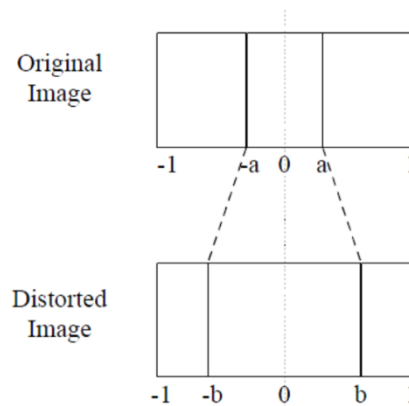


- O definir una función para cada foco (o crear una función que contemple esos 3 focos)



3.2.2 Bifocal Display

- Dado un punto a en la imagen origen y su punto b correspondiente en la imagen transformada:[Spence and Apperly 1982].

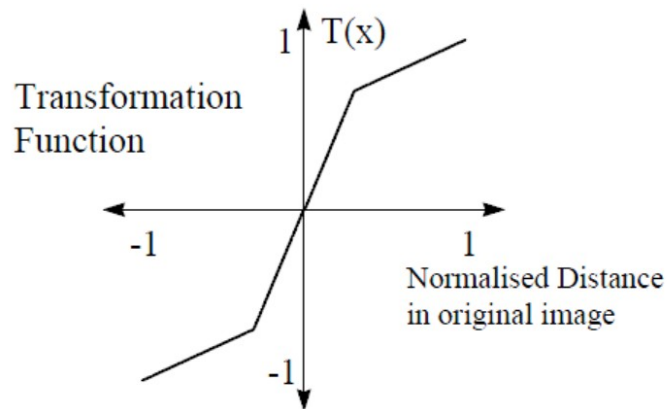


- La **función de Transformación** es:

$$|x| \leq a, \quad T(x) = x \cdot \frac{b}{a} \quad (\text{Focal region})$$

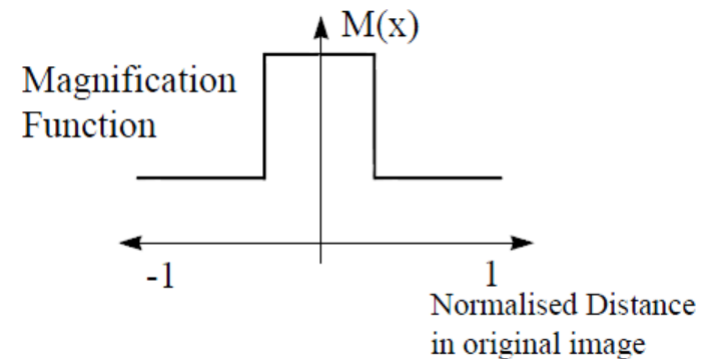
$$\left. \begin{aligned} x > a, \quad T(x) &= b + (x - a) \cdot \frac{(1 - b)}{(1 - a)} \\ x < -a, \quad T(x) &= -T(|x|) \end{aligned} \right\} (\text{Context Region})$$

3.2.2 Bifocal Display



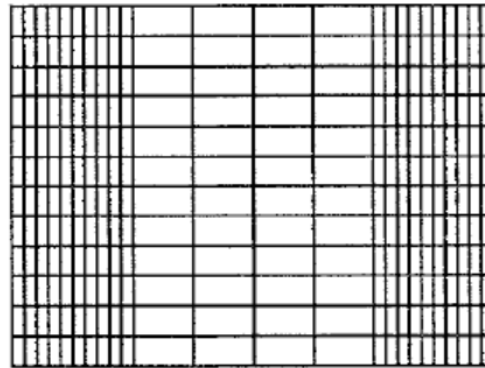
- Y por lo tanto la **función de Magnificación** será la derivada correspondiente:

$$\begin{aligned} |x| \leq a, \quad M(x) &= \frac{b}{a} && \text{(Focal region)} \\ |x| > a, \quad M(x) &= \frac{(1-b)}{(1-a)} && \text{(Context region)} \end{aligned}$$



3.2.2 Bifocal Display

- Resultados de aplicar distorsión bifocal :

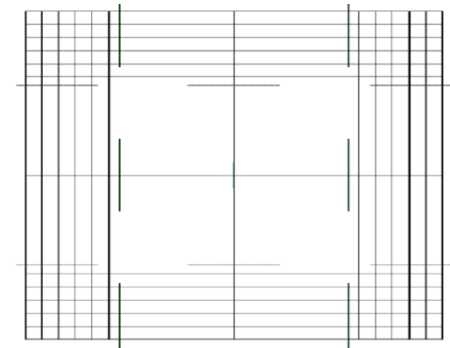


- Nótese que esta técnica **no distorsiona los bordes** del continente.
- Aplicado a 1 dimensión, la porción de imagen **fuera del foco se desmagnifica de igual manera** en todos los puntos (al contrario que en poliyocal)
- La función de magnificación contiene **cambios bruscos** entre grados de magnificación
- Puede verse como un caso concreto de Polyfocal Display

3.2.2 Bifocal Display

- En [Leung, 1989], se presenta una técnica bifocal para 2 dimensiones, que divide el área en 9 secciones:

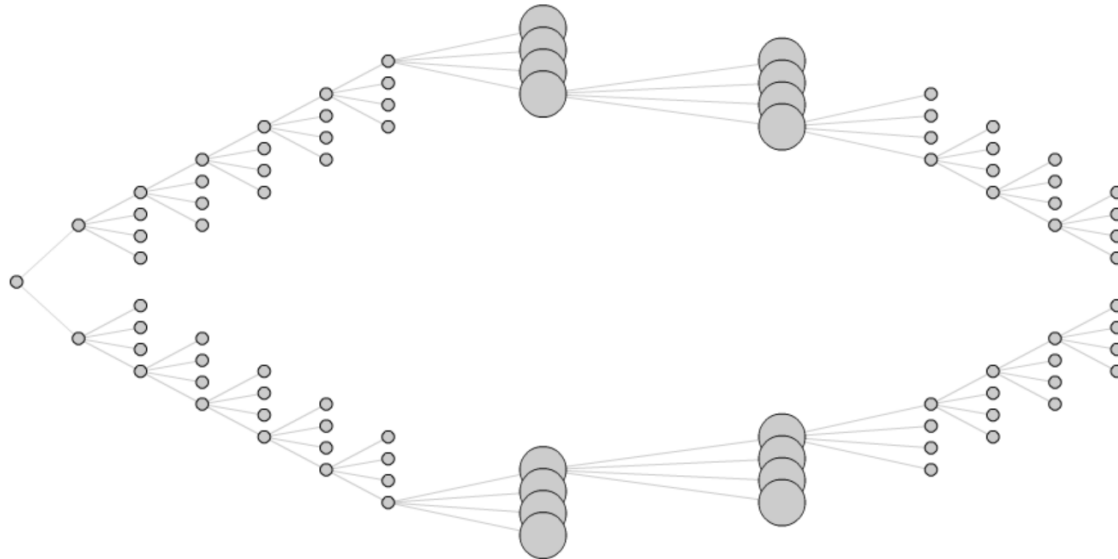
Demagnification in both X and Y dimensions	Demagnification in Y dimension	Demagnification in both X and Y dimensions
Demagnification in X dimension	Central 'Focus' Region no demagnification	Demagnification in X dimension
Demagnification in both X and Y dimensions	Demagnification in Y dimension	Demagnification in both X and Y dimensions



- Las esquinas se desmagnifican en las 2 dimensiones, así que el resultado es que no se distorsionan sino que solo se reduce su tamaño.

3.2.2 Bifocal Display

- Otros ejemplos:
 - Visualizar grafos: <http://flare.prefuse.org/demo>

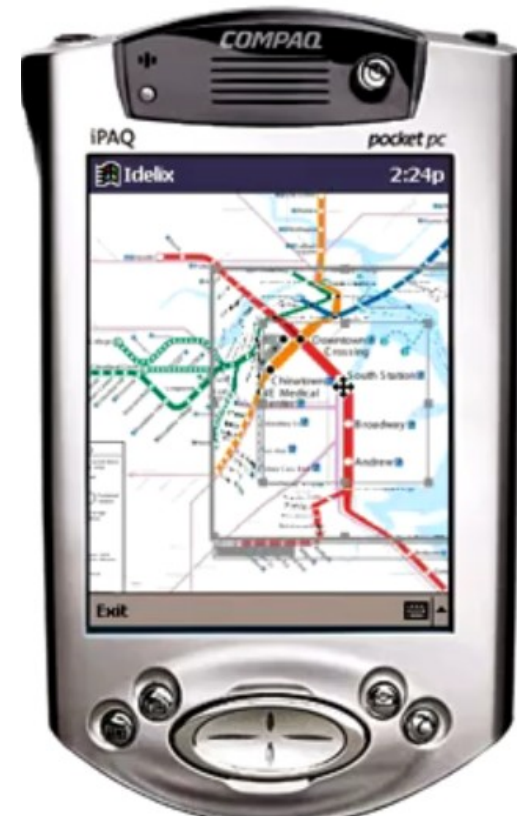


- Exploración de Iconos: (si tienen tamaño decreciente es Fishey View)



3.2.2 Bifocal Display

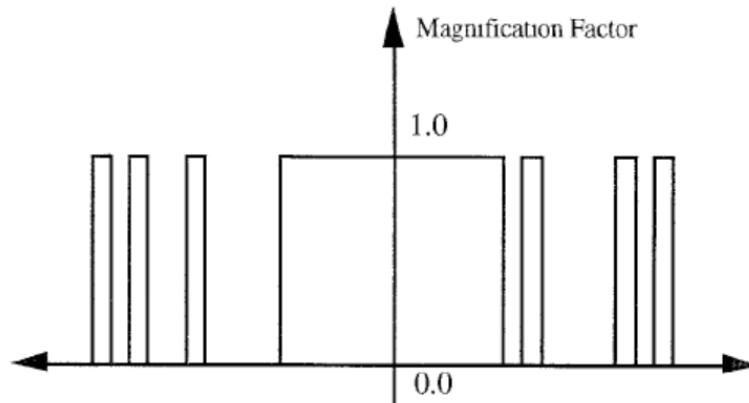
- Explorar mapas:
 - <http://www.youtube.com/watch?v=nTglJdK3kcY>
 - <http://www.youtube.com/watch?v=FDYo9uvrNJ0>



3.2.3 Fisheye View (Furnas)

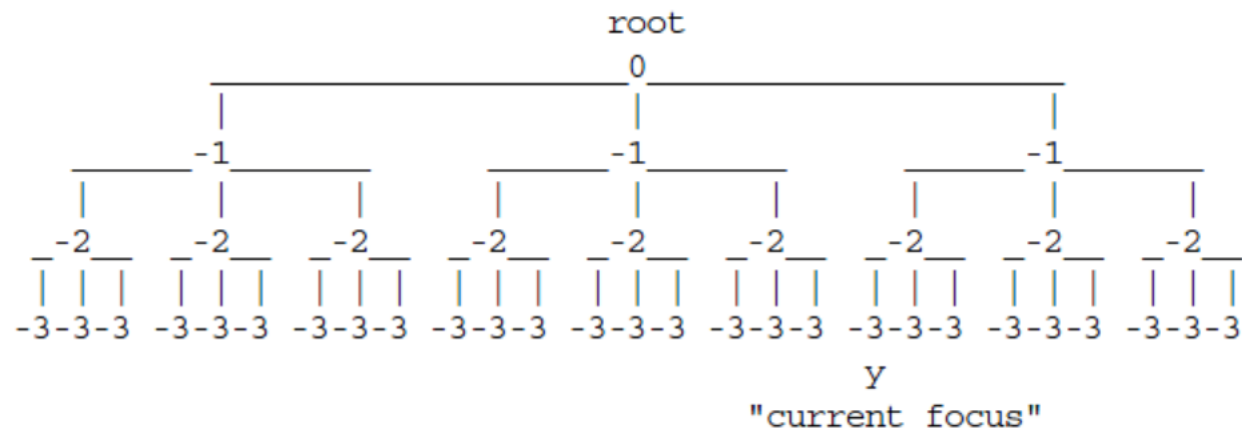
- La primera técnica de Fisheye fue propuesta en [Furnas, 1986] para la visualización de información jerárquica.
- **API(a)**= *a priori importance* de cada elemento en la jerarquía dada la tarea actual.
- **D(a,b)**= *distancia* del punto *a* al elemento *b*.
- **DOI(a|b)**= **API(a) - D(a,b)** *degree of interest* del usuario en el punto *a* dado que el foco está en el punto *b*.
- **Threshold > DOI** : se magnifica la información; si no, se elimina.
- La función **DOI** *suprime* información, pero no magnifica ni desmagnifica.

