



Refinamiento de los algoritmos de dimensionado y posicionamiento de nodos en Árboles de Conos

Franklin Hernández C.

franklin@lincos.net

Escuela de Diseño Industrial

Instituto Tecnológico de de Costa Rica

Jorge Monge F.

jomonge@itcr.ac.cr

Escuela de Matemática

Instituto Tecnológico de de Costa Rica

Resumen

En el presente artículo se estudiaron con detalle los algoritmos de posicionamiento, dimensionado y rotación dinámica de conos actualmente conocidos en árboles de conos. Se realiza un análisis orientado a refinar los algoritmos, entender su entorno y mejorar su eficiencia desde el punto de vista de tiempo de ejecución así como la necesidad y posibilidad de que se presenten choques entre los subárboles. Se ofrece también una mejora al algoritmo base para disminuir los casos en que se presentan estos choques.

Palabras claves: Visualización de información, navegación, cone-trees, rotaciones de conos, árboles de conos, BioVisualizador.

1.1 Introducción

Los "árboles de conos" son representaciones tridimensionales de estructuras de árbol, que se representan en el espacio tridimensional. Son representaciones del

tipo de nodos vinculados y parte de las técnicas de visualización sin distorsión ([9]). La raíz del árbol se localiza en el extremo superior de un cono, y los hijos de éste se organizan alrededor de la circunferencia base del cono. Además cada hijo puede ser raíz de un subárbol representado del mismo modo en forma recursiva. Ver (1.1).

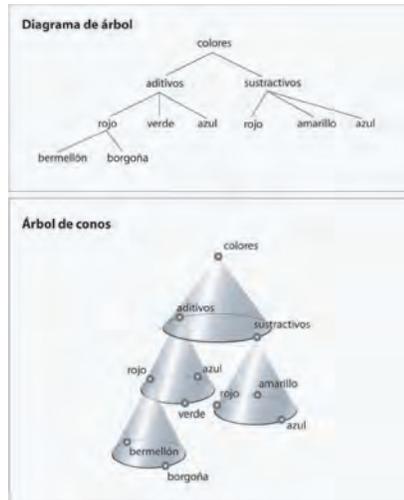


Figura 1.1 Paradigma de árboles de conos.

Para su representación en sistemas digitales la altura del árbol se define a priori para que se ajuste al espacio disponible en la pantalla. De igual forma la altura en cada nivel se mantiene constante y es obtenida al dividir la altura disponible entre la profundidad del árbol. Los diámetros de las bases de los conos se reducen en progresión garantizando que los conos del último nivel "caben" en el espacio disponible. En términos generales los árboles de conos buscan maximizar el espacio tridimensional disponible, sin embargo, dentro de las desventajas que presenta está la implementación de las alturas ([13]) se puede mencionar que debido a la razón de crecimiento constante descrita anteriormente, la cantidad de niveles visualizables en una computadora estándar se limita a 10 con alrededor de 1000 nodos

Refinamiento de los algoritmos... F. Hernández, Jorge Monge.

Derechos Reservados © 2009 Revista digital Matemática, Educación e Internet (www.cidse.itcr.ac.cr/revistamate/)

1.1.1 Definición del problema

Este paradigma es similar al de representar las relaciones jerárquicas por medio de árboles, solo que en este caso se trabaja en el espacio tridimensional y, en lugar de utilizar líneas para establecer las relaciones, utilizamos conos. La utilización de conos en un espacio tridimensional implican ciertos retos, entre ellos encontramos:

1. La distribución de los conos en el espacio
2. La distribución de las hojas sobre la base del cono
3. La altura de los conos
4. La ubicación y la orientación exacta de las hojas sobre la base de los conos

Los algoritmos propuestos hasta el momento presentan el inconveniente de que bajo ciertas condiciones en la estructura del árbol se presentan traslapes como se muestra en la figura 1.2



Figura 1.2 Visualización del problema.

1.1.2 Estado del arte

Para el establecimiento de la dimensión y el posicionamiento de los conos, existe un algoritmo clásico y algunas modificaciones que se han realizado en las últimas

dos décadas. Seguidamente analizamos rápidamente algunas de estas propuestas.

1.1.3 Artículo de Kazman/Carriere ([5])

En 1995 se publica un primer artículo sobre el posicionamiento en árboles de conos, en esta cita, considerada como clásica en el tema de árboles de conos, se describe el primer algoritmo formal para su distribución en el espacio. La decisión principal en la construcción de árboles de conos es el radio de los mismos. Es decir, si un "nodo padre" tiene tales hijos de qué tamaño debería de ser. La respuesta clásica expuesta en este primer enfoque del 95 es, el cono padre debería de tener una circunferencia igual a la suma de los radios de todos sus hijos. Ver figura 1.3.

