

[Informe final de proyecto]

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

Investigadores Ph. D. Erick Mata, M.Sc. Jorge Monge y Ph. D. Franklin Hernández-Castro

Tabla de contenidos

| | |
|---|----|
| Resumen | 3 |
| Lista de figuras | 3 |
| Introducción | 4 |
| Antecedentes: | 4 |
| El ordenamiento y dimensionado de los conos | 4 |
| Cálculo de ángulos y posicionamiento de los nodos u hojas | 9 |
| Definición del problema: | 12 |
| Revisión de literatura: | 12 |
| Artículo de Kazman/Carriere [CK95]: | 12 |
| Artículo de Chang-Sung/Pang [CHSP98]: | 13 |
| Artículo de David Auber [A04]: | 14 |
| Materiales y métodos: | 15 |
| Resultados y discusión: | 16 |
| 1. Definición del dominio del problema: | 16 |
| 2. Escenarios de generación de choques: | 16 |
| 2.1. Posibilidad de choques en la generación inicial de los árboles: | 17 |
| Observaciones: | 19 |
| 2.1.1 Definición de los árboles sin posibilidad de choques | 26 |
| 2.2 Posibilidad de choques en la ejecución de una secuencia de rotación sincrónica (conocida como “Bring to Front” o BTF) | 27 |
| 2.2.1 Niveles en los que se puede producir el choque | 27 |
| 2.2.2 Posición del observador | 28 |
| 2.2.3. Alineación de los subárboles a través del BTF | 29 |
| Conclusiones y recomendaciones: | 31 |
| Aportes y alcances: | 31 |
| Bibliografía | 31 |
| | |
| DOCUMENTO II | 33 |
| Cumplimiento de objetivos: | 33 |
| Objetivos específicos: | 33 |
| Limitaciones y problemas encontrados: | 33 |
| Observaciones generales y recomendaciones: | 33 |

Resumen

El presente proyecto estudió con detalle los algoritmos de posicionamiento, dimensionado y rotación dinámica de conos actualmente conocidos en árboles de conos. No estuvo orientado a programar, sino a refinar los algoritmos, entender su entorno y mejorar su eficiencia desde el punto de vista de tiempo de ejecución.

Lista de figuras

| | |
|---|----|
| Figura 1 Definición de los parámetros básicos | 5 |
| Figura 2. Problemas de traslape de conos en los niveles inferiores | 6 |
| Figura 3 Solución a los problemas de traslape de conos en los niveles inferiores | 7 |
| Figura 4 Problemas de traslape de conos en los niveles inferiores | 7 |
| Figura 5 Nuevo alineamiento de los conos | 8 |
| Figura 6 Proyección del nodo j sobre el plano XZ..... | 10 |
| Figura 7 Trigonometría para determinar el vector de posición del r-ésimo hijo..... | 11 |
| Figura 8. Visualización del problema..... | 12 |
| Figura 9 Estrategia base publicado por Kazman y Carriere en 1995 | 12 |
| Figura 10 Situación en la que un hijo puede resultar con un diámetro mayor al de su padre. | 13 |
| Figura 11 Caso de choques siguiendo la estrategia Kazman | 13 |
| Figura 12 Corrección Chang-Sung/Pang..... | 14 |
| Figura 13 Problemas de choques en la estrategia Chang-Sung/Pang | 14 |
| Figura 14 Estrategia de David Auber 2004. | 15 |
| Figura 14. Visualización del dominio del problema. | 16 |
| Figura 15. Taxonomía de construcción en los árboles de conos..... | 17 |
| Figura 16. Taxonomía general de construcción en los árboles de conos..... | 18 |
| Figura 17. Visualización del problema..... | 20 |
| Tabla 1. Prueba de Kolmogorov-Smirnov para una muestra | 21 |
| Tabla 2. Estadísticas de choques | 21 |
| Figura 18. | 22 |
| Tabla 3. Prueba de Kolmogorov-Smirnov para una muestra | 23 |
| Tabla 4 Comparación de choques entre algoritmos..... | 24 |
| Figura 19. Comparación grafica de los choques..... | 24 |
| Figura 20. Estructura desplegada con el algoritmo normal | 25 |
| Figura 21. Estructura desplegada con el algoritmo refinado..... | 25 |
| Figura 22. Estructura desplegada con el algoritmo normal | 26 |
| Figura23. Estructura desplegada con el algoritmo refinado..... | 26 |
| Figura 24. Condiciones para la generación de un choque durante una secuencia de BTFs. | 27 |
| Figura 25. Análisis de los ángulos entre los brazos en posible colisión dependiendo del nivel. | 28 |
| Figura 26. Análisis de la posición del observador en un posible BTF | 28 |
| Figura 27. Análisis de subárboles a través de varios BTF | 29 |
| Figura 28. Posiciones necesarias para que con un BTF un nodo se alinee con su abuelo..... | 30 |

Introducción

Antecedentes:

Como resultado del proyecto de investigación **Visualización tridimensional de estructuras jerárquicas**, nace la necesidad de un mejor algoritmo usado en el dimensionado y el posicionamiento de los conos.

Como se recordará, el proyecto citado dio como resultado una aplicación que visualiza taxonomías de las especies de Costa Rica. El proyecto usa el paradigma de árboles de conos para desplegar los diferentes niveles de información.

El ordenamiento y dimensionado de los conos

El paradigma de árboles de conos se caracteriza por que los nodos se ubican en la parte superior de un cono y sus hijos se distribuyen sobre la base del mismo. El cono es un elemento que se utiliza como indicador de pertenencia de las hojas (hijos) al nodo(padre), es decir establece la relación jerárquica entre los datos.

Este paradigma es similar al de representar las relaciones jerárquicas por medio de árboles, solo que en nuestro caso se trabaja en el espacio tridimensional y, en lugar de utilizar líneas para establecer las relaciones, utilizamos conos. La utilización de conos y trabajar en un espacio tridimensional implican ciertos retos en cuanto a la distribución de los conos en el espacio, la distribución de las hojas sobre la base del cono, la altura de los conos, la ubicación y la orientación exacta de las hojas sobre la base de los conos.

En nuestro caso empezamos por definir la altura de los conos h como constante en todos los niveles del árbol de conos. El algoritmo que distribuye y establece la ubicación de los nodos y hojas no solamente debe buscar la optimización del espacio requerido sino que debe ser estéticamente aceptable. En primera instancia el algoritmo que utilizamos se basó en estudios hechos por Matthew K. Ruby de la Universidad de Durham [R03], en los cuales se establece un algoritmo *botton-top* donde el radio de la base del cono depende de la de los radios de sus hijos (hojas), el radio de éstos a su vez, depende de sus propios hijos y así sucesivamente. Con esta estrategia se trata de que el nodo raíz cubra (o proteja) a sus hijos y ellos a los suyos.

La siguiente figura 1 sugiere la estrategia inicial sobre la cual se basa el algoritmo.

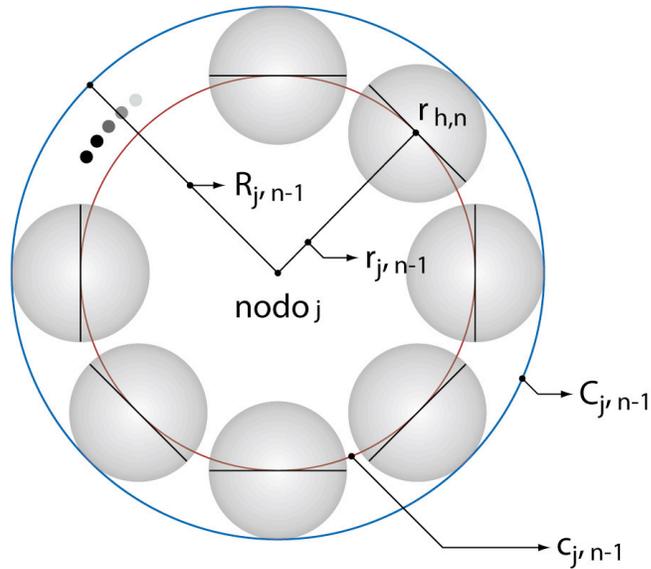


Figura 1 Definición de los parámetros básicos

Primero para las hojas disponemos de un diámetro por omisión, el cual lo definimos al inicio como $D_{h,n}$ (en realidad para cualquier n), de esta forma las hojas en el nivel n están asociadas a una

circunferencia $C_{h,n} = D_{h,n}$ de radio $r_{h,n}$ donde $r_{h,n} = \frac{C_{h,n}}{2\pi}$.