Furnas' Fisheye View

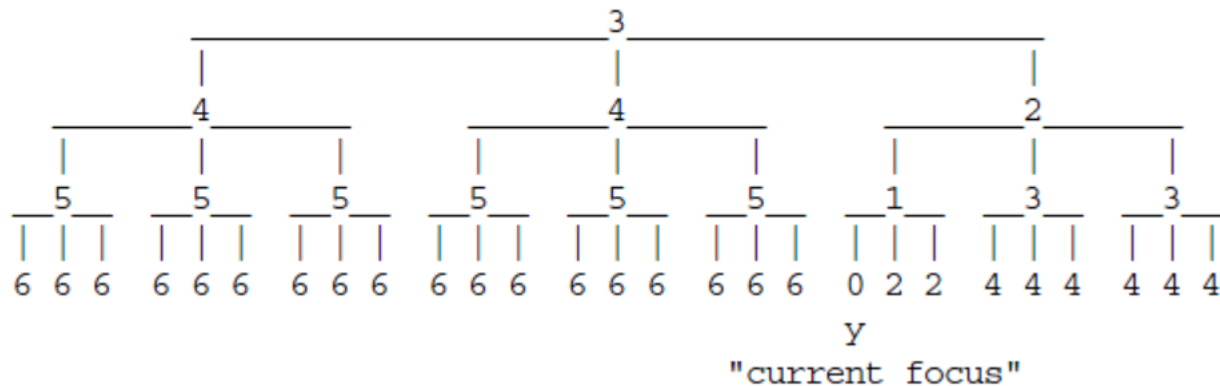


3.2.3 Fisheye View (Furnas)

- *Ejemplo aplicado a árboles jerárquicos*
 - **API(a)**=distancia(root,a)

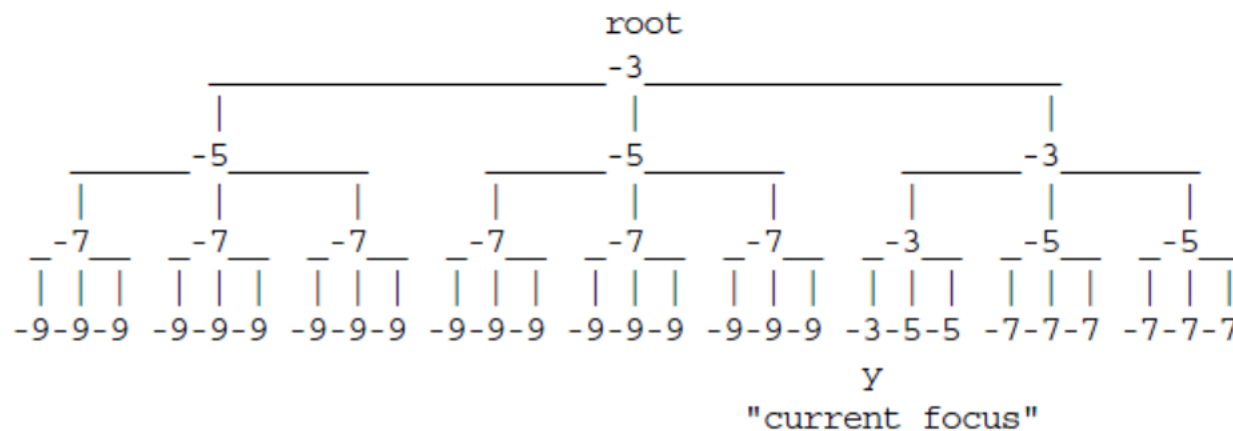


- **D(a,y)**=distancia entre el foco y el resto de puntos



3.2.3 Fisheye View (Furnas)

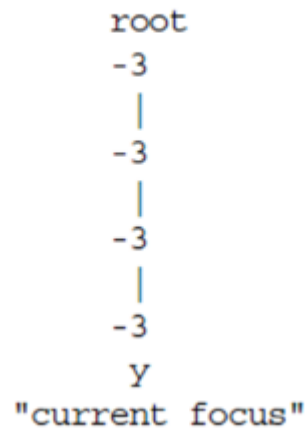
- $DOI(a|.y)=API(a) - D(a,y)$



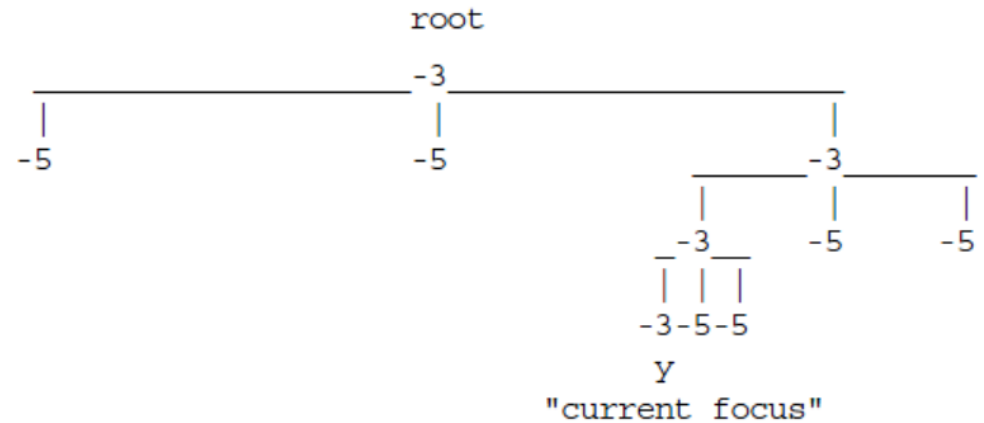
- Cuanto mayor es el valor, mayor grado de interés.
- **Threshold=-3: vista Fisheye de orden cero**
- **Threshold=-5: vista Fisheye de primer orden**
- **Threshold=-7: vista Fisheye de segundo orden**

3.2.3 Fisheye View (Furnas)

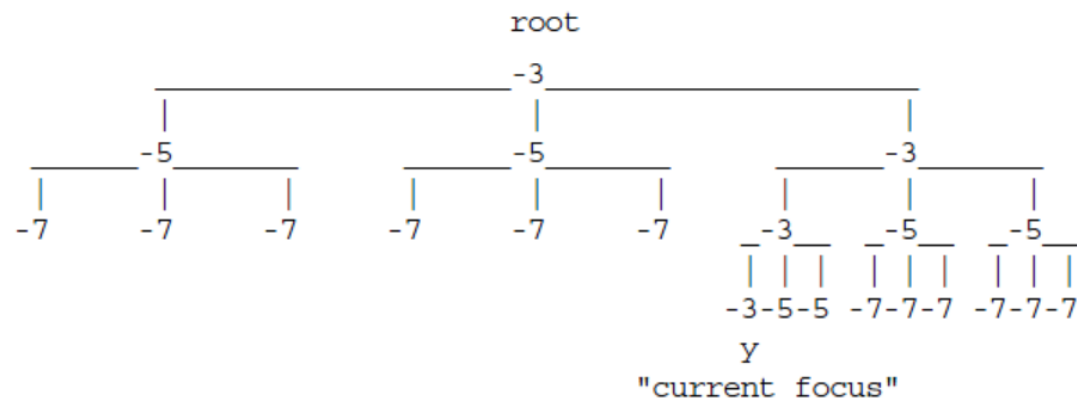
Zero-order tree fisheye:



First-order tree fisheye:



Second-order tree fisheye:



3.2.4 Fisheye View (Sarkar and Brown)

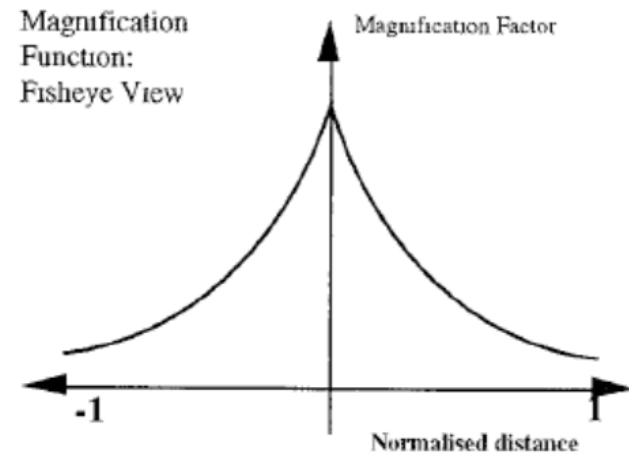
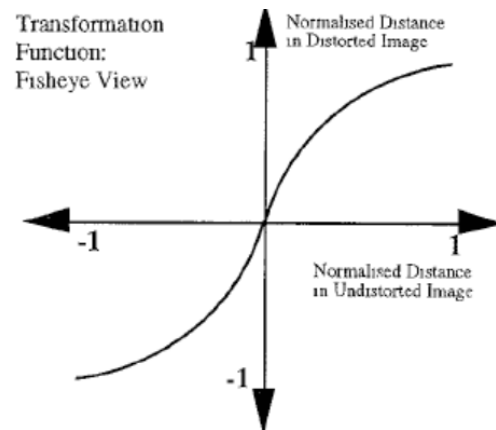
- En [Sarkar and Brown, 1992] (era post-ascii), se extiende la técnica de Fisheye de forma que no se suprima información
- Pretende solucionar el problema de la visión de nodos en grafos masivos
 - Existen muchas técnicas para decidir una distribución correcta en pantalla
 - Cuando la pantalla se llena, ¿cómo exploramos la información?
- Dos pasos:
 - Los nodos y ejes más importantes se magnifican, y el resto de desmagnifican.
 - Se vuelve a calcular la distribución del grafo para que ocupe la misma porción de pantalla.

3.2.4 Fisheye View (Sarkar and Brown)

- Las funciones de **Transformación** y **Magnificación** son:

$$T(x) = \frac{(d + 1)x}{(dx + 1)}$$

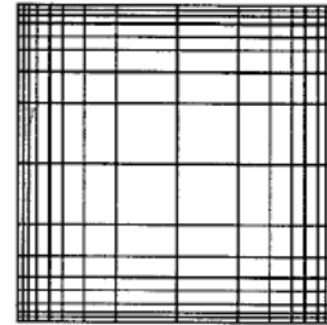
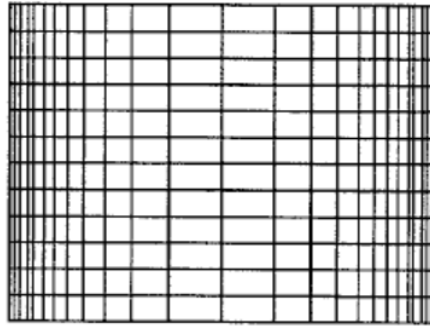
$$M(x) = \frac{(d + 1)}{(dx + 1)^2}$$



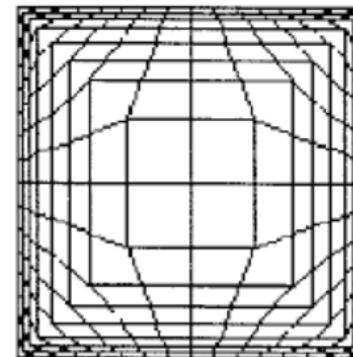
- Donde:
 - d es el factor de distorsión. Cuanto mayor es, mayor magnificación.
 - x es la distancia normalizada desde el punto x en consideración y el foco.