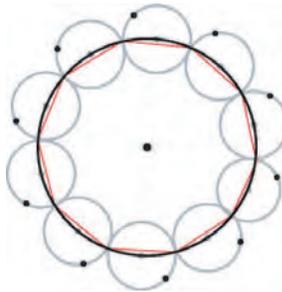


Figura 1.3 Estrategia base publicado por Kazman y Carriere en 1995.

Con este algoritmo se pueden dar choques en la construcción del árbol pues no toma en cuenta los casos en los que los hijos son muy heterogéneos. No compara el radio del padre con el de los hijos, de modo que si un padre tiene un hijo que, a su vez, tiene muchos hijos, el padre resultará con un cono de menor tamaño que su hijo. Ver figura 1.4.

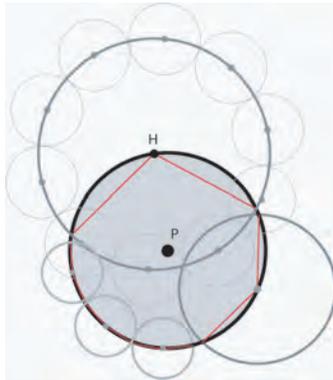


Figura 1.4 Situación en la que un hijo puede resultar con un diámetro mayor al de su padre. En la figura el hijo H tiene muchos hijos y como consecuencia su diámetro resulta mayor que el de su padre P

A las anteriores consideraciones se le debe añadir que al sumar los radios de los hijos, el valor de la circunferencia resultante es solo aproximado, pudiendo resultar mucho menor de lo necesario en casos muy heterogéneos. Ver figura 1.5.

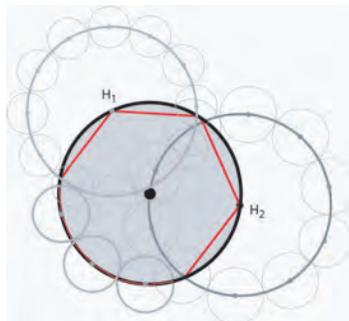


Figura 1.5 Caso de choques siguiendo la estrategia Kazman, aunque los diámetros de los hijos 1 y 2 estén definidos en función de los diámetros de sus descendientes aun así se produce el choque

1.2 Artículo de Chang-Sung/Pang ([4])

Este artículo publicado tres años después mejora el anterior pues después de sumar los radios de los hijos de un cono, compara ese valor con el radio del hijo de mayor diámetro y de ser menor toma este último como radio para el padre. Esta comparación asegura que ningún padre tenga un diámetro menor que alguno de sus hijos. Ver figura 1.6

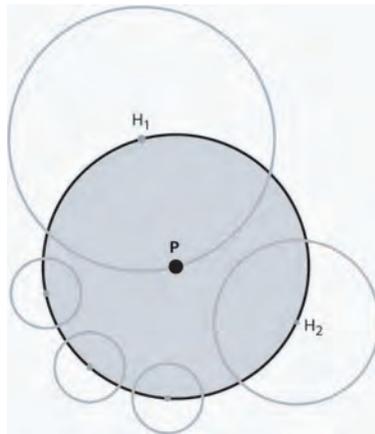


Figura 1.6 Corrección Chang-Sung/Pang, en este caso el nodo padre recibe el diámetro igual al de su mayor hijo H1 a pesar de que por algoritmo debería ser menor.

Sin embargo, esta corrección evita los choques pues no toma en cuenta la posición angular de los hijos y en el caso de que estos se encuentren cerca, los choques aun son posibles. Ver figura 1.8.

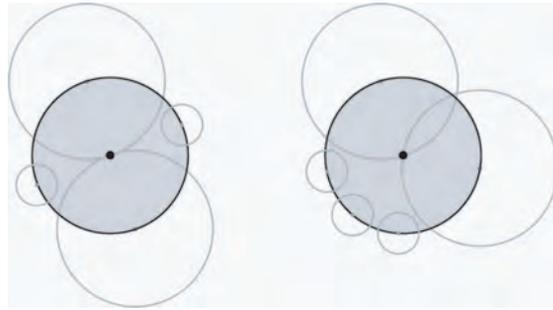


Figura 1.7 Problemas de choques en la estrategia Chang-Sung/Pang, mientras en la figura de la izquierda el algoritmo logra su cometido, en la figura de la derecha se observa que un cambio en la distribución angular de los mismo hijos podría generar choques.