Si definimos j como el j -ésimo nodo del nivel $n-1$ y k como el número de hijos (en este caso todas las hojas) del nodo j . La circunferencia de la base del cono (como sugiere la figura 4.16) en el nivel $n-1$ del nodo j -ésimo viene dado por

$$C_{j,n-1} = \sum_{i=1}^k D_{H,n} \text{ aproximadamente, su diámetro por}$$

$$D_{j,n-1} = \frac{C_{j,n-1}}{\pi} \text{ y su radio por}$$

$$r_{j,n-1} = \frac{D_{j,n-1}}{2} = \frac{\frac{C_{j,n-1}}{\pi}}{2} = \frac{C_{j,n-1}}{2\pi}$$

$$r_{j,n-1} = \frac{\sum_{i=1}^k D_{H,n}}{2\pi} = \frac{1}{2\pi} \sum_{i=1}^k D_{H,n}$$

así se tiene que

$$r_{j,n-1} = \frac{1}{2\pi} \sum_{i=1}^k D_{H,n}$$

Sin embargo esta estrategia no evita el traslape de conos en niveles inferiores. Este problema se evidencia en la figura 2.

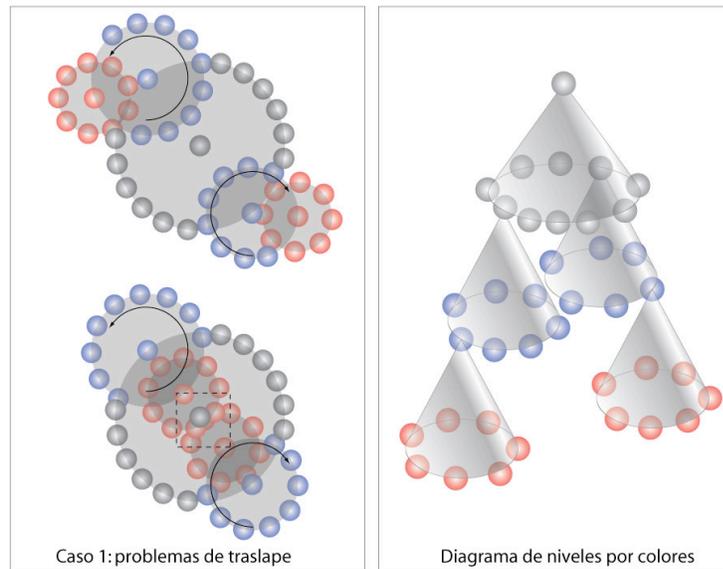


Figura 2. Problemas de traslape de conos en los niveles inferiores

En este caso, todas las bases de los conos están calculados como se indicó para el nivel $n - 1$, en principio no hay un traslape evidente, sin embargo cuando se rotan los conos en el tercer nivel se produce un traslape. Esta problemática se acentúa aún más cuando la cantidad de hijos en un nodo es asimétrica respecto a los otros.

Para resolver este problema se ajustó el procedimiento (figura 3) para un nodo cualquiera j en el nivel m con k hijos, en el caso en que $m = n - 1$ (como se muestra en la figura 1) se redefine un nuevo diámetro $D_{j,n-1}$, donde

$$D_{j,n-1} = 2(r_{j,n-1} + r_{h,n})$$

y en consecuencia una nueva circunferencia

$$C_{j,n-1} = \pi D_{j,n-1}$$

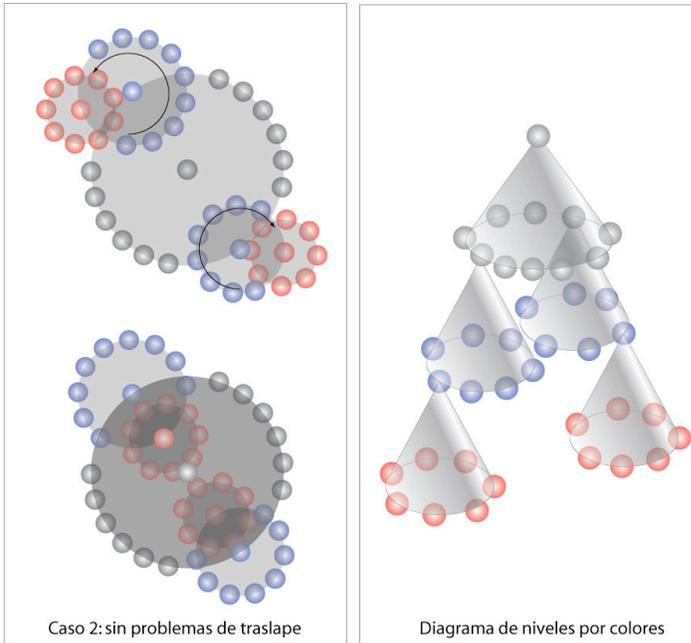


Figura 3 Solución a los problemas de traslape de conos en los niveles inferiores

Generalizando esta estrategia, donde un nodo j tiene hijos (nodos u hojas) y estos a su vez con un número diferente de hijos como muestra la figura 4.

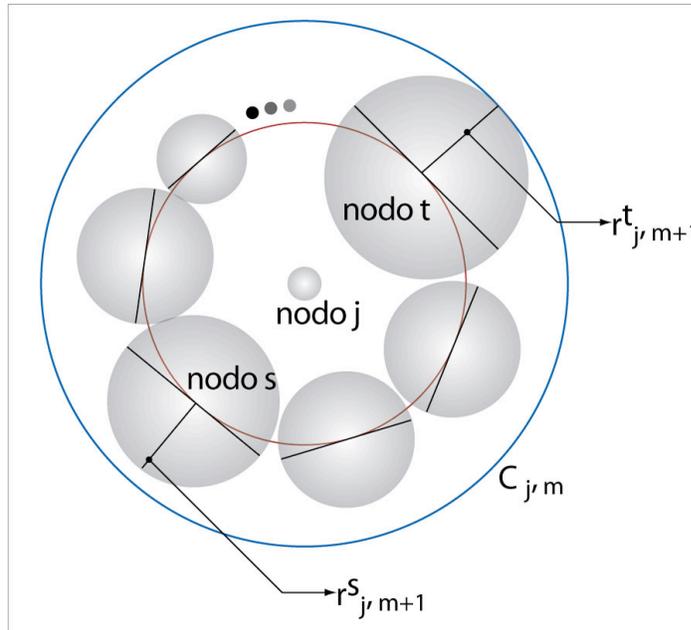


Figura 4 Problemas de traslape de conos en los niveles inferiores

Definimos el radio máximo del nodo j como

$$r_{\max} = \max\{r^1_{j,m+1}, r^2_{j,m+1}, r^3_{j,m+1}, \dots, r^k_{j,m+1}\}$$

y $r^t_{j,m+1}$ representa el radio del hijo t -ésimo del nodo j en el nivel m y $r^s_{j,m+1}$ representa el radio del hijo s -ésimo del nodo j en el nivel m .

Dado que el cono "cubre" a sus hijos y los hijos a su vez cubren a sus propios hijos, primero se calcula el radio $r_{j,m}$ como el caso en que $m = n - 1$, es decir

$$r_{j,m} = \frac{1}{2\pi} \sum_{i=1}^k D^i_{j,m+1}$$

y si $r_{j,m} < r_{\max}$ entonces $r_{j,m} = r_{\max}$.

Además definimos $D_{j,m} = 2(r_{j,m} + r_{\max})$ como el diámetro de la base del cono del nodo j . La suma de los $D^i_{j,m+1}$ viene a ser la circunferencia de la base del cono, esto es

$$C_{j,m} = \sum_{i=1}^k D^i_{j,m+1}$$

entonces significa que estamos guardando una circunferencia mayor que la obtenida con la simple suma de los diámetros de sus hijos.