3.2.4 Fisheye View (Sarkar and Brown)

- La distorsión en 1 y 2 dimensiones quedaría así:

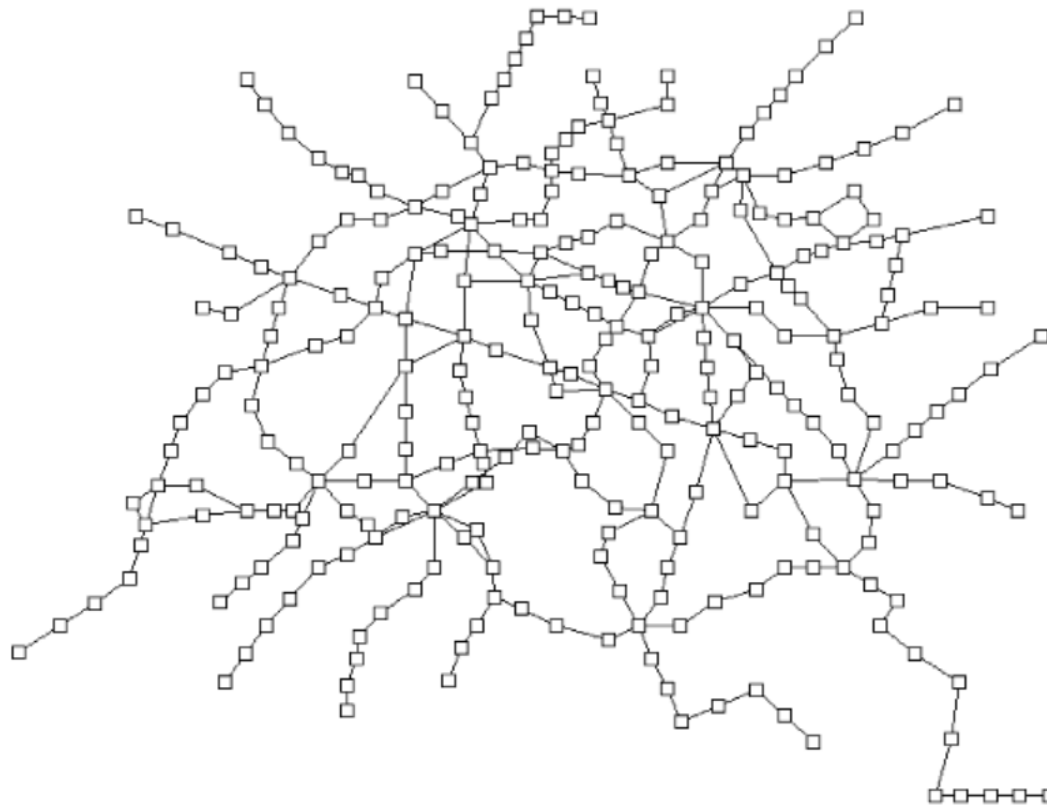


- Además de coordenadas Cartesianas, proponen aplicarlo la transformación a **coordenadas Polares**. Esto distorsiona el continente, así que añaden un segundo paso que es enmarcar la transformación en la forma del continente original.



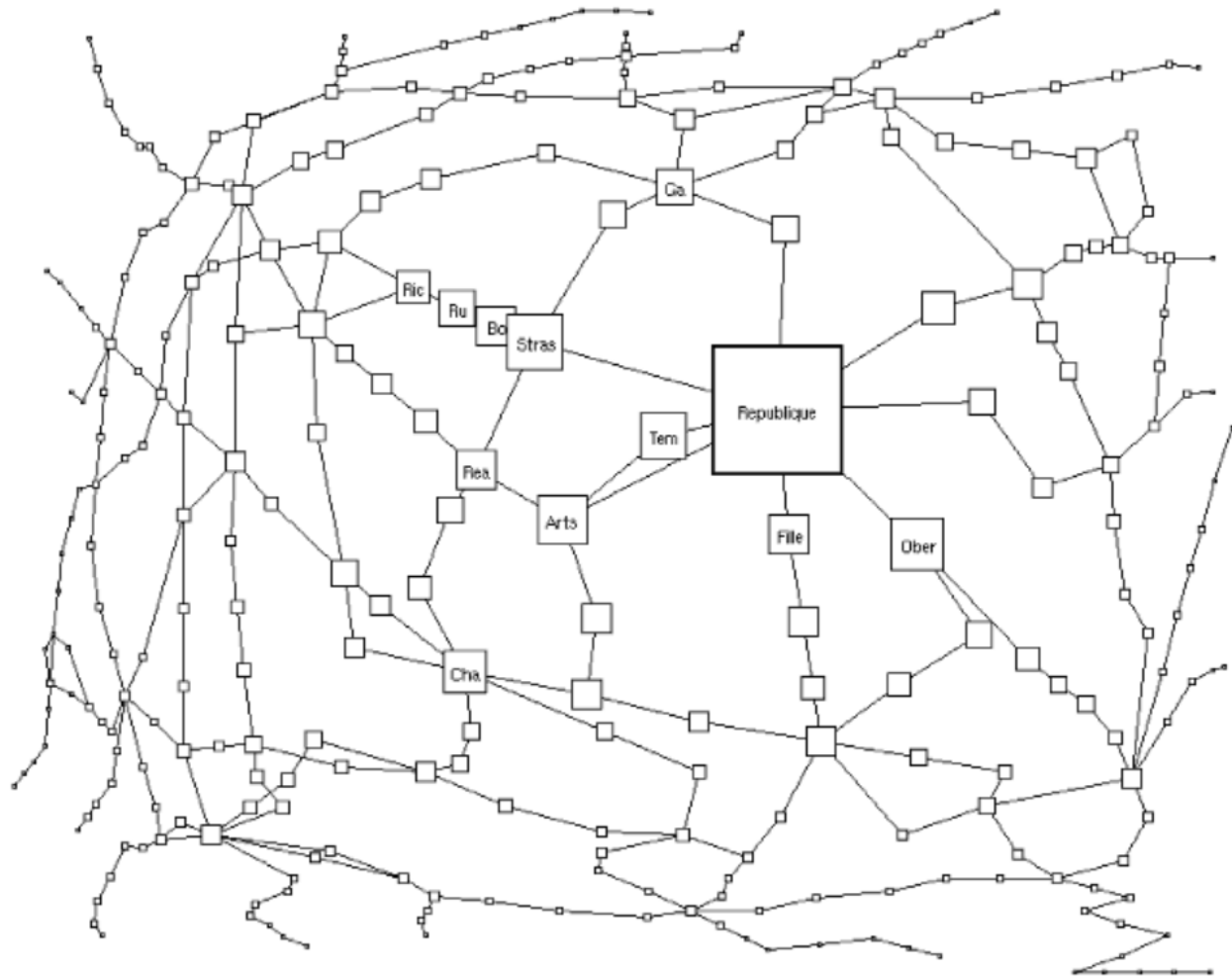
3.2.4 Fisheye View (Sarkar and Brown)

- Ejemplo Metro de París sin distorsionar:



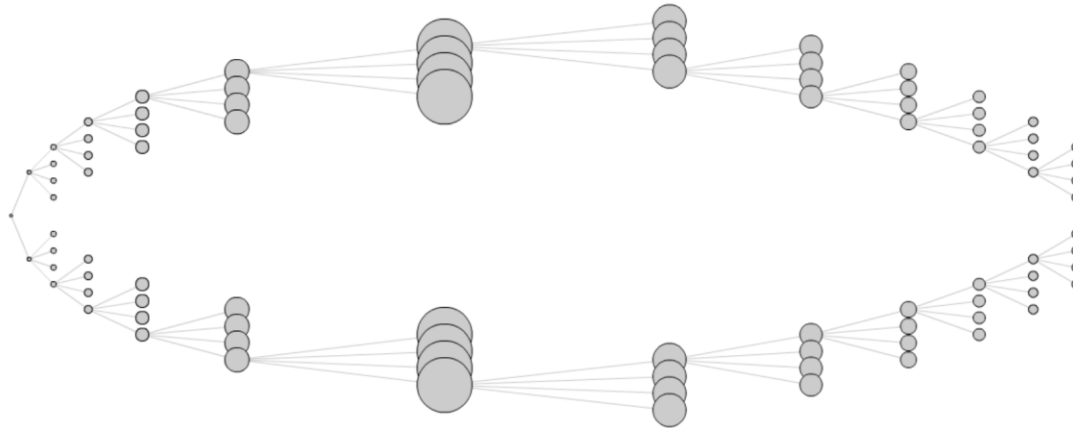
3.2.4 Fisheye View (Sarkar and Brown)

- Ejemplo metro de París distorsionado enfocando en la estación de interés

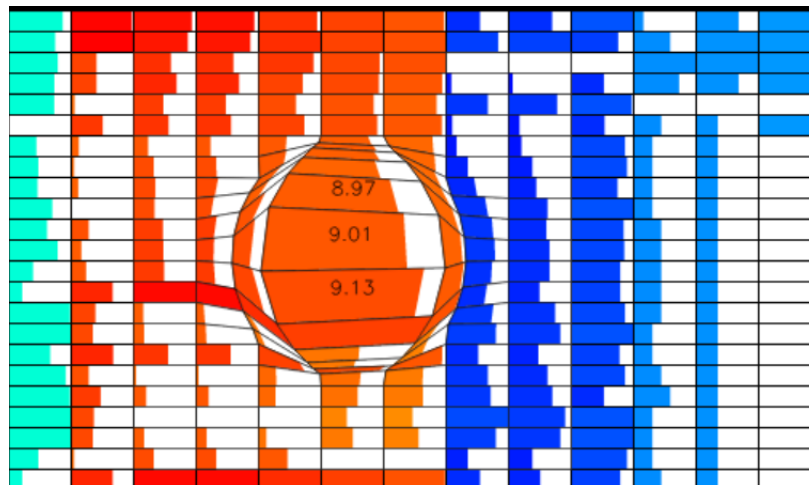


3.2.4 Fisheye View (Sarkar and Brown)

- <http://flare.prefuse.org/demo>



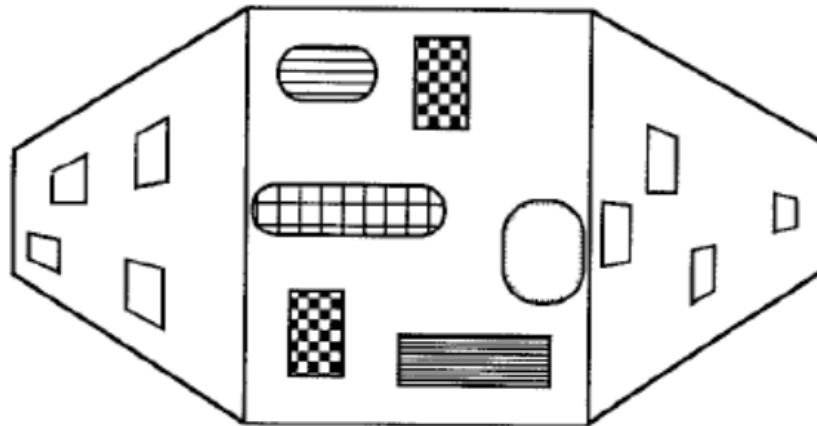
- http://magic-table.googlecode.com/svn/trunk/magic-table/google_visualisation/example_1.html