1.3 Artículo de David Auber ([1])

Varios años después, para 2004, encontramos una estrategia basada en el círculo que inscribe a todos los elementos de un cono y sus hijos (inscrito en vista superior), sin embargo, este círculo no es concéntrico al cono y por tanto es difícil de manejar. De hecho David Auber no contempla giros de subárboles, de este modo los árboles así creados son más compactos pero muy estáticos y a la menor rotación se produciría un choque. Ver figura 1.8

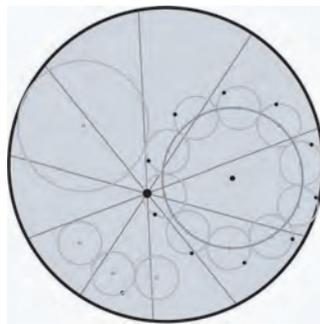


Figura 1.8 Estrategia de David Auber 2004.

las dos circunstancias en las que se pueden provocar colisiones entre subárboles son:

1. En la generación inicial del árbol
2. En la ejecución de una secuencia de rotación sincrónica (conocida como "Bring to Front" o BTF)

1.4.1 Posibilidad de choques en la generación inicial de los árboles

1.4.1.1 Análisis teórico La posibilidad de que se generen choques en el proceso de generación del árbol está ligada al algoritmo de formación de éstos. Es importante clarificar cómo funciona el algoritmo de dimensionado y posicionamiento de conos dentro del Biovisualizador (Sistema actual desarrollado por los autores en el Instituto Tecnológico de Costa Rica para el Instituto Nacional de Biodiversidad InBio). Para esto consideremos la figura (1.10).

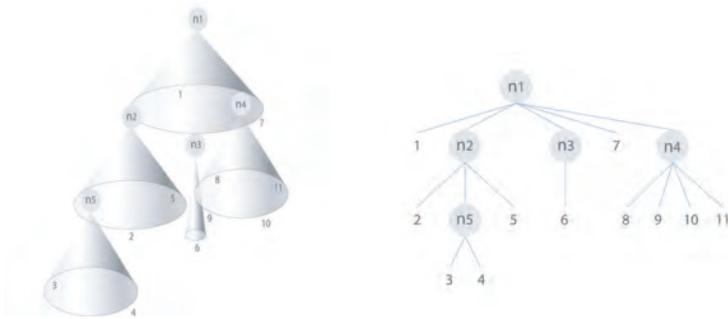


Figura 1.10 Taxonomía de construcción en los árboles de conos en el Biovisualizador.

Las etiquetas puestas en la estructura sugieren la forma en que se visualiza la estructura para el algoritmo, es decir se hace una lectura de izquierda a derecha y en profundidad. Es decir, una estructura general se vería como sigue:

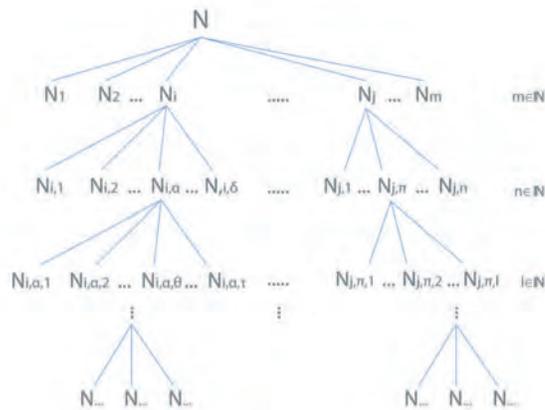


Figura 1.11 Taxonomía general de construcción en los árboles de conos.

Donde N corresponde al nodo padre, sin pérdida de generalidad pues esta estructura podría estar presente en cualquier subárbol y en consecuencia producir los mismos efectos. Consideraciones sobre el árbol representado por la estructura de la figura (1.12):

1. o Para los nodos N_k con $k = 1...m$ existe asimetría en el número de hijos. Por ejemplos el número de hijos de N_i y N_j en comparación con los hijos de los otros nodos en el mismo nivel.
2. o Si la distancia entre i y j es pequeña, por lo que conforme $j \rightarrow i + 1$ el traslape ocurrirá en niveles cercanos a la raíz.
3. o Si la distancia entre i y j tiende a crecer, el traslape podría ocurrir si los nodos N_i y N_j crecen en profundidad y el traslape ocurriría en niveles lejos de la raíz.
4. o Para el subárbol con raíz N_i se espera que $s \rightarrow \alpha + 1$ además de mantener la asimetría en el número de hijos de $N_{i,\alpha}$
5. o Para el subárbol con raíz N_i se espera que $s \rightarrow \alpha + 1$ además de mantener la asimetría en el número de hijos de $N_{i,\alpha}$

6. o Bajo las condiciones impuestas a los nodos $N_{i,\alpha}$ y $N_{i,\alpha}$ en el punto 4 y 5 se produce traslape en la construcción del árbol de conos. El nivel donde ocurren dependerá de $s \rightarrow \alpha + 1$.