Bajo estas condiciones el problema propuesto en la figura 2 se resuelve parcialmente, esto por cuanto los conos quedan tangentes después de alguna rotación. La solución se visualiza como el caso 2 mostrado en la figura 3, más exactamente descrito en la figura 4.

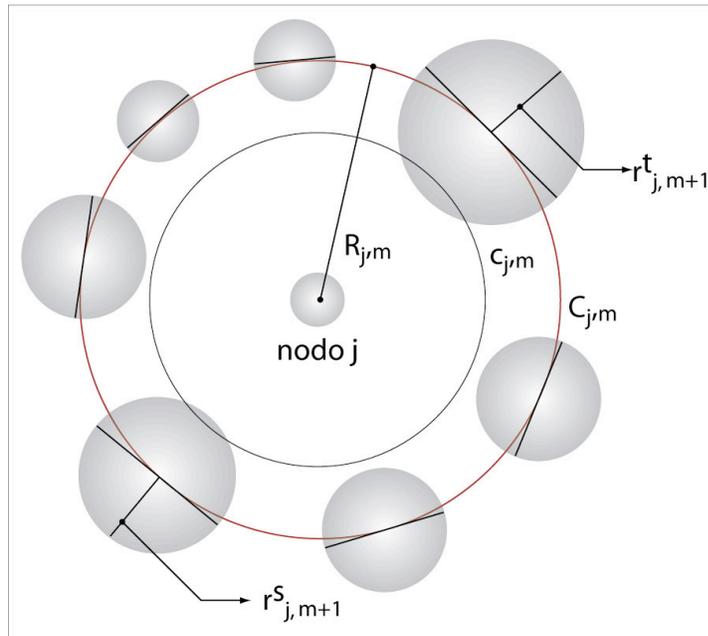


Figura 5 Nuevo alineamiento de los conos

En la figura anterior (5), se muestra la ubicación de los nodos en la nueva circunferencia y el espacio disponible para la rotación de sus hijos evitando el traslape.

Dado que los conos asociados a nodos hermanos consecutivos bajo estas condiciones tienden a ser tangentes, consideramos necesario insertar un factor de tolerancia t , para este efecto el diámetro cambia a

$$D_{j,n-2} = D_{j,n-2} + t$$

permitiendo un espacio mayor entre conos asociados a nodos hermanos.

Con la altura de los conos ya fijada y la base de los conos calculada con el procedimiento descrito anteriormente nos queda por definir la forma en que se posicionarán los hijos de un nodo cualquiera sobre la base del cono.

Cálculo de ángulos y posicionamiento de los nodos u hojas

Dado que ya tenemos la circunferencia de la base de cada cono, el problema que nos queda es distribuir los nodos u hojas para posicionarlos a lo largo de la circunferencia en el espacio adecuado, partiendo del nodo raíz. En este caso el nodo raíz se ubica en el origen del sistema

$$O = (0,0,0).$$

Para esto debemos tener en cuenta que la altura (h) de los conos es constante por lo que la componente en y decrece o crece como función de la altura del cono. Para ello consideremos el nodo j -ésimo en el nivel m con k hijos, determinaremos el vector posición del r -ésimo hijo(en nivel $m + 1$) del nodo j .

Sea $P = (x_{j,m}, y_{j,m}, z_{j,m})$ la posición del nodo j y sea $Q = (x^r_{j,m+1}, y^r_{j,m+1}, z^r_{j,m+1})$ la posición a determinar del r -ésimo hijo. La figura (4.21) muestra la proyección del nodo j sobre el plano XZ y la proyección $P' = (x_{j,m}, z_{j,m})$ de P y $Q' = (x^r_{j,m+1}, z^r_{j,m+1})$ de Q .

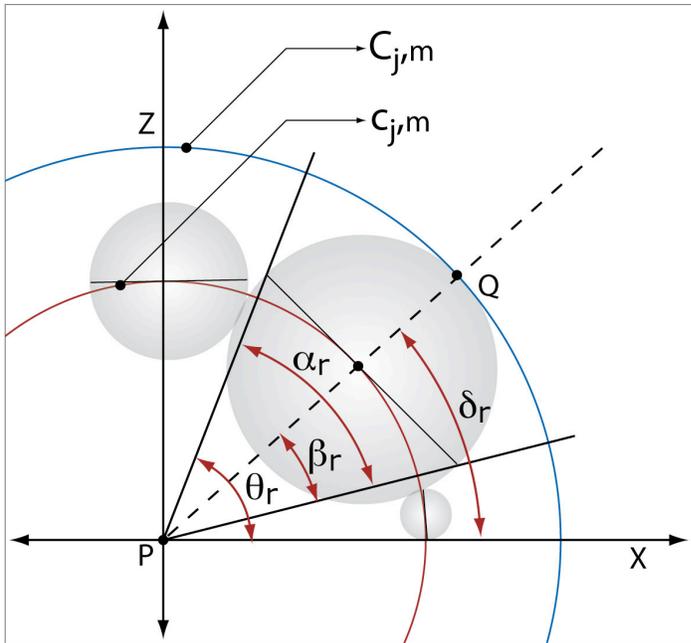


Figura 6 Proyección del nodo j sobre el plano XZ

Lo primero es determinar el ángulo α (que llamaremos ángulo real, siendo el ángulo que abarca el diámetro del nodo asociado), para ello se define α_r como sigue

$$\alpha_r = \frac{D^r_{j,n-1}}{c_{j,n-1}} * 360$$

posteriormente calculamos β_r (que llamaremos ángulo medio) este viene dado por

$$\beta_r = \frac{\alpha_r}{2}$$

y definimos el ángulo θ_r , como el ángulo medido desde el eje X hasta el rayo final del ángulo α_r en sentido anti-horario, este viene dado por

$$\theta_r = \sum_{i=1}^{r-1} \alpha_i$$

Por último de finimos el ángulo δ_r como el ángulo medido desde el eje X hasta el rayo $\overrightarrow{P'Q'}$ y este viene dado por

$$\delta_r = \theta_r + \beta_r$$

Para determinar el vector posición del r-ésimo hijo utilizamos trigonometría elemental como se muestra en la figura 7.

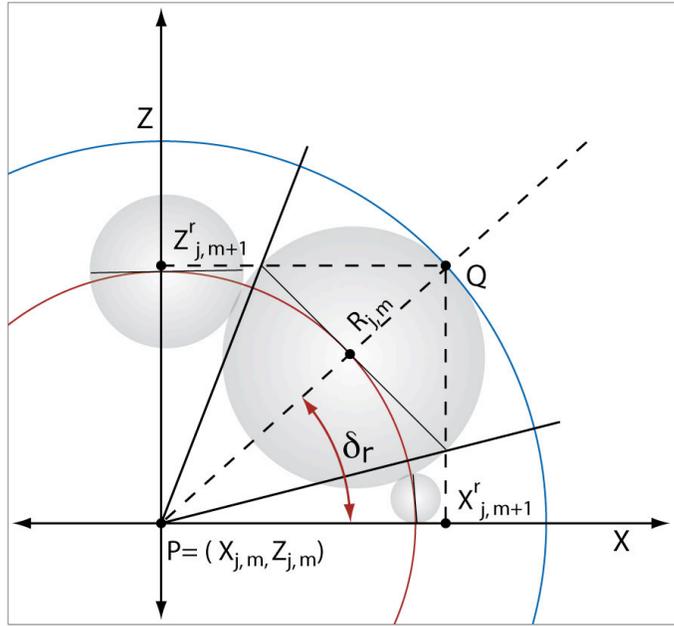


Figura 7 Trigonometría para determinar el vector de posición del r-ésimo hijo

de donde se concluye que

$$\cos(\delta_r) = \frac{x_{j,m+1}^r - x_{j,m}}{R_{j,m}} \quad \text{y} \quad \text{sen}(\delta_r) = \frac{z_{j,m+1}^r - z_{j,m}}{R_{j,m}}$$

concluyendo que las componentes $x_{j,m+1}^r$ y $z_{j,m+1}^r$ vienen dadas por

$$x_{j,m+1}^r = R_{j,m} * \cos(\delta_r) + x_{j,m}$$

$$z_{j,m+1}^r = R_{j,m} * \text{sen}(\delta_r) + z_{j,m}$$

y la posición del r-ésimo en función a la de su padre viene dada por $Q = (x_{j,m+1}^r, y_{j,m+1}^r, z_{j,m+1}^r)$ donde

$$x_{j,m+1}^r = R_{j,m} * \cos(\delta_r) + x_{j,m}$$

$$y_{j,m+1}^r = y_{j,m} - h$$

$$z_{j,m+1}^r = R_{j,m} * \text{sen}(\delta_r) + z_{j,m}$$

Esto nos permite posicionar al hijo Γ del nodo j .