También podría verse como bifocal
porque no cambia el resto de la tabla

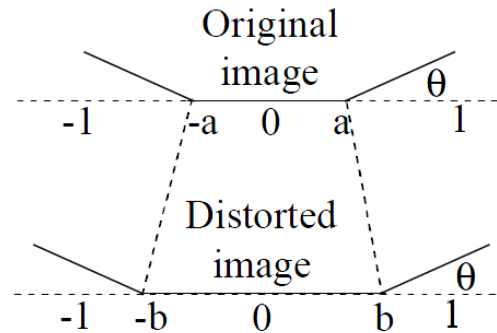
3.2.5 Perspective Wall

- Esta técnica de distorsión consiste [Mackinlay et al, 1991] en desmagnificar las regiones a ambos lados del foco de interés:
 - **El foco no se magnifica.**
 - El resto de **desmagnifica de forma proporcional** a la distancia, resultando así en una desmagnificación gradual.
 - El resultado ofrece una sensación 3D.
 - Útil para visualizar datos lineales

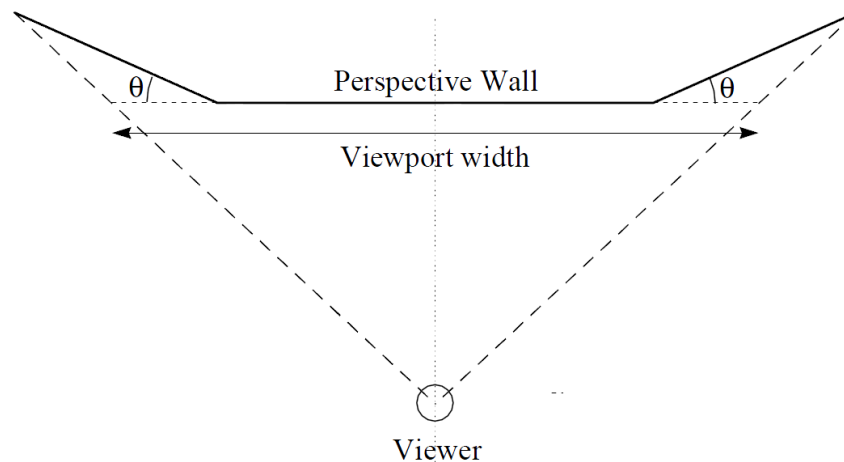


3.2.5 Perspective Wall

- Teniendo un punto de corte el panel frontal a y su transformado b , y el ángulo de transformación θ de los laterales,



- Y siendo V_y la distancia entre el visor (viewer) y el panel frontal



3.2.5 Perspective Wall

- La función de transformación es:

$$\text{for } x \leq a. \quad T_{\text{perspective}}(x) = x \cdot \frac{b}{a},$$

$$\text{for } x > a, \quad T_{\text{perspective}}(x) = \frac{[b + (x - a) \cdot \cos \theta]}{1 - \left[\frac{(1 - b)}{(1 - a)} \cos \theta \right] \cdot (x - a)}$$

- Pero como queremos que en foco no haya distorsión, hacemos $a=b$ y así la función de **transformación** queda:

$$\text{for } x \leq a, \quad T_{\text{perspective}}(x) = x,$$

$$\text{for } x > a, \quad T_{\text{perspective}}(x) = \frac{[a + (x - a) \cdot \cos \theta]}{1 - [1 - \cos \theta] \cdot (x - a)}$$

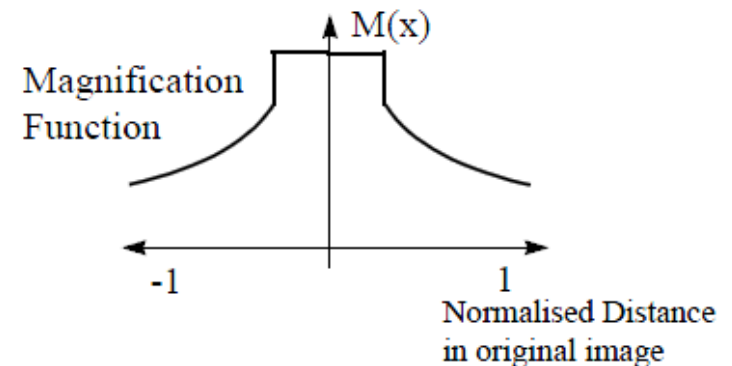
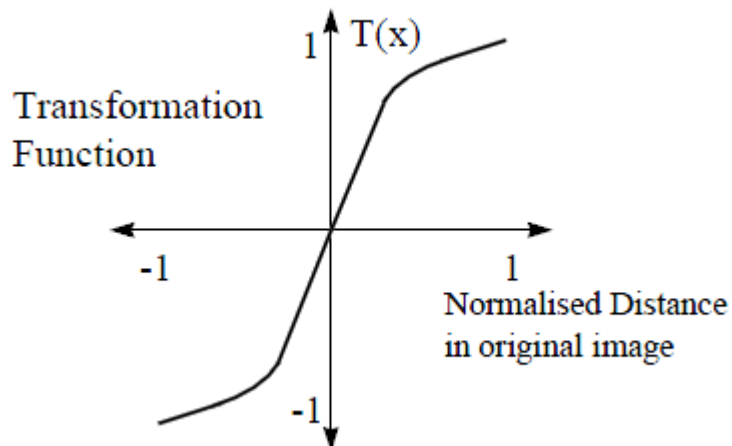
3.2.5 Perspective Wall

- Y siguiendo con la suposición de $a=b$, tenemos que la función de **magnificación** es:

$$\text{for } x \leq a, \quad M_{\text{perspective}}(x) = 1$$

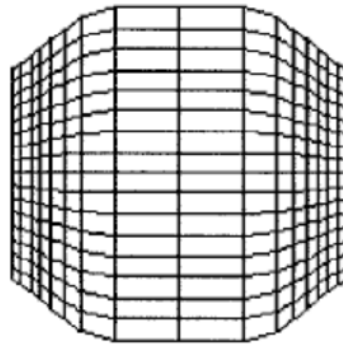
$$\text{for } x > a, \quad M_{\text{perspective}}(x) = \frac{a + (1 - a) \cdot \cos \theta}{[(1 - \cos \theta) \cdot x + (a \cdot \cos \theta - a - 1)]^2}.$$

- Y como ejemplo general de estas funciones (sin $a=b$):



3.2.5 Perspective Wall

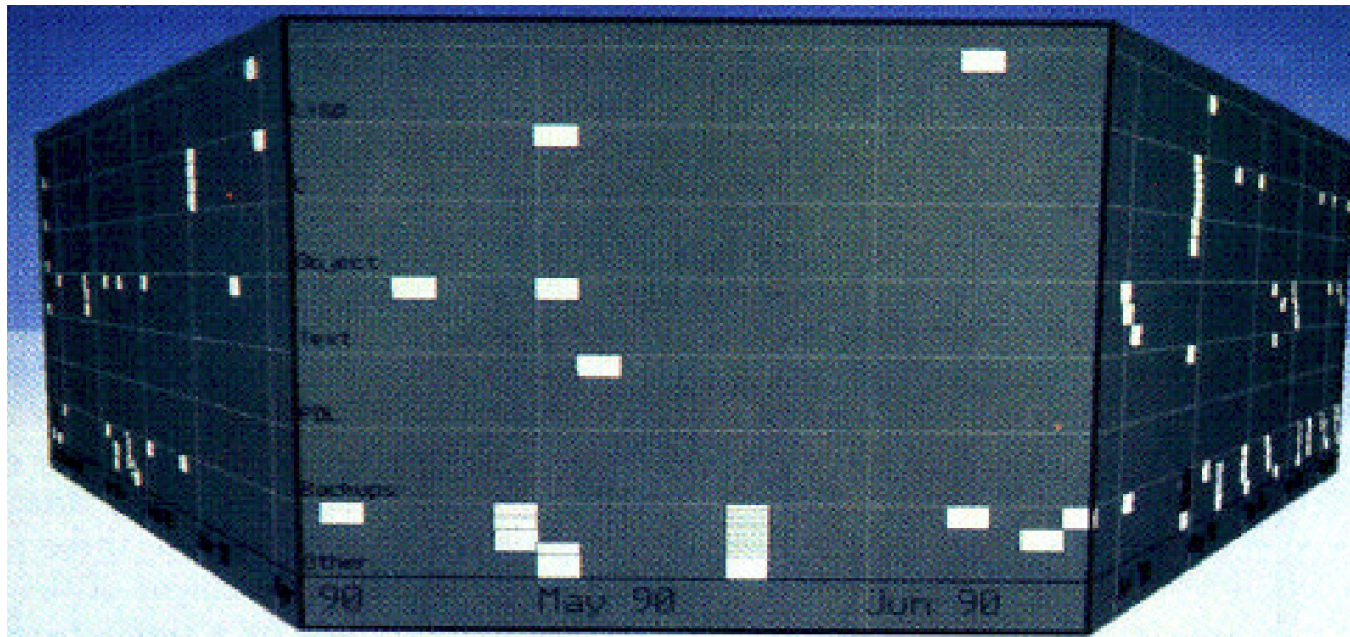
- El efecto generado sobre nuestra figura de referencia sería (2 dimensiones):



- Nótese que se **malgasta espacio** en las esquinas, contradiciendo uno de los objetivos de la visualización de la información que es *aprovechar todo el espacio posible*.

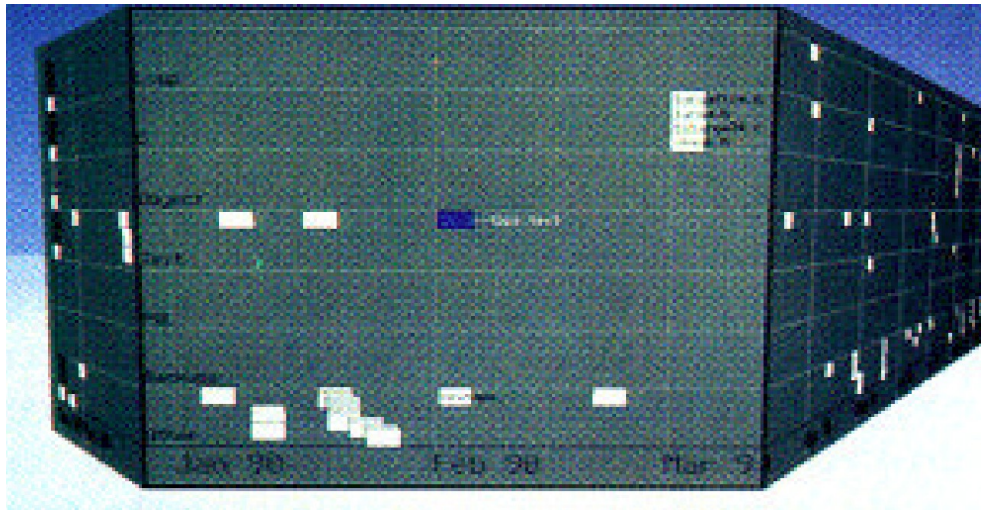
3.2.5 Perspective Wall

- Ejemplo original : visualización de documentos de un sistema ordenados:
 - Horizontalmente por fecha de modificación
 - Verticalmente por tipo de archivo