Si no se producen choques o traslapes durante la construcción del árbol de conos, es posible que el usuario con las operaciones definidas en el BioVisualizador alcance una estructura similar a la mostrada en la figura (1.12). El traslape planteado tanto en la construcción como la que se obtiene por la aplicaciones de un número de operaciones, podría ocurrir en un punto fuera del foco de atención del usuario por lo que podría no ser percibido. Para el caso particular de un subárbol como el de la figura (1.12), se tiene que, por el algoritmo convencional definido por Carriere y Kazman ([5]), la longitud de la circunferencia y del radio vendrían dadas por:

$$D_{j,m} = 2(r_{j,m} + r_{\max}), \quad D_{j,m} = r_{j,m} + r_{\max}$$

sin embargo, cuando el número de nodos es muy grande (mayor a 10) y hay simetría en el número de hijos, el aporte de r_{\max} es muy pequeño comparado con $D_{j,m} = 2r_{j,m}$

La figura (??) muestra un caso particular, el nodo j tiene k hijos cuyo radio de la circunferencia es igual a r para todo hijo del nodo j .

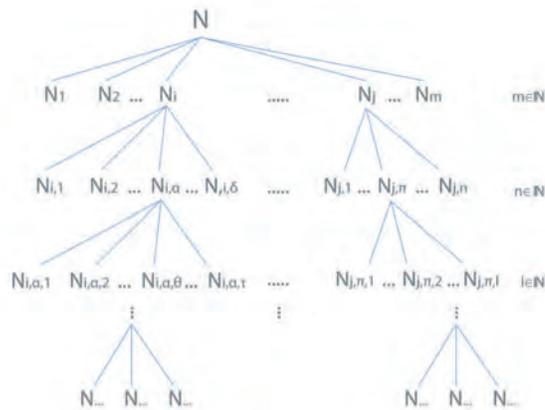


Figura 1.12 Caso particular del problema

Como se indicó anteriormente la longitud de la circunferencia del nodo j vendría dada por:

$$C_{j,m} = \sum_{i=1}^k 2r_{j,m-1}$$

y el radio máximo de los k hijos sería

$$r_{\max} = r$$

De esta forma si definimos el cociente

$$\frac{r_{\max}}{C_{j,m}} = \frac{r}{\sum_{i=1}^k 2r_{j,m-1}} = \frac{r}{\sum_{i=1}^k 2r} = \frac{r}{2kr}$$

simplificando obtenemos

$$\frac{r_{\max}}{C_{j,m}} = \frac{1}{2k}$$

Por lo que si el número de nodos es grande (mayor a 10) el aporte del radio máximo es mínimo a la circunferencia total. Situación similar ocurre cuando los radios no son iguales pero hay cierta simetría con el número de hijos de los k hijos del nodo j .

Empíricamente el valor obtenido para este cociente llega proporcionalmente a ser inferior a 30% por lo que su aporte a la circunferencia es mínimo y así aumenta la posibilidad de choques como muestra la figura 12. En el algoritmo propuesto por los autores en el Biovisualizador, se calcula este cociente y si este valor llega a ser menor al 35%, se establece la siguiente corrección experimental

$$D_{j,m} = 2(r_{j,m} + 5r_{\max})$$

permitiendo disminuir considerablemente la posibilidad de choques.

1.4.1.2 Análisis experimental Las caracterizaciones sobre la estructura permiten tener un escenario con menos posibilidades de choques, para fortalecer esta argumentación se requiere un análisis sobre un mayor número de estructuras. Dado que el Biovisualizador fue hecho exclusivamente para la estructura de anfibios que constaba únicamente de 160 nodos, fue necesario crear una rutina que generara estructuras jerárquicas aleatorias sujetas a ciertos parámetros como el número de niveles y el número máximo de hijos para cada nodo. Por otro lado para determinar si hay o no hay choques en la construcción, se tenía como único recurso dar un vistazo a la construcción y determinar a ojo si había o no choques, esto hacía que el trabajo fuera más laborioso. Además entre más grandes son las estructuras, el tiempo de ejecución de algoritmo de desplazamiento del árbol empieza a crecer en forma considerable. Para evitar esta situación se programó una rutina que permitiera la detección de choques sin necesidad de graficar el árbol de conos. Este algoritmo realiza una verificación en forma exhaustiva por niveles y cuando encuentra el primer choque informa. Este algoritmo tiene un tiempo de ejecución que crece exponencialmente crece el número de nodos en la estructura,