Por último como tanto las hojas como los nodos están representados por planos (en los que se "texturean" las ilustraciones de las especies), lo ideal es que estos sean tangentes a la base del cono, para esto se define el ángulo de rotación γ_r sobre el eje Y para el r-ésimo hijo como $\gamma_r = 90 - \delta_r$.

Definición del problema:

Refinamiento de los algoritmos de dimensionado de los conos así como el posicionamiento radial de los nodos y rotación dinámica de los conos en un árbol de representación de información jerarquizada.

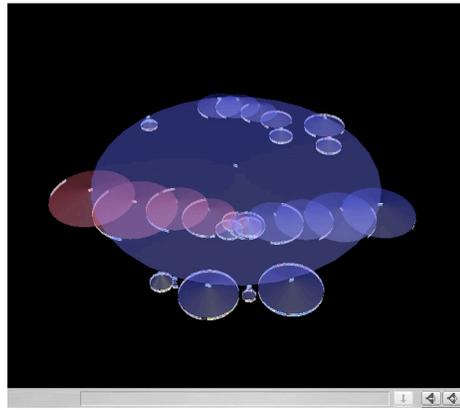


Figura 8. Visualización del problema.

Revisión de literatura:

Además de la literatura citada en este informe encontramos de especial relevancia, estos artículo:

Artículo de Kazman/Carriere [CK95]:

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. Ver figura 9.

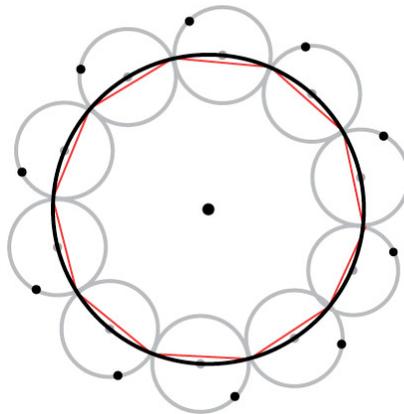


Figura 9 Estrategia base publicado por Kazman y Carriere en 1995

Con este algoritmo aun 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 solo hijo que a su vez tiene varios, el padre resultará con un cono de menor tamaño que su hijo. Ver figura 10.

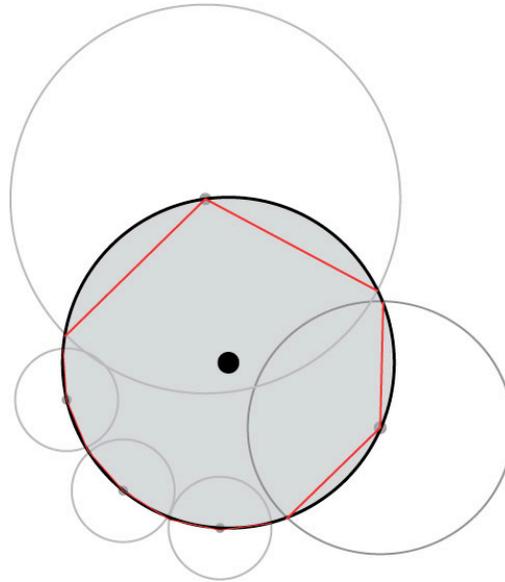


Figura 10 Situación en la que un hijo puede resultar con un diámetro mayor al de su padre.

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 que lo necesario en casos muy heterogéneos. Ver figura 11.

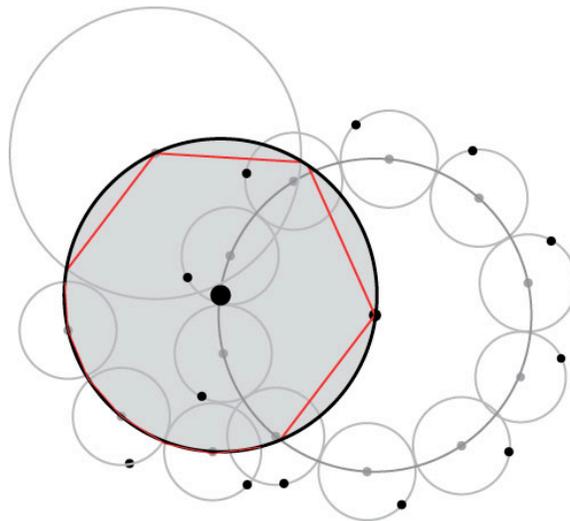


Figura 11 Caso de choques siguiendo la estrategia Kazman

Artículo de Chang-Sung/Pang [CHSP98]:

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 12.

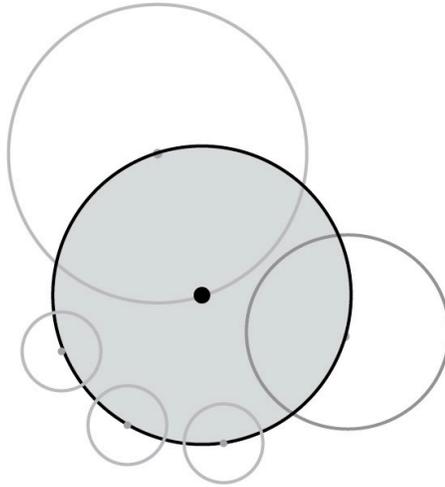


Figura 12 Corrección Chang-Sung/Pang

Sin embargo, no garantiza 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 13.

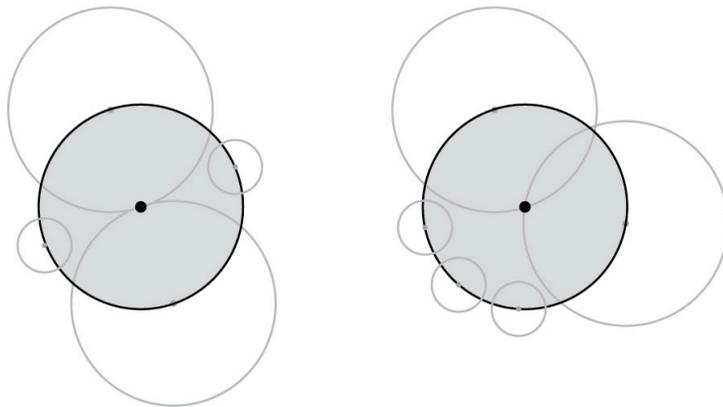


Figura 13 Problemas de choques en la estrategia Chang-Sung/Pang

Artículo de David Auber [A04]:

Varios años después, para 2004, encontramos una estrategia basada en el círculo que inscribe a todos 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 14.

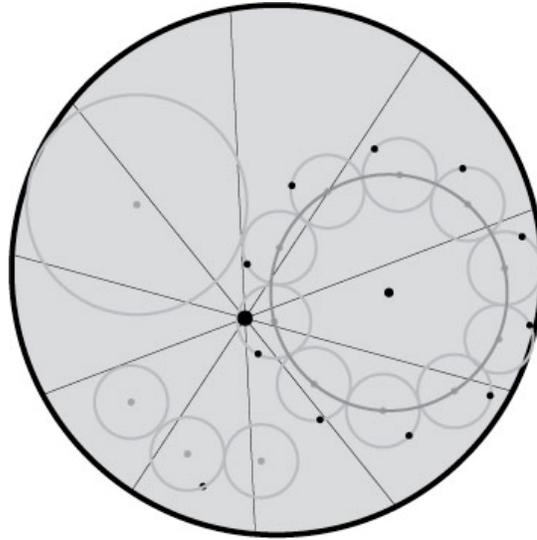


Figura 14 Estrategia de David Auber 2004.

Materiales y métodos:

Se analizó la eficiencia y precisión de los algoritmos actuales, tanto los propios como los descritos en la literatura. Posteriormente se formalizaron los conceptos y se desarrollaron modificaciones a los algoritmos actuales para lograr los objetivos específicos del proyecto. Veamos el detalle.