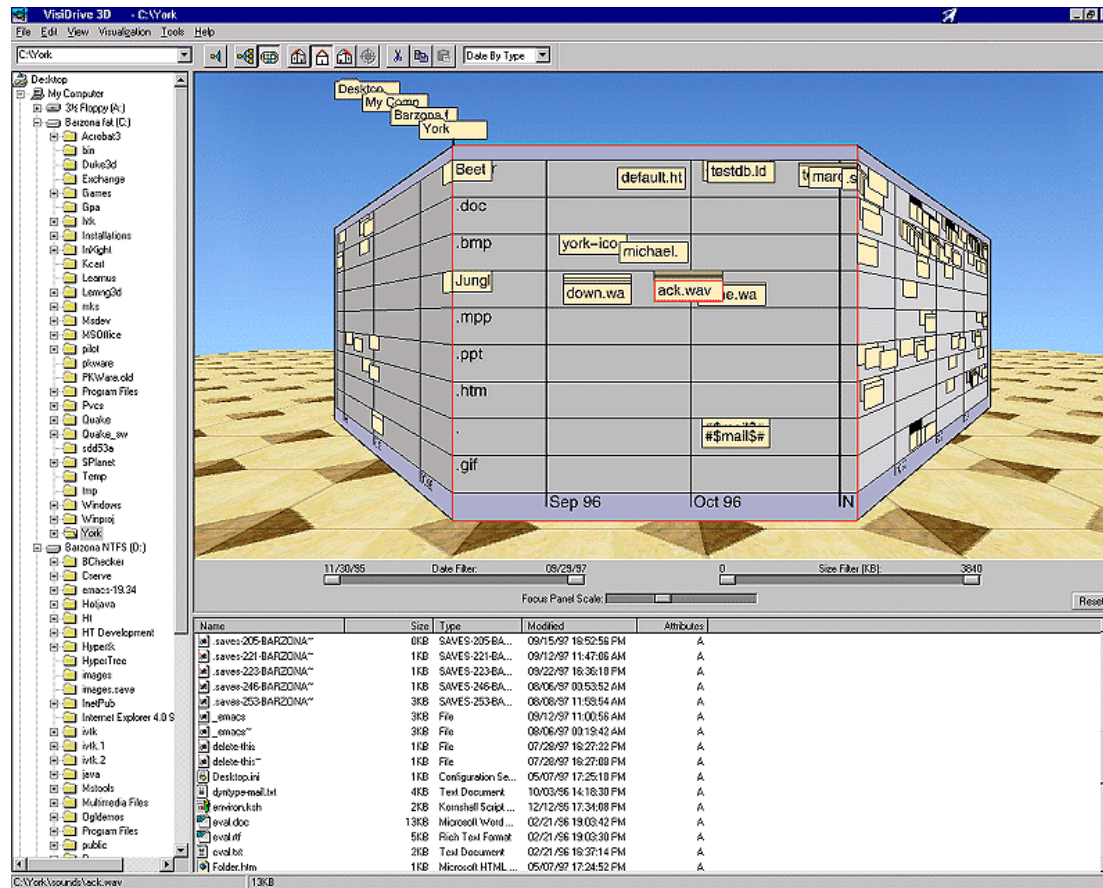
3.2.5 Perspective Wall

- El foco se mueve hacia el panel izquierdo al seleccionar un documento de Febrero:



3.2.5 Perspective Wall

- Aplicación real a un navegador de archivos



3.2.6 Table Lens

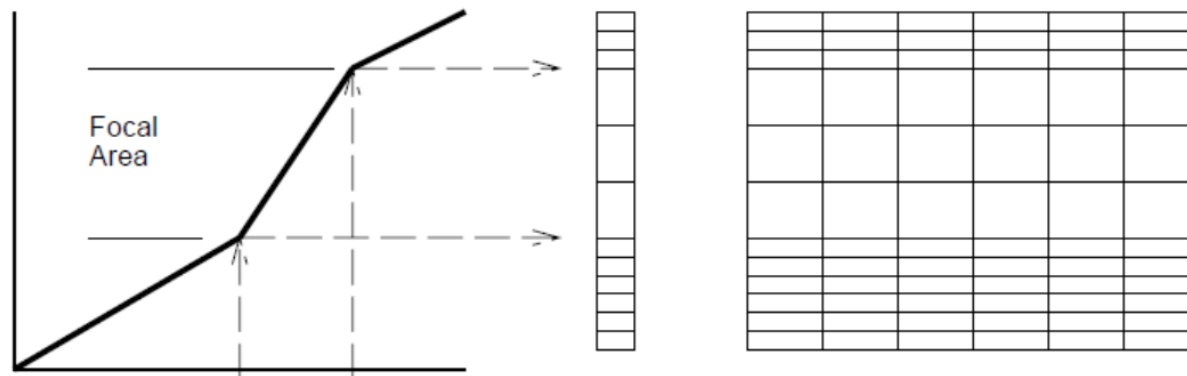
- Basada en la naturaleza tabular de las tablas
- Cambia la distribución de filas y columnas sin doblar sus límites
 - Ejemplo al seleccionar 2 columnas y 3 filas

	G					H				
4					G4		H4			
5					G5		H5			
6					G6		H6			

- Similar al Fisheye de Sarkar y Brown, pero:
 - El foco no se define por puntos sino en filas y columnas
 - Permite multifoco con distintos niveles

3.2.6 Table Lens

- La función de **transformación** sería la siguiente (aplicada a 3 filas):



- Y la función de **magnificación** para este ejemplo:

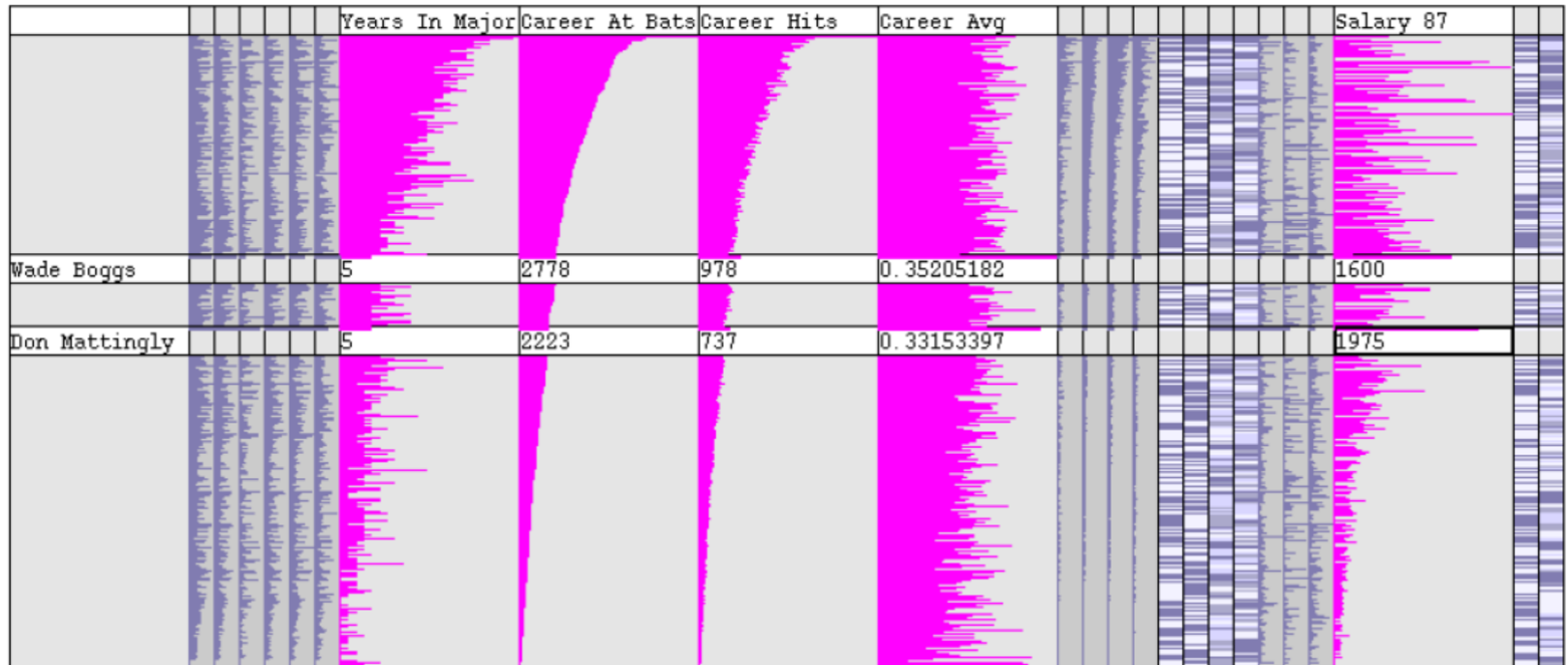


3.2.6 Table Lens

- La **representación del valor** de una celda depende de:
 - *Tipo de Valor:*
 - Textual: texto
 - Numérico: una barra horizontal de longitud proporcional al valor
 - Categórico: color
 - *Tipo de celda:*
 - las celdas en el foco tienen color más fuerte
 - las celdas en el foco tienen son más grandes
 - Las celdas fuera del foco se desmagnifican y no pueden mostrar información detallada
- Vamos a ver un ejemplo de una base de datos que recoge estadísticas de jugadores de baseball en 1986:
 - 323 jugadores (filas)
 - 23 variables (columnas)
 - 17 numéricas
 - 6 categóricas

3.2.6 Table Lens

- Selección 4 columnas, ordenadas según 'Career At Bats' (career avg aún no existe)

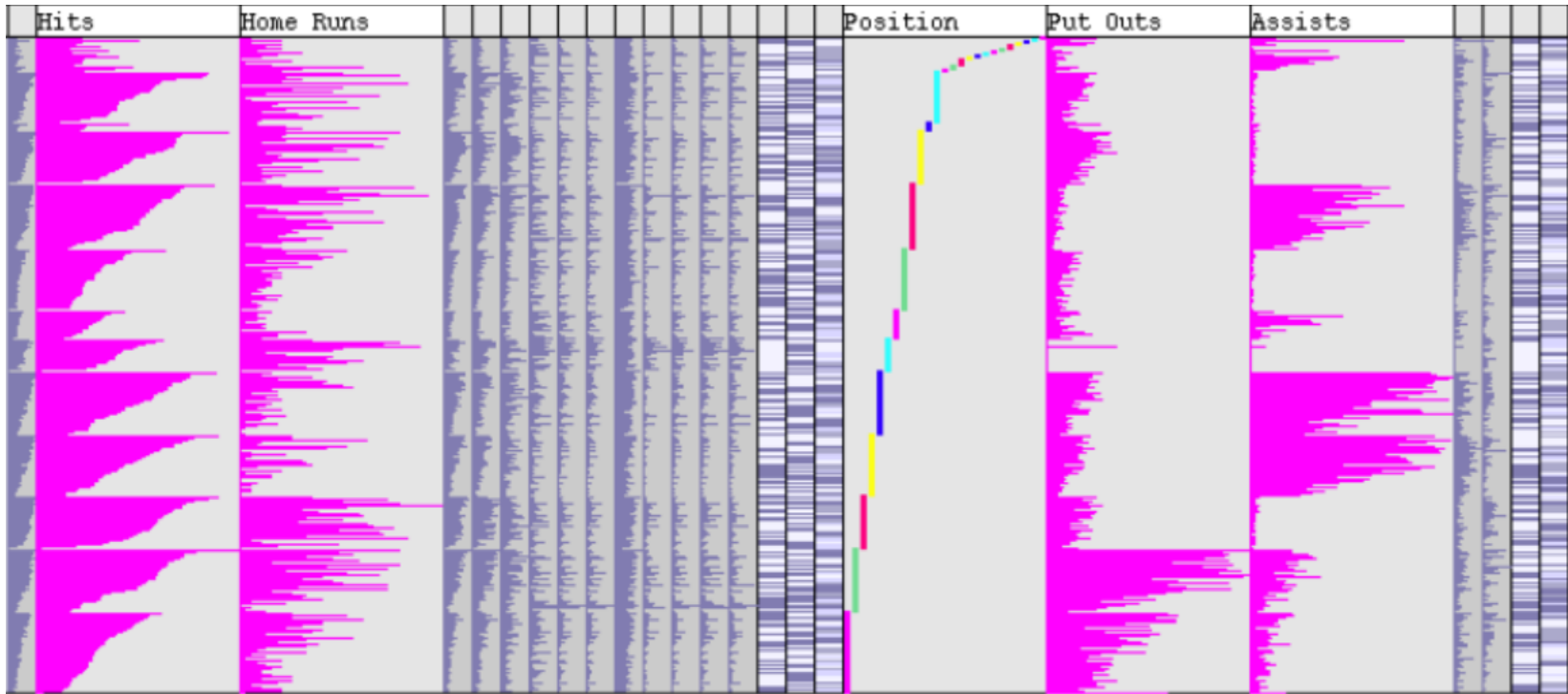


				Years In Major	Career At Bats	Career Hits	Career Avg											Salary 87
Wade Boggs				5	2778	978	0.35205182											1600
Don Mattingly				5	2223	737	0.33153397											1975