por ejemplo para una estructura de 3000 nodos aproximadamente, el tiempo que tarda el algoritmo para hacer la verificación es de al menos 50 horas, utilizando como equipo una computadora portátil estándar. Por esta razón únicamente se trabajó con estructuras con niveles inferiores a 4. Las pruebas se realizaron tanto para el algoritmo normal como para el algoritmo refinado (el que contenía la corrección explicada en el apartado anterior). Un experimento con un tamaño de muestra de 50, y para árboles con un nivel máximo de 3 niveles y 4 niveles con un número de hijos máximo para cada nodo de 20. El análisis de normalidad con la prueba de Kolmogorov-Smirnov nos dio resultados favorables a la normalidad de los datos. Ambos valores son claramente superiores al nivel de significancia predefinido(5%) por lo que la distribución del número de nodos para árboles con nivel no mayor a tres sigue una distribución normal. La siguiente tabla muestra algunos estadísticos de interés relacionado a la existencia o no de choques tanto para el algoritmo original como el algoritmo refinado:

Algoritmo		Frecuencia	Porcentaje	Porcentaje Válido	Porcentaje acumulado
Sin refinar	No hubo choque	35	700	700	700
	Hubo choque	15	300	300	1000
	Total	50	1000	1000	
Refinado	No hubo choque	44	880	880	880
	Hubo choque	6	120	120	1000
	Total	50	1000	1000	

Tabla 1.1 Estadísticas de choques

En este caso podemos ver que hay una leve mejoría con la disminución de choques existentes, la tabla 1 muestra que la media de nodos para ambos casos anda alrededor de 265 nodos para la estructura. Para el caso de árboles con un nivel no mayor a 4, la prueba de Kolmogorov-Smirnov nos dio resultados favorables a la normalidad.

Al igual que en el caso anterior ambos valores fueron claramente superiores al nivel de significancia (5%) por lo que la distribución del número de nodos para árboles con nivel no mayor a 4 sigue una distribución normal, esto debería ser así dado que el algoritmo no sufre ninguna modificación y el número de nodos está creciendo . La siguiente tabla muestra algunos estadísticos de interés relacionado a la existencia o no de choques tanto para el algoritmo original como el algoritmo

refinado:

Algoritmo		Frecuencia	Porcentaje	Porcentaje Válido	Porcentaje acumulado
Sin refinar	No hubo choque	9	180	180	180
	Hubo choque	41	820	820	1000
	Total	50	1000	1000	
Refinado	No hubo choque	37	740	740	740
	Hubo choque	13	260	260	1000
	Total	50	1000	1000	

Tabla 1.2 Comparación de choques entre algoritmos

En este caso podemos ver que hay una gran diferencia, el número de choques para el algoritmo normal creció considerablemente en comparación con el algoritmo refinado que mantuvo en valor promedio de un 15% de choques para estructuras con un promedio de 800 nodos . Aunque es claro que el algoritmo refinado mejora considerablemente la proporción de choques, no se debe olvidar que esta disminución podría ir en detrimento de un aprovechamiento adecuado del espacio. Recordemos que el óptimo en cuanto a la no existencia de choques genera estructuras en las que no se hace un uso adecuado del espacio, por lo que la búsqueda del umbral adecuado se caracterizaría por una cierta disminución de los choques pero manteniendo un uso razonable del espacio. Las siguientes imágenes muestran una estructura con 1100 nodos desplegada utilizando cada uno de los algoritmos, ver figuras (1.13) y (1.14) .



Figura 1.13 Estructura desplegada con el algoritmo normal

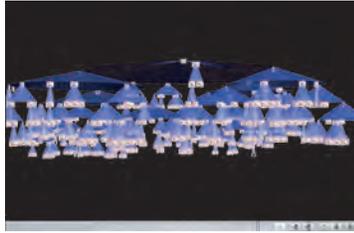


Figura 1.14 Estructura desplegada con el algoritmo refinado

Una mejor vista nos permite visualizar pequeños cambios en el tamaño de la estructura, pero que no parecen ser significativos. Ver figuras (1.15) y (1.16).



Figura 1.15 Estructura desplegada con el algoritmo normal

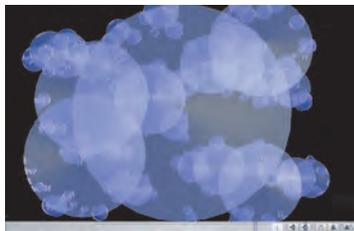


Figura 1.16 Estructura desplegada con el algoritmo refinado

Por lo que al parecer la corrección en el algoritmo minimiza la proporción de choques y no compromete el uso del espacio utilizado en el despliegue del árbol de conos.