Resultados y discusión:

1. Definición del dominio del problema:

Como primera actividad para acercarse al análisis del problema se decidió especificar formalmente el dominio del problema.

En el universo de los árboles de conos podríamos comenzar por definir una escala de densidad en la que en un extremo estarían alineados los árboles de conos menos densos (ralos) y en el otro extremo posicionaríamos los árboles más densos. Ver figura 2.

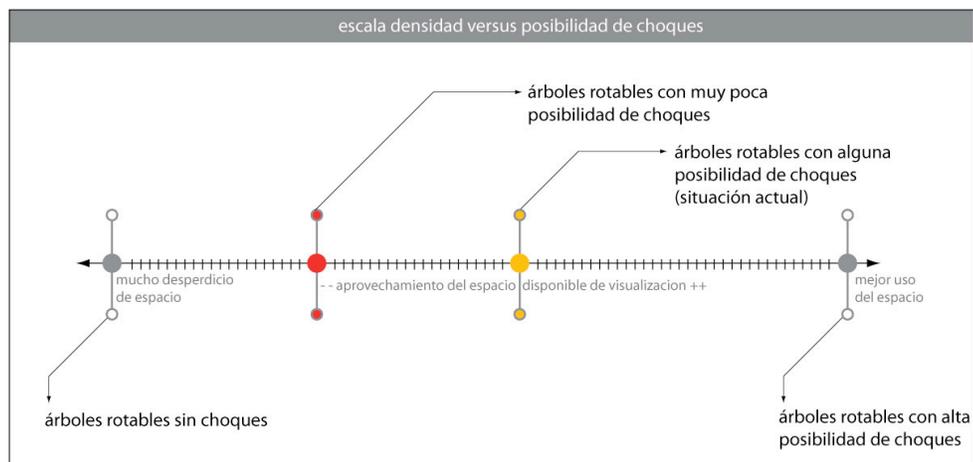


Figura 14. Visualización del dominio del problema.

Obviamente la escala también grafica una métrica de aprovechamiento del espacio disponible, pues entre más densos sean los árboles mejor se utilizará el espacio. Cabe aclarar que "mejor" en este sentido, solo toma en cuenta la cantidad de información por unidad de espacio.

Además queda claro con la definición de este dominio que la probabilidad de colisiones entre subárboles aumenta conforme se avanza hacia una mayor densidad en la escala. En este sentido los algoritmos desarrollados anteriormente se sitúan en esta escala en la zona central, como se ve en la figura.

2. Escenarios de generación de choques:

Otro aspecto a analizar sería la definición de las circunstancias en que se pueden dar los choques entre subárboles. Después de un análisis minucioso se definió que las dos circunstancias en las que se pueden provocar colisiones entre subárboles son:

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

2.1. Posibilidad de choques en la generación inicial de los árboles:

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 redimensionado y posicionamiento dentro del biovisualizador, para esto consideramos la figura 15.

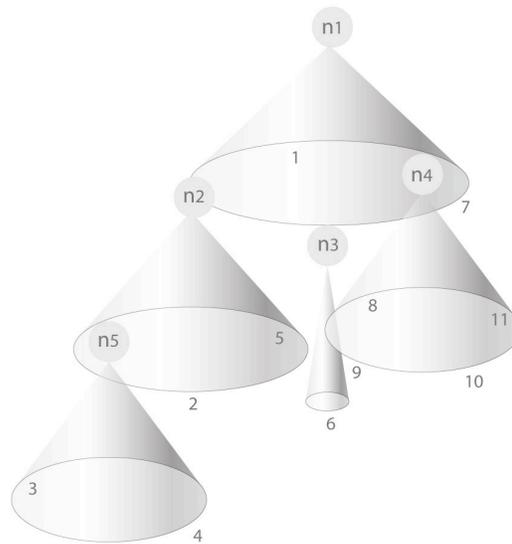
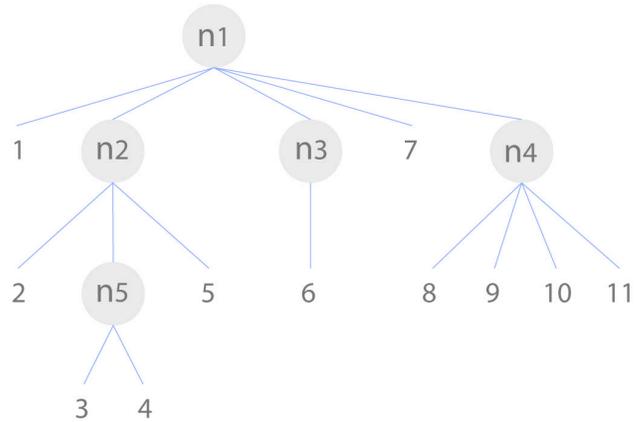


Figura 15. Taxonomía de construcción en los árboles de conos

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. En un árbol de conos esta estructura se representa como muestra la figura 15.

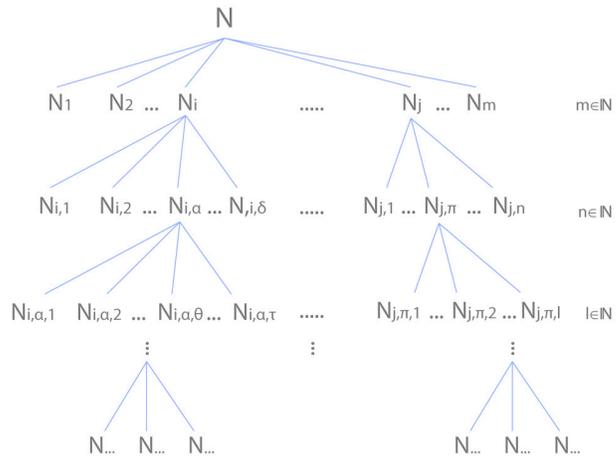


Figura 16. Taxonomía general de construcción en los árboles de conos

En esta estructura N corresponde al nodo padre, sin embargo 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 16:

1. 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. Si la distancia entre i y j es pequeña, por lo que conforme $j \rightarrow i + 1$ el traslape ocurrirá en niveles inferiores.
3. 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 inferiores.
4. Para el subárbol con raíz N_i se espera que $s \rightarrow \alpha + 1, s \rightarrow \alpha + 1$ además de mantener la asimetría en el número de hijos de $N_{i,\alpha}$.
5. Para el subárbol con raíz N_j se espera que $s \rightarrow \alpha + 1, s \rightarrow \alpha + 1$ además de mantener la asimetría en el número de hijos de $N_{j,\alpha}$.
6. Bajo las condiciones impuestas a los nodos N_i y N_j en el punto 4 y 5 se produce traslape en la construcción del árbol de conos. El nivel donde ocurren dependerá de $s \rightarrow \alpha + 1, s \rightarrow \alpha + 1$.

Observaciones:

Si no se produjeran 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 16.

El traslape planteado tanto en la construcción como la que se obtiene por la aplicaciones en 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 16 se tiene que por el algoritmo convencional definido por Carriere y Kazman (1996) la longitud de la circunferencia y del radio vendrían dadas por $D_{j,m} = 2 \cdot (r_{j,m} + r_{\max})$ y $R_{j,m} = r_{j,m} + r_{\max}$, 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 $D_{j,m} = 2 \cdot r_{j,m}$.

La figura (5) 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. 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

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

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 17.