- Se seleccionan las 2 filas cuyo valor de "Career Hits" se desmarca de la tendencia decreciente
- Se crea la columna "Career Avg" = Career Hits / Career At Bats
- Se comprueba que tienen el mejor ratio de carreras por 1 golpe que a base de golpes
- se identifica fácilmente quién batea mejor.
- coincide que tienen de los mejores sueldos

3.2.6 Table Lens

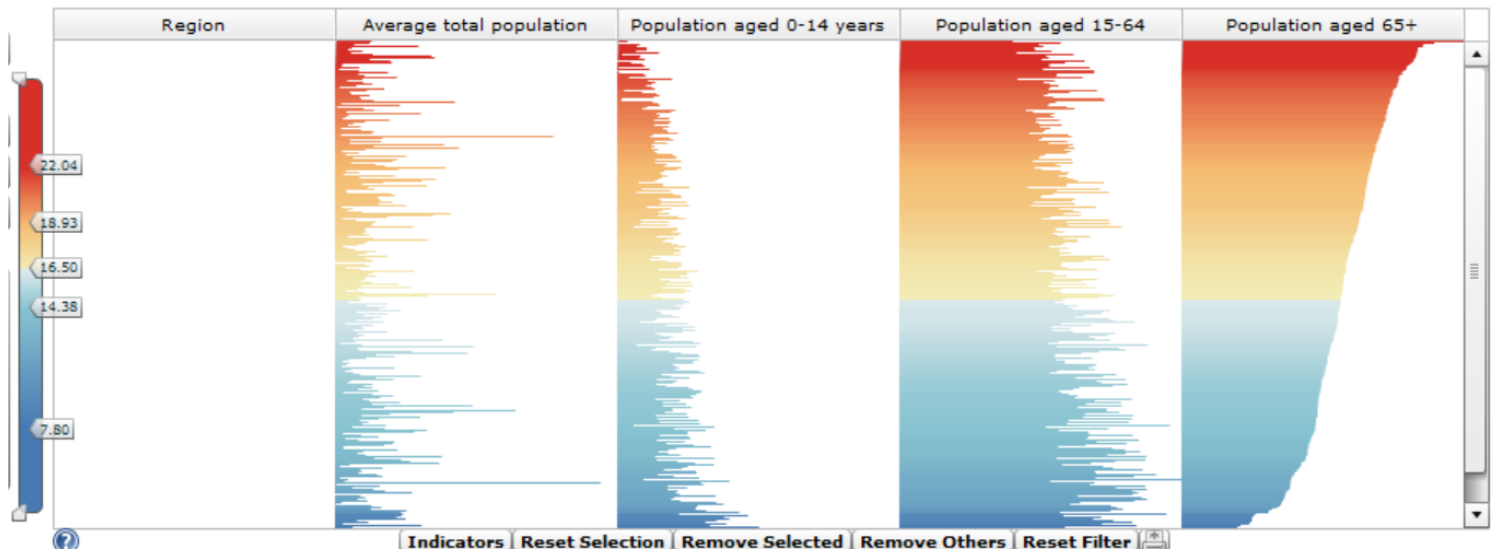
- Se ordenan por Posición del jugador:



- El número de golpes y home runs suelen ser mayores para las posiciones rojo y verde .

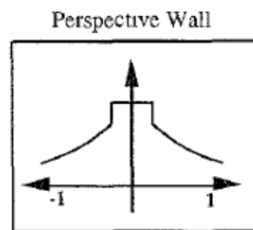
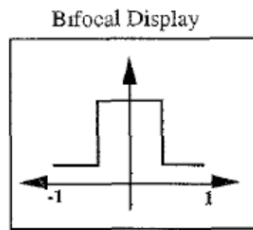
3.2.6 Table Lens

- Vídeo que usa el ejemplo explicado:
<http://www.youtube.com/watch?v=qWqTrRAC52U>
- http://www.ncomva.se/guide/?chapter=Visualizations§ion=Table%20Lens#_General

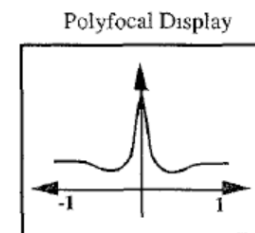
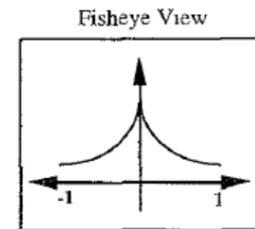


3.2.7 Tipo de Función

- Las técnicas vistas pueden clasificarse según la continuidad de su función de magnificación:
 - No continuas
 - Continuas



Non-continuous
Magnification
Functions



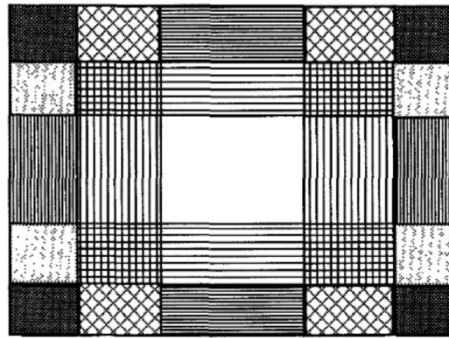
Continuous
Magnification
Functions

3.2.7 Tipo de Función

- Las funciones **continuas** tienden a **distorsionar el marco** limitador de la imagen:
 - Las esquinas son absorbidas hacia el foco
 - Cuanto mayor es el factor de magnificación, mayor alteración del marco
- Hay 3 posibles soluciones:
 - 1) Insertar la imagen final en un marco (como el Fisheye View de Sarkar and Brown en coordenadas polares)
 - 2) Realizar la distorsión primero en el eje x, y luego en el eje y (como el Fisheye View de Sarkar and Brown en coordenadas cartesianas)
 - 3) Convertir las funciones no continuas en continuas, o casi.

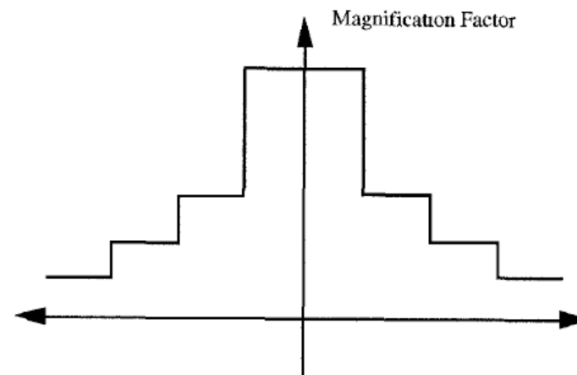
3.2.7 Tipo de Función

- A una función no continua se le pueden añadir muchos niveles de magnificación hasta parecer continua (esto depende de los recursos de la máquina): Bifocal Display con 3 niveles que crea 25 regiones:



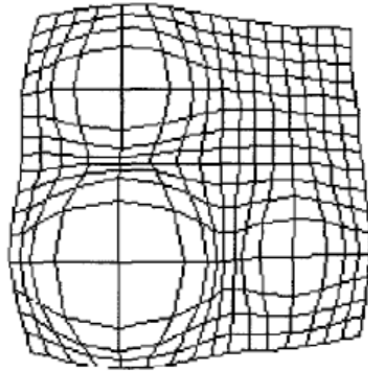
- Una función continua puede discretizarse fácilmente en varios niveles, por ejemplo la de Fisheye View en 4 niveles:

Magnification Function:
A piecewise Fisheye View

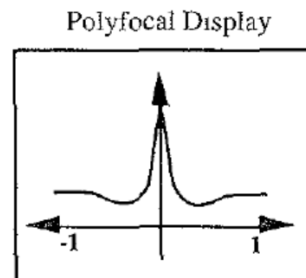


3.2.7 Tipo de Función

- Hemos visto que la Polyfocal View puede ser **multifocal**:



- Según [Leung y Apperley, 1994], esto solo es posible para una técnica de distorsión cuando su función de magnificación tiene '**hoyos**':



3.2.8 Implementación

- Existen varios criterios que se pueden tener en cuenta para mejorar el comportamiento de los sistemas de distorsión
- Métodos con funciones de magnificación con **intervalos discretos**:
 - Cálculo anticipado de las regiones generadas según el foco, y se almacenan en **memoria**

Demagnification in both X and Y dimensions	Demagnification in Y dimension	Demagnification in both X and Y dimensions
Demagnification in X dimension	Central 'Focus' Region no demagnification	Demagnification in X dimension
Demagnification in both X and Y dimensions	Demagnification in Y dimension	Demagnification in both X and Y dimensions

- Si se cambia el foco, se copian y pegan los bitmaps correspondientes a las regiones que hayan cambiado
- Consume mucha memoria pero poco procesado

3.2.8 Implementación

- Los métodos con magnificación continua pueden aplicar cierta discretización, como ya se ha visto.
- Si hay **n niveles** de magnificación, habrá que almacenar en memoria **n^2 bitmaps**.
- A las interfaces que permiten **scrolling** les conviene más almacenar bitmaps y tirar de **memoria**.
- A las interfaces que permiten **seleccionar directamente** una zona, les conviene más hacer los cálculos en tiempo real y tirar de capacidad e **procesamiento**.

3.2.8 Implementación

- Cuando una técnica quiere ampliarse para ser multifocal, si su función de magnificación no contiene ‘hoyos’:
 - El marco se distorsiona
 - En este ejemplo, al magnificar A y B, se magnifican como efecto secundario X e Y.