1.4.2 Definición de los árboles sin posibilidad de choques

La configuración de árboles que excluyan por completo la posibilidad de choque obliga a éstos a ser demasiado extensos, es decir, muy ralos. Se realizó un análisis de peor caso y se concluyó que las condiciones suficientes para que no haya choques ante ninguna secuencia de operaciones BTF llevan a la construcción de árboles cuyo diámetro crece exponencialmente con respecto a la profundidad del árbol, por lo cual resultan imprácticos, no solo por el ineficiente uso del espacio, sino por la poca frecuencia con que se puede dar estos escenarios de peor caso y por que, como se explica en las siguientes secciones, en el raro caso que se de un choque, éste se daría en una región poco visible para el observador.

1.4.3 Posibilidad de choques en la ejecución de una secuencia de rotación sincrónica (conocida como "Bring to Front" o BTF)

Las condiciones necesarias para que se dé una colisión durante un BTF se pueden resumir del siguiente modo:

1. Dos subárboles "hermanos" deben tener muchos hijos
2. Los dos subárboles deben de estar cerca angularmente uno del otro, es decir, con un ángulo parecido de distribución desde su padre
3. Los subárboles deben de tener su descendencia ordenada en línea
4. El usuario debe de sacar un nodo que esté a 90 grados en el sentido anti-horario del subárbol primero en el sentido horario.
5. El usuario debe de sacar un nodo que esté a 90 grados en el sentido horario de otro subárbol. Ver figura (1.17).

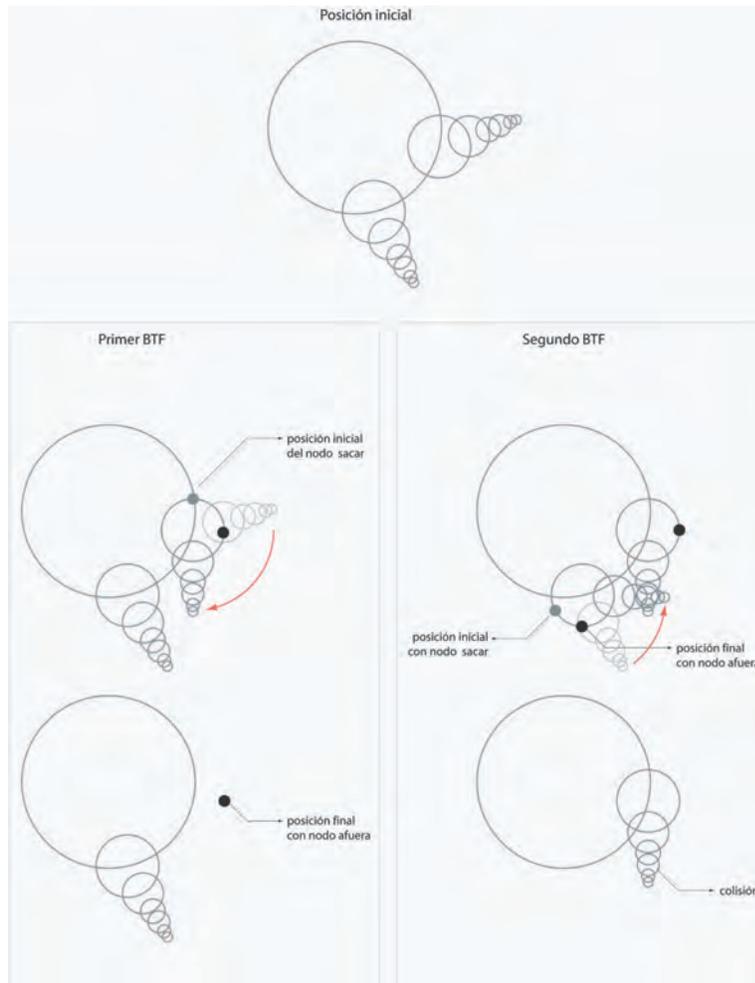


Figura 1.17 Condiciones para la generación de un choque durante una secuencia de BTFS

1.4.3.1 Niveles en los que se puede producir el choque En lo referente a los niveles en que se puede producir el choque eso dependerá de lo cerca que estén en el sentido angular los subárboles mencionados en el punto anterior. Ver figura (1.18).

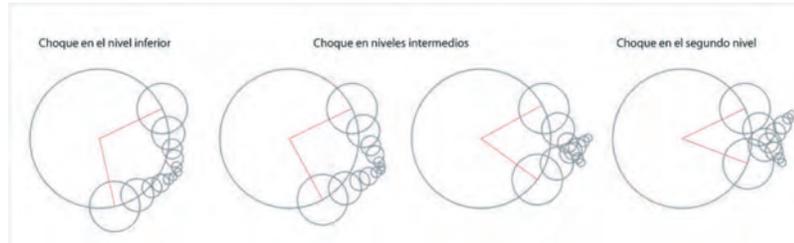


Figura 1.18 Análisis de los ángulos entre los brazos en posible colisión dependiendo del nivel.