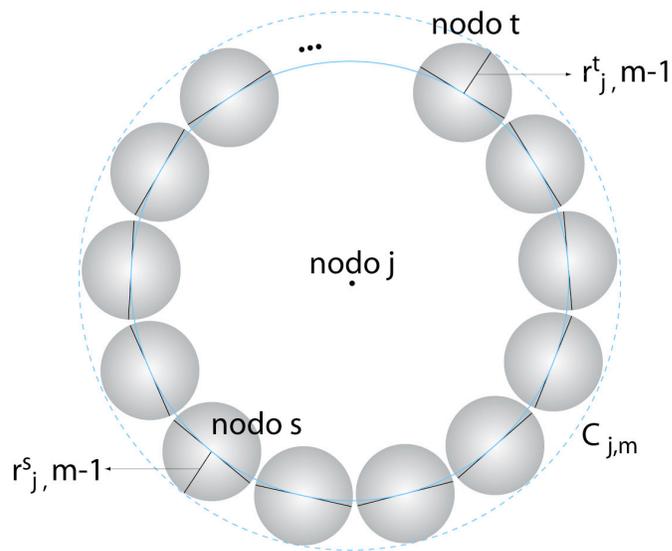


Figura 17. Visualización del problema

En el algoritmo se calcula este cociente y si este valor llegase a ser menor al 35%, se establece la siguiente corrección experimental en el algoritmo $D_{j,m} = 2 \cdot (r_{j,m} + 5 \cdot r_{\max})$, permitiendo disminuir considerablemente la posibilidad de choques.

Las caracterizaciones sobre la estructura permiten únicamente tener un escenario con posibilidad de choque, 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 programar 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 conforme 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 Macbook. 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. 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 el siguiente resultado para árboles de nivel no mayor a 3, tanto para el algoritmo normal como al algoritmo refinado:

Tabla 1. Prueba de Kolmogorov-Smirnov para una muestra

| Algoritmo | | | NumNodos |
|-------------|---------------------------|-------------------|----------|
| Sin refinar | N | | 50 |
| | Parámetros normales(a,b) | Media | 277,16 |
| | | Desviación típica | 187,760 |
| | Diferencias más extremas | Absoluta | ,106 |
| | | Positiva | ,106 |
| | | Negativa | -,071 |
| | Z de Kolmogorov-Smirnov | | ,749 |
| | Sig. asintót. (bilateral) | | ,629 |
| Refinado | N | | 50 |
| | Parámetros normales(a,b) | Media | 255,78 |
| | | Desviación típica | 185,353 |
| | Diferencias más extremas | Absoluta | ,169 |
| | | Positiva | ,169 |
| | | Negativa | -,085 |
| | Z de Kolmogorov-Smirnov | | 1,194 |
| | Sig. asintót. (bilateral) | | ,115 |

a La distribución de contraste es la Normal.

b Se han calculado a partir de los datos.

Ambos valores son claramente superiores al nivel de significancia (5%) por lo que la distribución del número de nodos para árboles con nivel no mayor a 3 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:

Tabla 2. Estadísticas de choques

| Algoritmo | | Frecuencia | Porcentaje | Porcentaje válido | Porcentaje acumulado |
|-------------|----------------|------------|------------|-------------------|----------------------|
| Sin refinar | No hubo choque | 35 | 70,0 | 70,0 | 70,0 |
| | Hubo choque | 15 | 30,0 | 30,0 | 100,0 |
| | Total | 50 | 100,0 | 100,0 | |
| Refinado | No hubo choque | 44 | 88,0 | 88,0 | 88,0 |
| | Hubo choque | 6 | 12,0 | 12,0 | 100,0 |
| | Total | 50 | 100,0 | 100,0 | |

Gráficamente podemos visualizar la relación existente como se muestra en los gráficos Ver figura 18

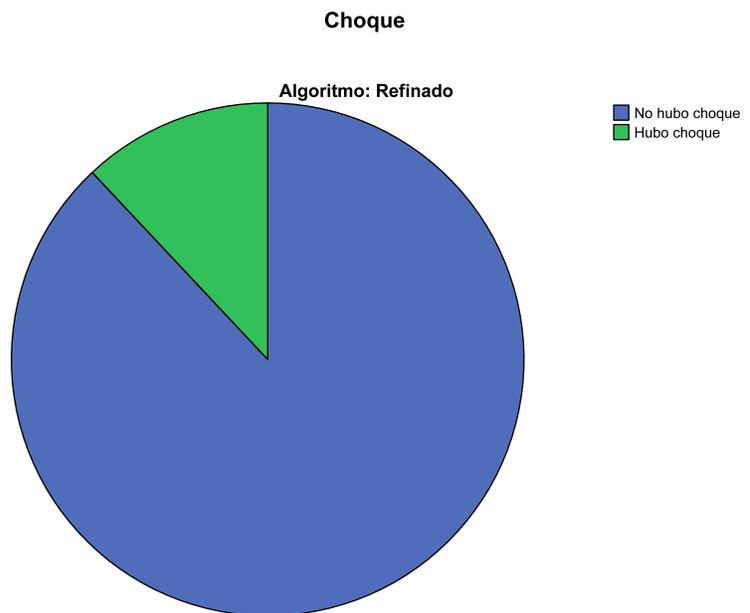
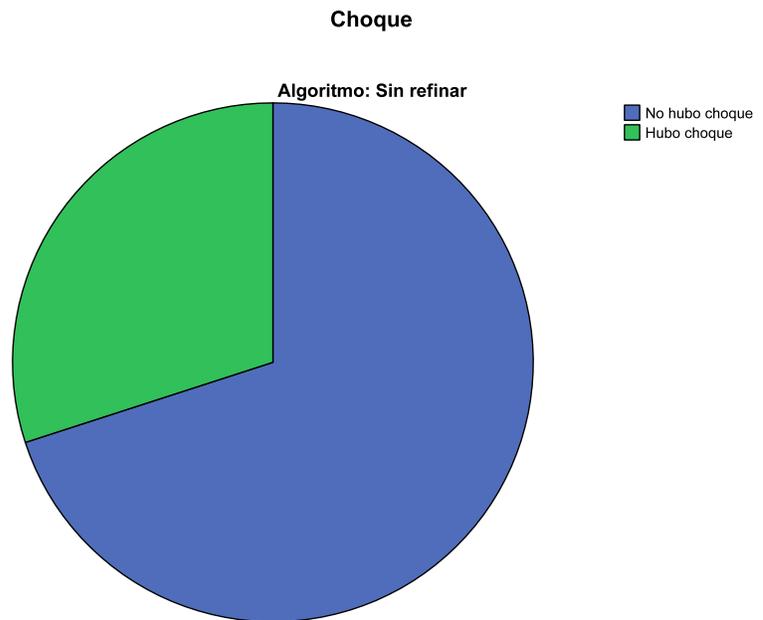


Figura 18.

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 el siguiente resultado, tanto para el algoritmo normal como al algoritmo refinado.

Tabla 3. Prueba de Kolmogorov-Smirnov para una muestra

| Algoritmo | | | NumNodos |
|-------------|---------------------------|-------------------|----------|
| Sin refinar | N | | 50 |
| | Parámetros normales(a,b) | Media | 975,96 |
| | | Desviación típica | 642,870 |
| | Diferencias más extremas | Absoluta | ,097 |
| | | Positiva | ,097 |
| | | Negativa | -,083 |
| | Z de Kolmogorov-Smirnov | | ,684 |
| | Sig. asintót. (bilateral) | | ,738 |
| Refinado | N | | 50 |
| | Parámetros normales(a,b) | Media | 806,20 |
| | | Desviación típica | 630,946 |
| | Diferencias más extremas | Absoluta | ,101 |
| | | Positiva | ,098 |
| | | Negativa | -,101 |
| | Z de Kolmogorov-Smirnov | | ,714 |
| | Sig. asintót. (bilateral) | | ,688 |

a La distribución de contraste es la Normal.

b Se han calculado a partir de los datos.

Al igual que en el caso anterior ambos valores son 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:

Tabla 4 Comparación de choques entre algoritmos

| Algoritmo | | Frecuencia | Porcentaje | Porcentaje válido | Porcentaje acumulado |
|-------------|----------------|------------|------------|-------------------|----------------------|
| Sin refinar | No hubo choque | 9 | 18,0 | 18,0 | 18,0 |
| | Hubo choque | 41 | 82,0 | 82,0 | 100,0 |
| | Total | 50 | 100,0 | 100,0 | |
| Refinado | No hubo choque | 37 | 74,0 | 74,0 | 74,0 |
| | Hubo choque | 13 | 26,0 | 26,0 | 100,0 |
| | Total | 50 | 100,0 | 100,0 | |

Gráficamente podemos visualizar la relación existente como se muestra en la figura 19.