	X		B	
	A		Y	

- Esto se puede evitar ampliando cada zona en distintas ventanas, pero
 - Consume más recursos
 - Dificulta la navegación al usuario



3.1 Introducción

3.2 Distorsión basada en funciones

3.2.1 Polyfocal Display

3.2.2 Bifocal Display

3.2.3 Fisheye View (Furnas)

3.2.4 Fisheye View (Sarkar and Brown)

3.2.5 Perspective Wall

3.2.6 Table lens

3.2.7 Tipo de función

3.2.8 Implementación

3.3 Otras técnicas

3.3.1 Hyperbolic Trees

3.3.2 Bundles

3.3.3 Document Lens

3.3.4 Hyperbox

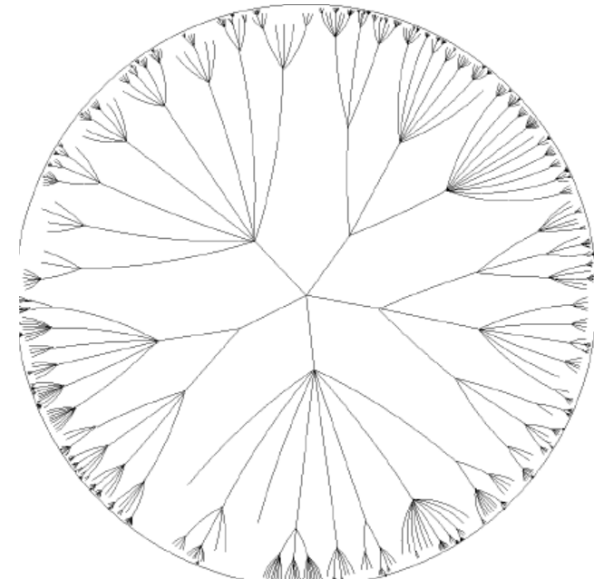
3.4 Conclusiones

3.3.1 Hyperbolic Trees

- Diseñado [Lamping et al., 1995] para la visualización de grandes jerarquías en forma de árbol.
- Efecto parecido a Fisheye
- Se basa en que no existe una distribución satisfactoria en 2D para **árboles** grandes debido a que éstos presentan un **crecimiento exponencial**.
 - La distribución pasa del plano Euclideo al Hiperbólico.
 - La distribución se calcula solo la primera vez
 - Al cambiar el foco, se transforma el plano pero no se vuelve a calcular la distribución

3.3.1 Hyperbolic Trees

- La **distribución de un nodo** en el árbol se realiza en estos pasos:
 1. Se le otorga una porción del plano hiperbólico proporcional a su cantidad de hijos
 2. Se reparte de la misma forma el espacio entre todos sus hijos
 3. Los segmentos que unen a los hijos con el padre son *paralelas en el plano hiperbólico*,
 - asegurando que no se solapen
 - La porción obtenida por cada hijo tendrá el mismo ángulo que el padre
 4. Volver a 2 para cada hijo
 5. Mapear a 2D: modelo de Poincaré
 - Preserva ángulos
 - Arquea las líneas del espacio hiperbólico

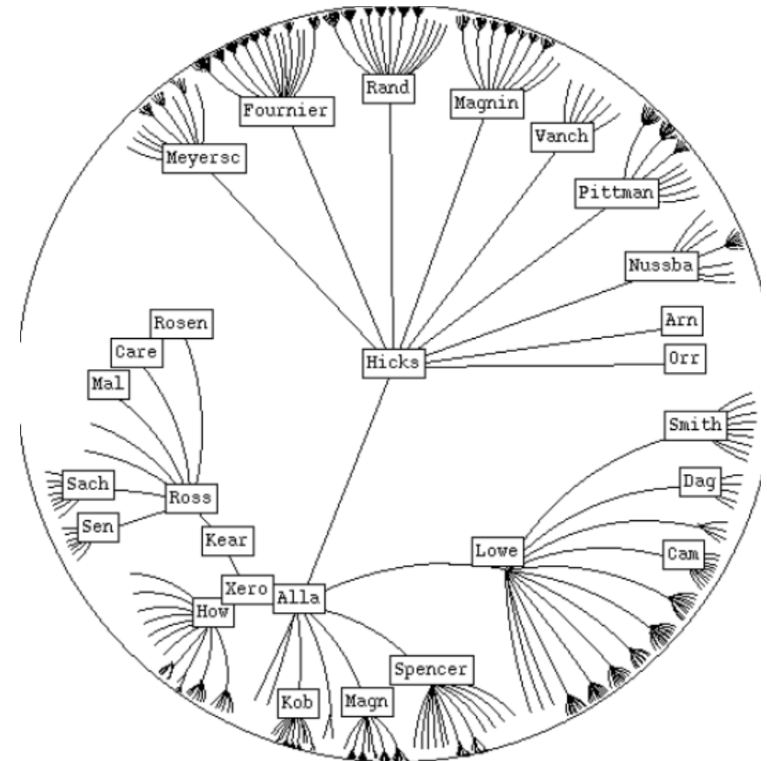
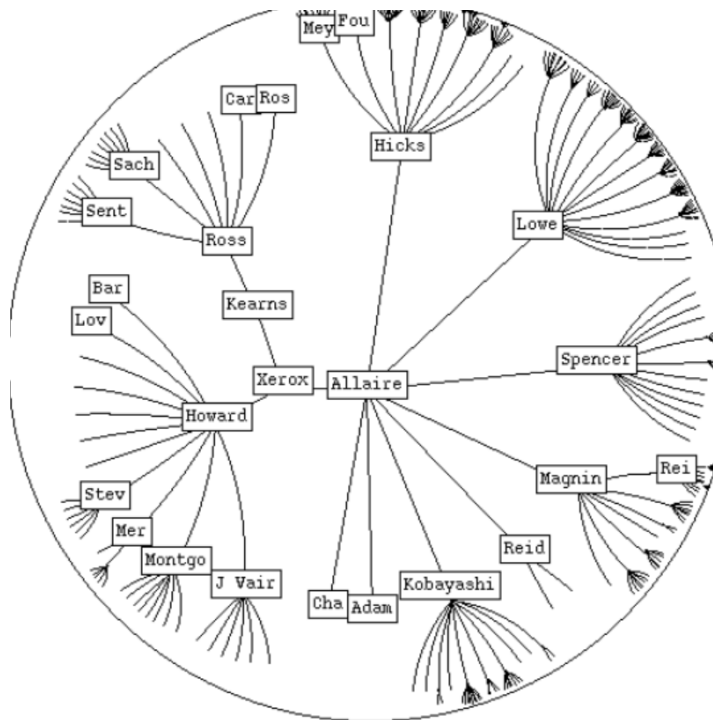


3.3.1 Hyperbolic Trees

- **Cambio de foco:**
 - El usuario hace click en una zona o la arrastra al centro
 - Los nodos que están por el centro se magnifican y el resto se desmagnifican
 - La nueva distribución no se calcula, sino que **se traslada el espacio** hiperbólico de forma que el nodo seleccionado quede en el centro.
 - En la implementación, para no perder precisión de coma flotante a lo largo de varios cambios de foco, las traslaciones del espacio hiperbólico son **acumulativas respecto a la primera distribución.**

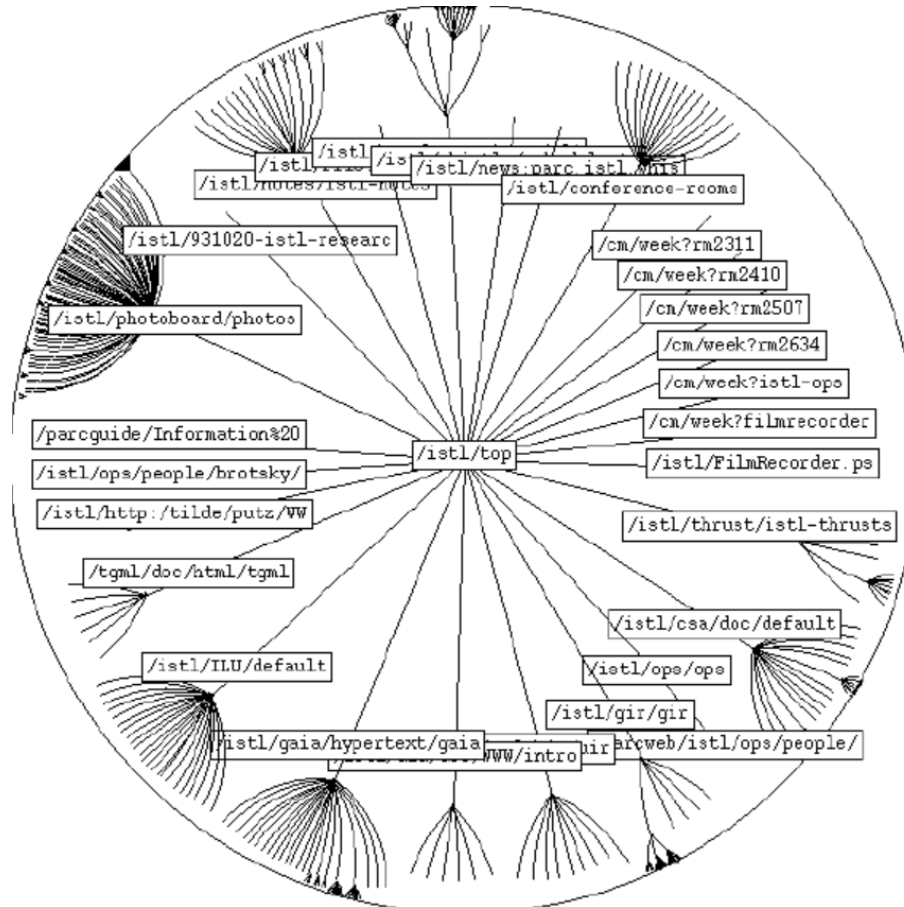
3.3.1 Hyperbolic Trees

- Hyperbolic Tree antes y después de cambiar el foco



3.3.1 Hyperbolic Trees

- Árbol de un espacio de la WWW

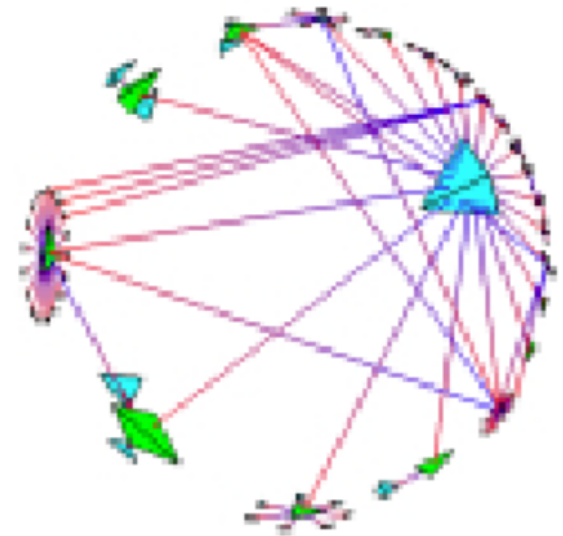
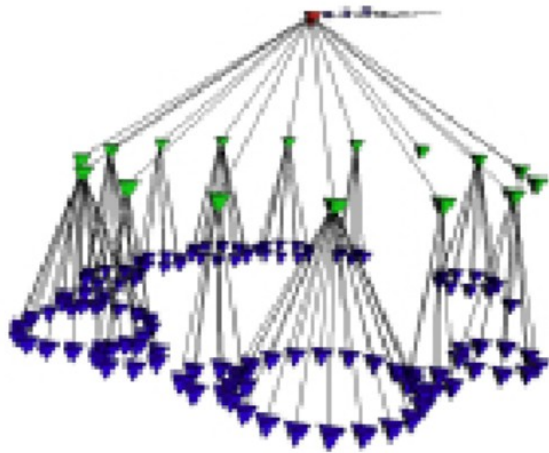


3.3.1 Hyperbolic Trees

- En [Munzner,1995] se presentan árboles hiperbólicos 3D
- También se basa en la idea de que en el espacio hiperbólico cabe más información que en el euclidio, pero:
 - Los nodos son conos
 - El mapeo a 2D se hace con el modelo de Klein:
 - Las líneas se mantienen pero se distorsionan los ángulos
- Es caro computacionalmente y menos interpretable