1.4.3.2 Posición del observador Sin embargo, la posición del observador no estará nunca de frente al choque, pues el BTF se trata precisamente de traer un nodo al ojo del usuario y como se vio en la sección anterior, el choque solo se produce cuando se saca un nodo que esté a 90 grados de la línea que forman los descendientes. Esto da como resultado que inevitablemente el choque en estas condiciones se da "de lado" con respecto al usuario. Ver figura (1.19)

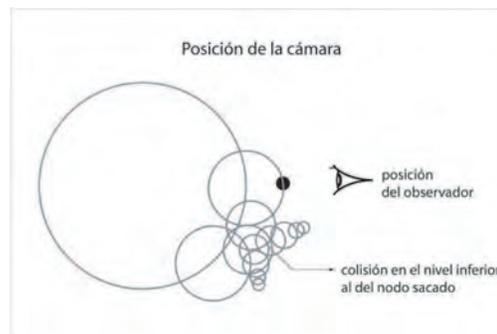


Figura 1.19 Análisis de la posición del observador en un posible BTF

Esto también concluye que si es posible un choque en el siguiente nivel al del nodo sacado y por lo tanto al del observador, para que se dé se deben cumplir las siguientes situaciones:

1. Dos nodos deben tener mucha descendencia
2. Los nodos deben de estar uno al lado del otro

3. La descendencia debe de estar alineada previamente en un solo sentido
4. Debe sacarse un nodo a 90 grados del subárbol vecino en el primer subárbol
5. Debe sacarse un nodo a 90 en el sentido contrario al paso anterior del otro subárbol

1.4.3.3 Alineación de los subárboles a través del BTF Los dos apartados anteriores hablan de situaciones en que los descendientes de un nodo están alineados, en este caso la pregunta es ¿qué tan posible es que esto suceda? Esta situación es improbable en la configuración inicial del árbol pues las distribuciones, como hemos visto, dependen de la cantidad de hijos por generación, así las cosas, la única posibilidad de alinear los descendientes (que no sea hacerlo adrede) sería que esta condición fuera un efecto colateral de uso intenso de los BTF. Para alinear los subárboles solo usando el BTF, es necesario "sacar" el nodo a alinear en el ángulo que están haciendo su padre y abuelo, es decir deben estar en esa posición o ponerlos con anterioridad así. De este modo, el trío de nodos queda alineado. Ver figura (1.20).

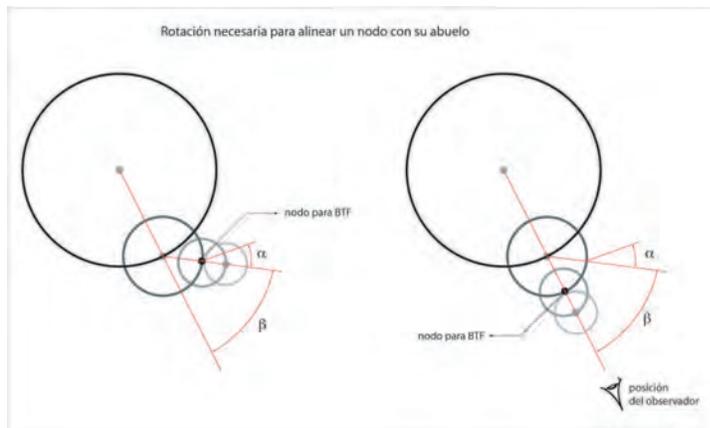


Figura 1.20 Análisis de subárboles a través de varios BTF

Esta situación debe repetirse con cada nodo hacia la raíz hasta llegar a ésta para tener una generación alineada, o estas circunstancias deben darse en forma consecutiva casualmente. Ver (1.21)

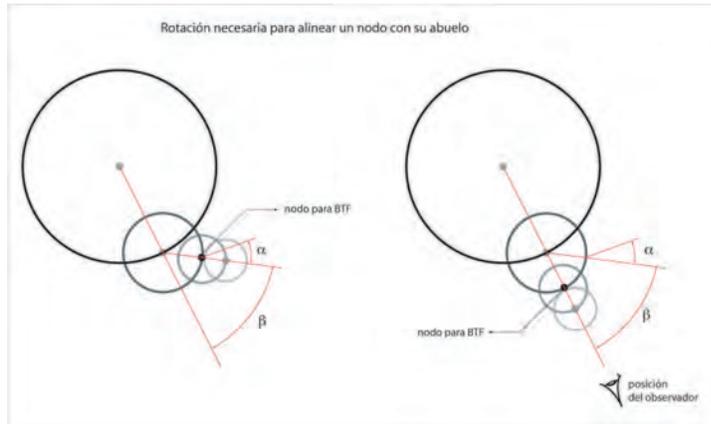


Figura 1.21 Posiciones necesarias para que con un BTF un nodo se alinee con su abuelo.