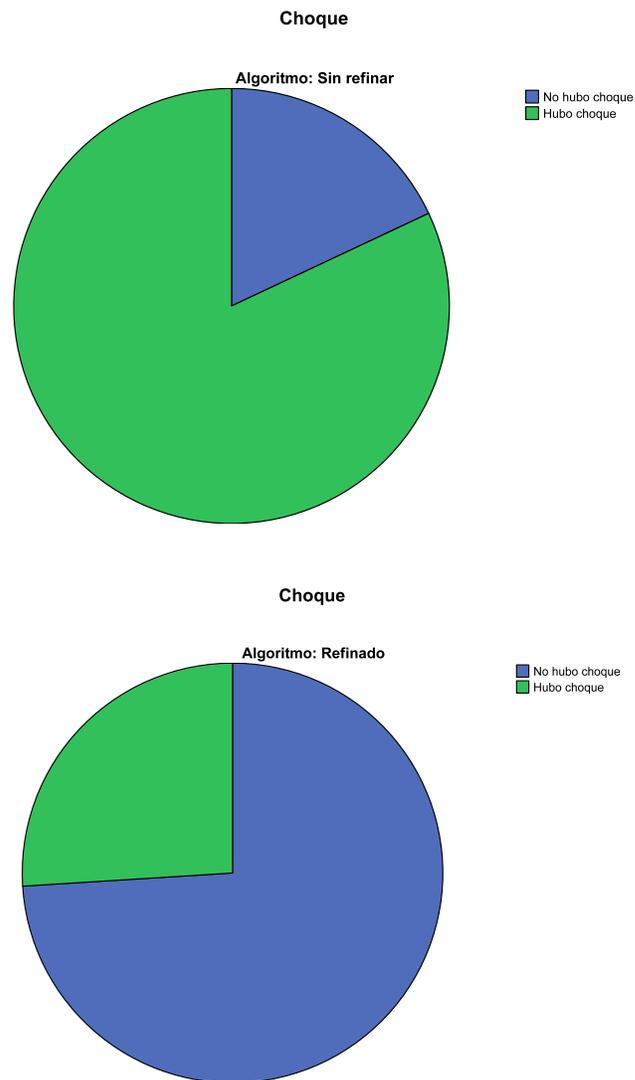


Figura 19. Comparación grafica de los choques

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 que 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 figura 20 y 21.

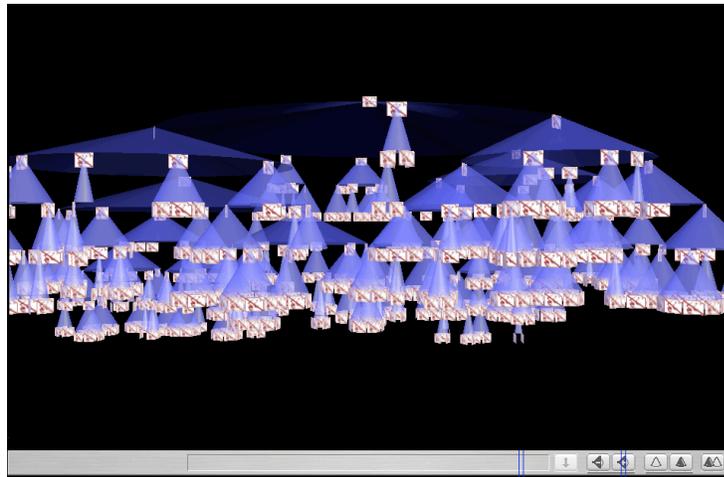


Figura 20. Estructura desplegada con el algoritmo normal

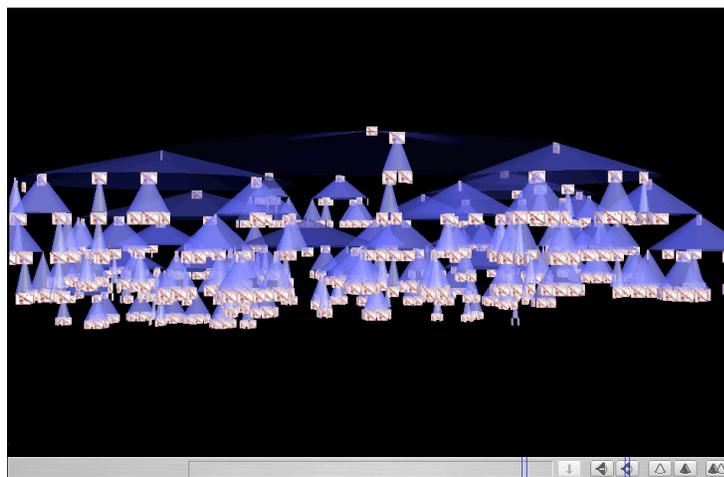


Figura 21. Estructura desplegada con el algoritmo refinado

Como el Biovisualizador ajusta la cámara dependiendo del tamaño del árbol, no se puede percibir si hay un cambio a menos que este cambio sea mínimo.

Una mejor vista nos permite visualizar pequeños cambios en el tamaño de la estructura, pero que no parecen ser significativos. Ver figura 22 y 23.

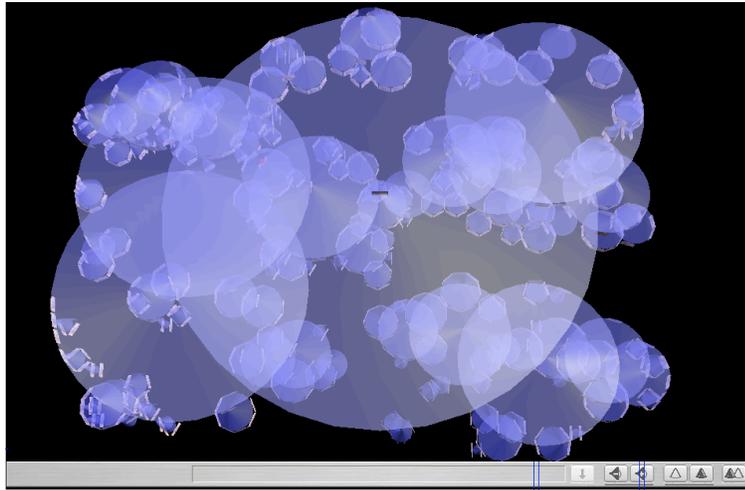


Figura 22. Estructura desplegada con el algoritmo normal

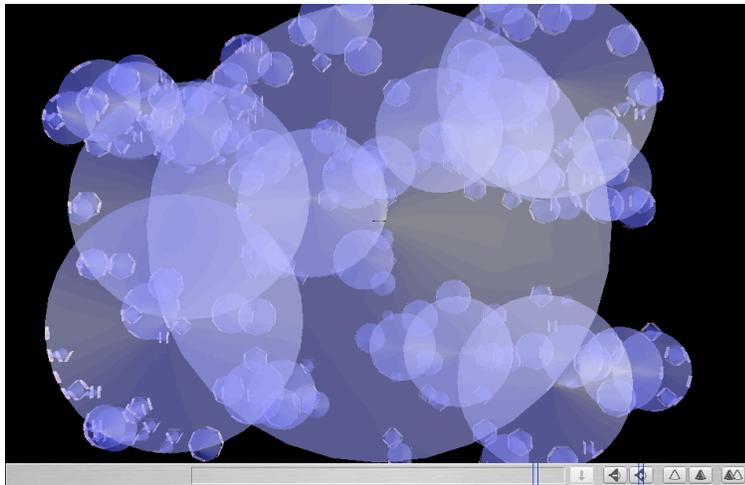


Figura 23. Estructura desplegada con el algoritmo refinado

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.

2.1.1 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.

2.2 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:

- Dos subárboles "hermanos" deben tener muchos hijos
- Los dos subárboles deben de estar cerca uno del otro, es decir, con un ángulo parecido de distribución desde su padre
- Los subárboles deben de tener su descendencia ordenada en línea
- El usuario debe de sacar un nodo que esté a 90 grados en el sentido antihorario del subárbol primero en el sentido horario.
- El usuario debe de sacar un nodo que esté a 90 grados en el sentido horario de otro subárbol. Ver figura 24.