3.3.1 Hyperbolic Trees

- Árbol en espacio Euclideo vs. espacio Hiperbólico

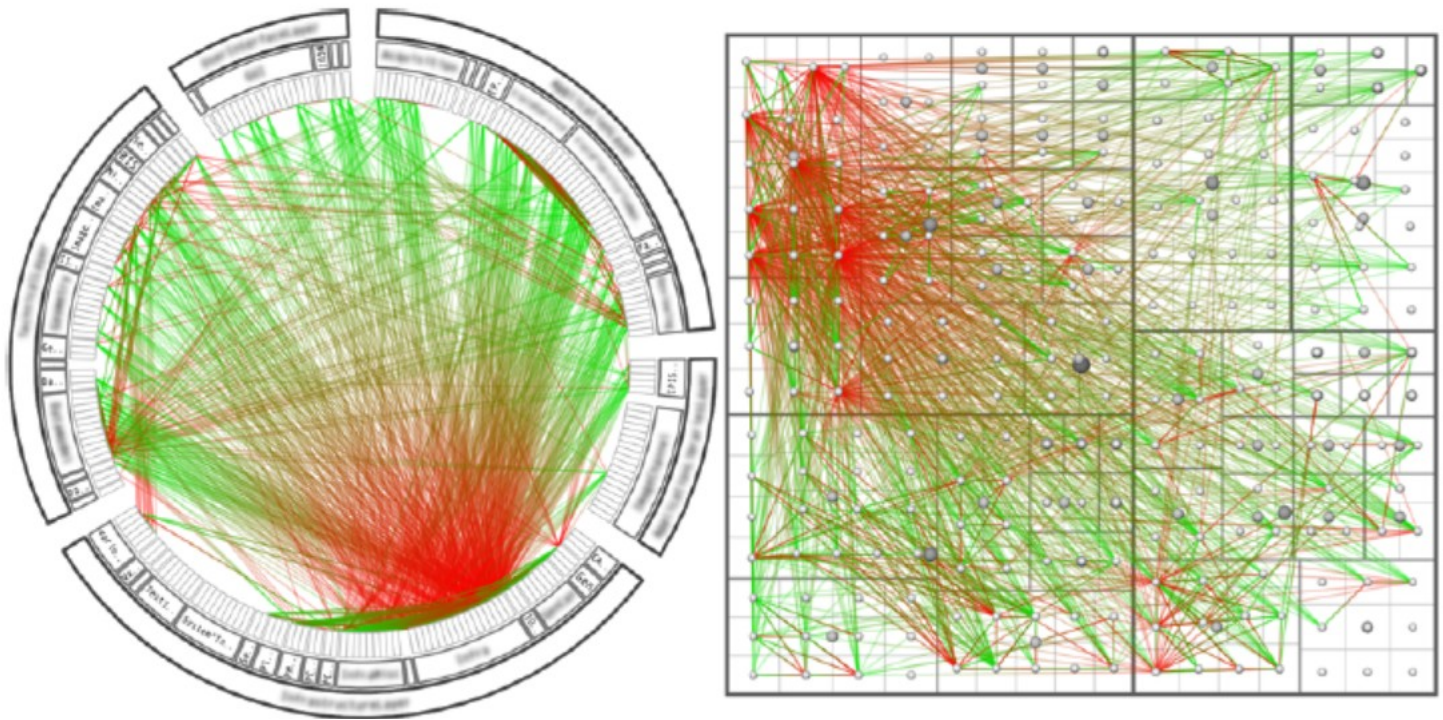


3.3.2 Bundles

- Se propone [Hotel, 2006] “atar” los arcos adyacentes en un camino: similar a los cables de corriente eléctrica, que solo se separan en el inicio y el final.
- Aplicable a cualquier técnica de visualización de árboles:
 - Elimina las flechas
 - Color verde para salientes
 - Color rojo para entrantes
- Ejemplo para ver las llamadas entre componentes de un software de un escáner médico:
 - 284 nodos
 - Más de mil adyacencias entre arcos

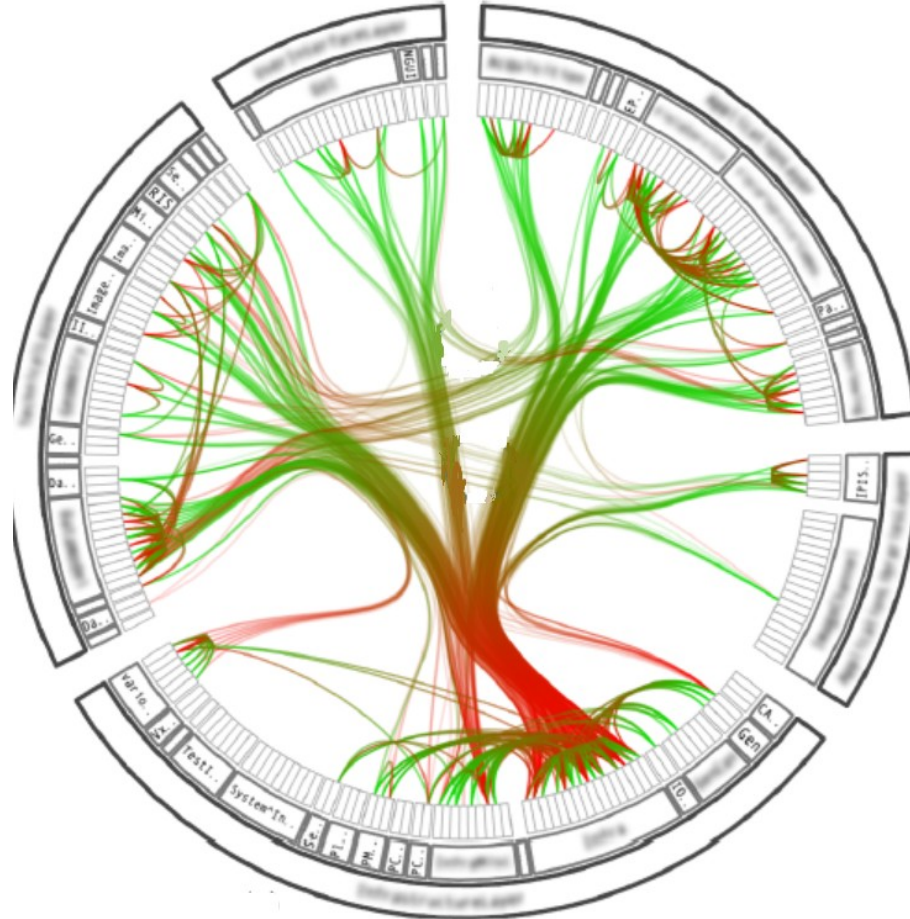
3.3.2 Bundles

- Grafo sin bundles, con visualización radial y visualización con TreeMap



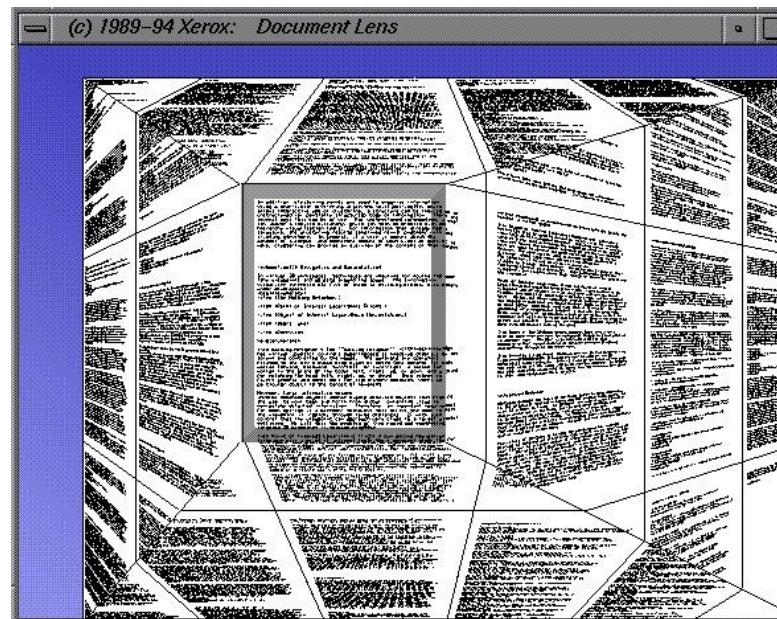
3.3.2 Bundles

- Resultado en visión radial tras atar arcos adyacentes



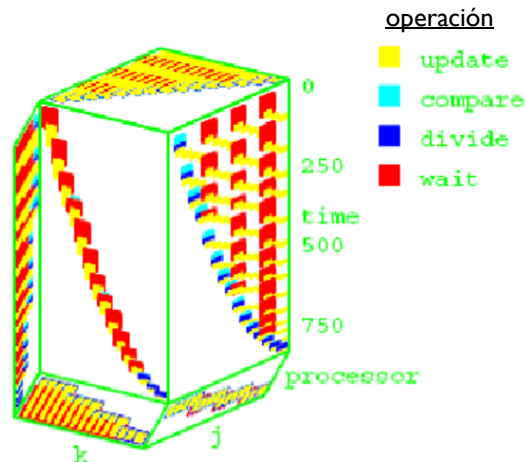
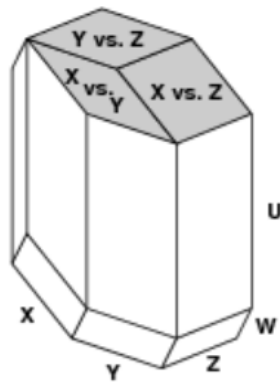
3.3.3 Document Lens

- Propuesto en [Robertson and Mackinlay, 1993] para la visión de múltiples documentos en una región rectangular.
- Produce el efecto de una lente con forma de pirámide truncada
 - El foco están en el centro
 - El contexto son los documentos en los lados de la pirámide



3.3.4 Hyperbox

- Aparece en [Alpern and Carter, 1997]
- Cada cara representa un par de variables.
- Un **conjunto direccional** es un conjunto de segmentos con la misma dirección
- Se asigna un conjunto direccional a cada variable



*Comportamiento de
Procesadores paralelos*



3.1 Introducción

3.2 Distorsión basada en funciones

3.2.1 Polyfocal Display

3.2.2 Bifocal Display

3.2.3 Fisheye View (Furnas)

3.2.4 Fisheye View (Sarkar and Brown)

3.2.5 Perspective Wall

3.2.6 Table lens

3.2.7 Tipo de función

3.2.8 Implementación

3.3 Otras técnicas

3.3.1 Hyperbolic Trees

3.3.2 Bundles

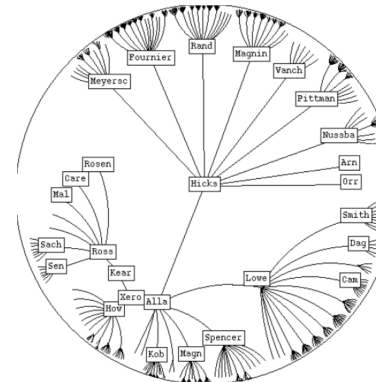
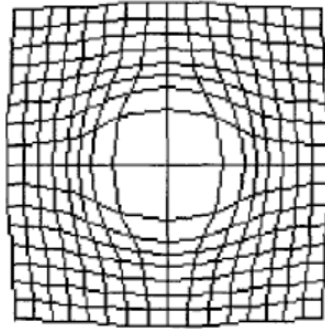
3.3.3 Document Lens

3.3.4 Hyperbox

3.4 Conclusiones

3.4 Conclusiones

- Distorsionar consiste en magnificar la zona del foco y desmagnificar el resto
- Existen técnicas basadas en funciones matemáticas y otras en algoritmos de distribución.



- En tiempo de implementación, podemos seleccionar si queremos que el peso de la distorsión sea soportado por capacidad de memoria o por capacidad de procesado.
- Para el usuario es deseable poder interaccionar para elegir la zona del foco de interés. Además de esta, en el siguiente tema veremos distintas técnicas de interacción.