1.5 Conclusiones y recomendaciones:

Como conclusiones de todo este trabajo se puede resumir lo siguiente:

1. Se logró optimizar los algoritmos de generación y posicionamiento de los conos con el fin de reducir la posibilidad de choques en el momento de su generación.
2. Los árboles con ninguna posibilidad de choques son inviables debido al uso muy ineficiente del espacio disponible para la visualización.
3. Se encontró que la posibilidad de que se de un choque en la ejecución de BTFs requiere de una secuencia ordenada de acontecimientos que se estiman poco probables.
4. En caso de que se dé choques ocasionados por secuencias de BTFs el choque se daría en un nivel inferior al plano del observador.

Bibliografía

- [1] Auber, D. *Tulip A Huge Graph Visualization Framework*. in Junger, M. Mutzel, P. Graph (2004) *Drawing Software*. Berlin: Springer, Pp:105-112
- [2] R. Boardman. *Bubble Trees. The Visualization of Hierarchical Information Structures*. Conference on Human Factors in Computing Systems. CHI '00 Imperial College. 2000
- [3] T. Bladh, D. Carr, and J. Scholl. "Extending Tree-Maps to Tree Dimesions: A Comparative Study". Springer Verlag, 2004
- [4] C, Jeon and A. Pang. *Interacting with Huge Hierarchies: Beyond Cone Trees*. Obtenido el 4 de marzo del 2007 de la base de datos de la IEEE Xplore.
- [5] Y. Carriere, R. Kazman. *Reconfigurable Disc Trees for Visualizing Large Hierarchical Information Space*. Obtenido el 4 de marzo del 2007 de la base de datos de la IEEE Xplore.
- [6] S. Card, J.D. Mackinlay, B. Schneiderman. "Readings in Information Visualization. Using Vision to Think". Morgan Kaufmann Publishers. San Diego, CA. 1999.
- [7] R. Becker, S. Eick, A. Wilks. *Visualizing Network Data*.
- [8] Fairchild, K. M., Poltrock, S. E., and Furnas, G. W. *SemNet. Three-Dimensional Representations of Large Knowledge Bases*. In R. Guindon (ed.) *Cognitive Science and its applications for human-computer interaction*. Hillsdale, NJ. 1988.
- [9] Hernández, F. Monge, J. *Diagnóstico General de Visualización de Estructuras Jerárquicas*, Tiempo Compartido, Vol. 7, No. 3, Agosto, 2007
- [10] Hernández, F. Monge, J. *Visualización tridimensional de estructuras jerárquicas*. Tesis de Maestría en Ciencias de la Computación. Escuela de Ingeniería en Computación, Instituto Tecnológico de Costa Rica. 2006
- [11] B. Johnson, B. Shneiderman. *Tree-Maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures*. Proceeding of IEEE Visualization'91 Conference. San Diego, 1991.
- [12] H. Koike and H. Yoshihara. *Fractal approaches for visualizing huge hierarchies*. In Proceedings of 1993 IEEE/CS Symposium on Visual Languages (VL'93), pages 55-60. IEEE CS Press, 1993.
- [13] Mackinlay J. D., Card S. K., Robertson G.G. *Cone Trees: Animated 3D Visualizations of Hierarchical Information*, UIR-R-1995
- [14] T. Munzner. *Laying Out Large Directed Graphs in 3D Hyperbolic Space*. Stanford University. IEEE Symposium on Information Visualization, Phoenix .October 20-21 1997,

- [15] T. Munzner, P. Burchard. *Visualizing the Structure of the World Wide Web in 3D Hyperbolic Space*. Proceedings of special issue of Computer Graphics, ACM SIGGRAPH, New York, 1995, pp. 33-38
- [16] J. D. Mackinlay, S. K. Card, G.G. Robertson. *Cone Trees: Animated 3D Visualizations of Hierarchical Information*. UIR-R-1995
- [17] P. Zhang, S. Shekhar, V. Kumar. "Spacial Cone Tree: An Index Structure for Correlation-based Similarity on Special Time Series Data". University of North Texas. 2003.
- [18] Ruby M. K. "Information Visualization for Financial Analysis". Tesis de Maestría en Ciencias de la Computación. Department of Computer Science, University of Durham. 2003
- [19] J. Wijk, H. Wetering. "Cushion Treemaps: Visualization of Hierarchical Information". Eindhoven University of Technology. 1999.
- [20] F. Yidoung, D. Shihai. "Network Information Visualization Based on Multi-Resolution Animated Cone Trees". Peking University. 2001