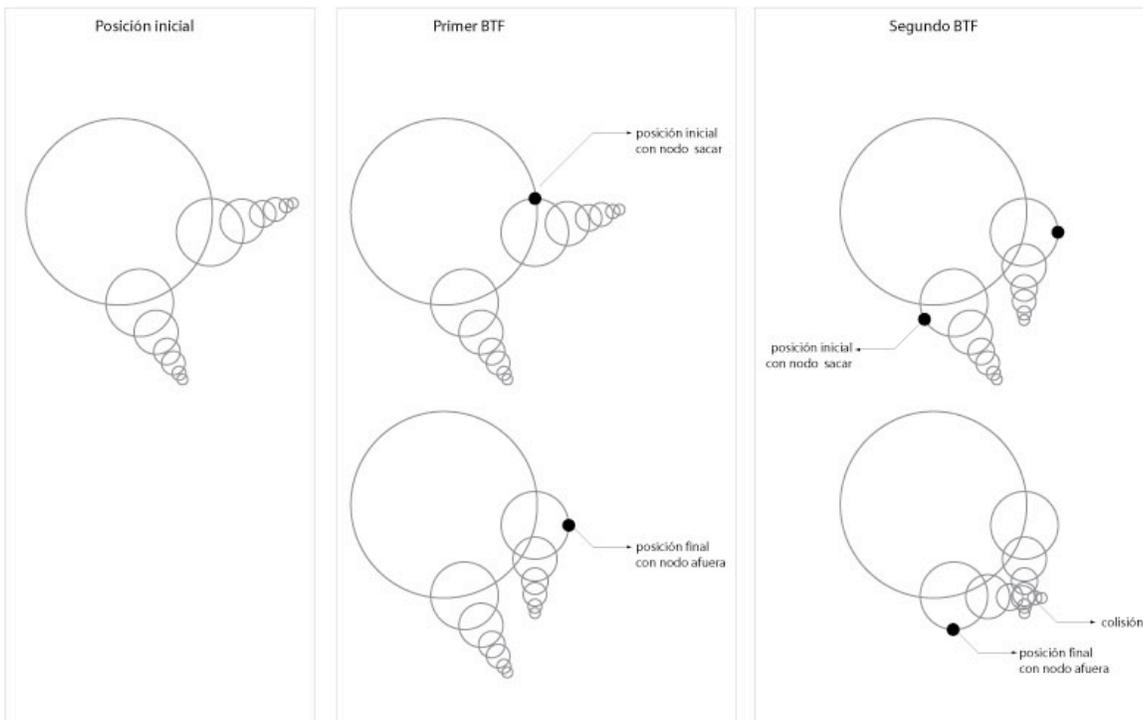


Figura 24. Condiciones para la generación de un choque durante una secuencia de BTFs.

2.2.1 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 circunferencial los subárboles mencionados en el punto anterior. Ver figura 25.

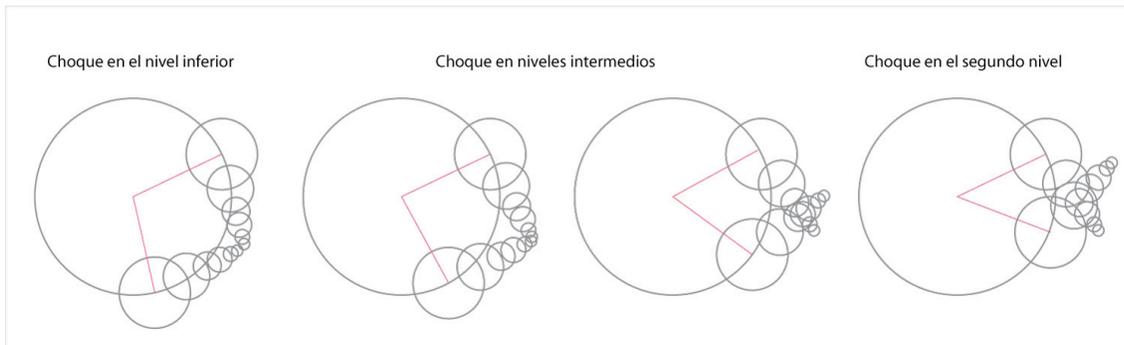


Figura 25. Análisis de los ángulos entre los brazos en posible colisión dependiendo del nivel.

2.2.2 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 de "de lado" con respecto al usuario. Ver figura 26.

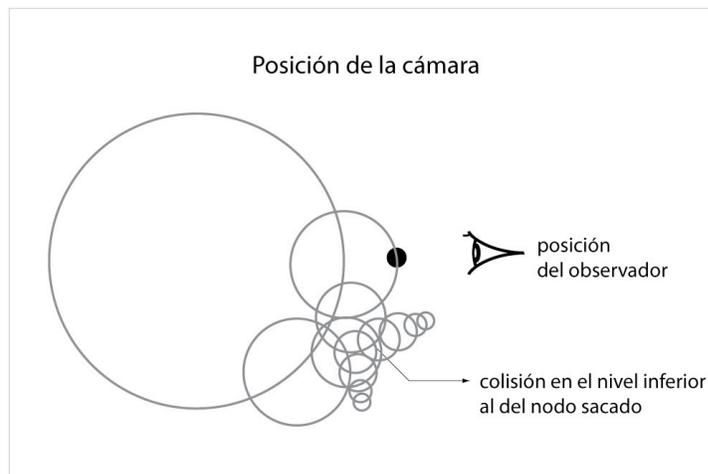


Figura 26. Análisis de la posición del observador en un posible BTF .

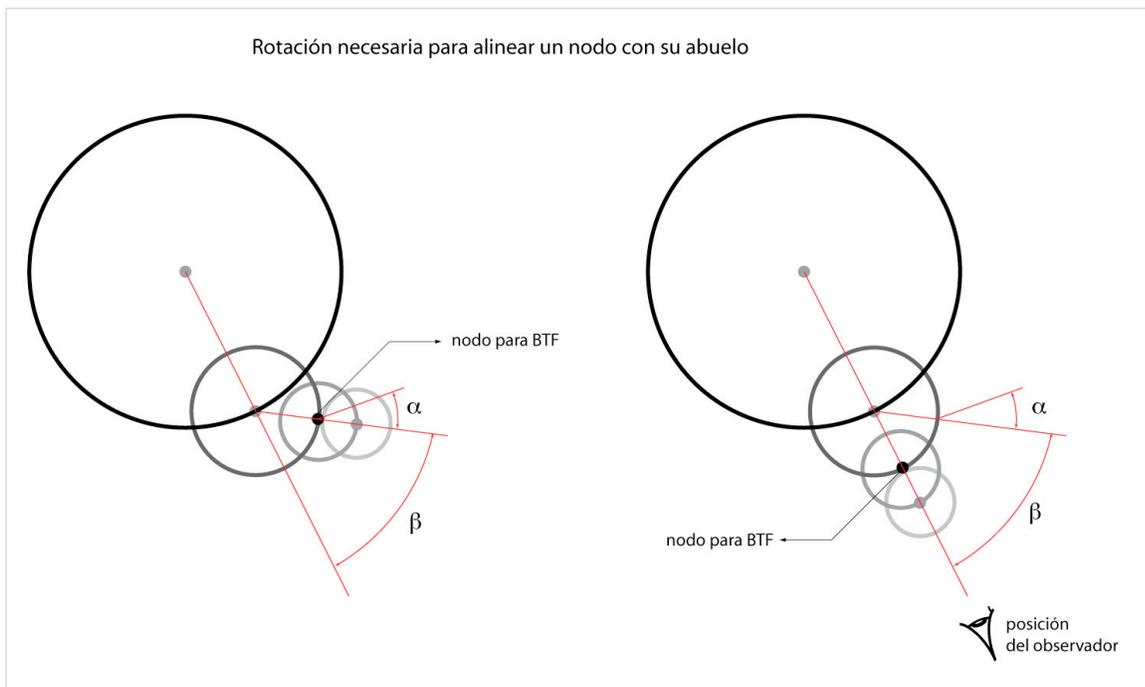
Esto también concluye que sí es posible un choque en el siguiente nivel al del nodo sacado y por lo tanto al del observador, para que se dé esta condición deben cumplirse las siguientes situaciones:

- Dos nodos deben tener mucha descendencia
- Los nodos deben de estar uno al lado del otro

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

2.2.3. Alineación de los subárboles a través del 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 27:



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 figura 28

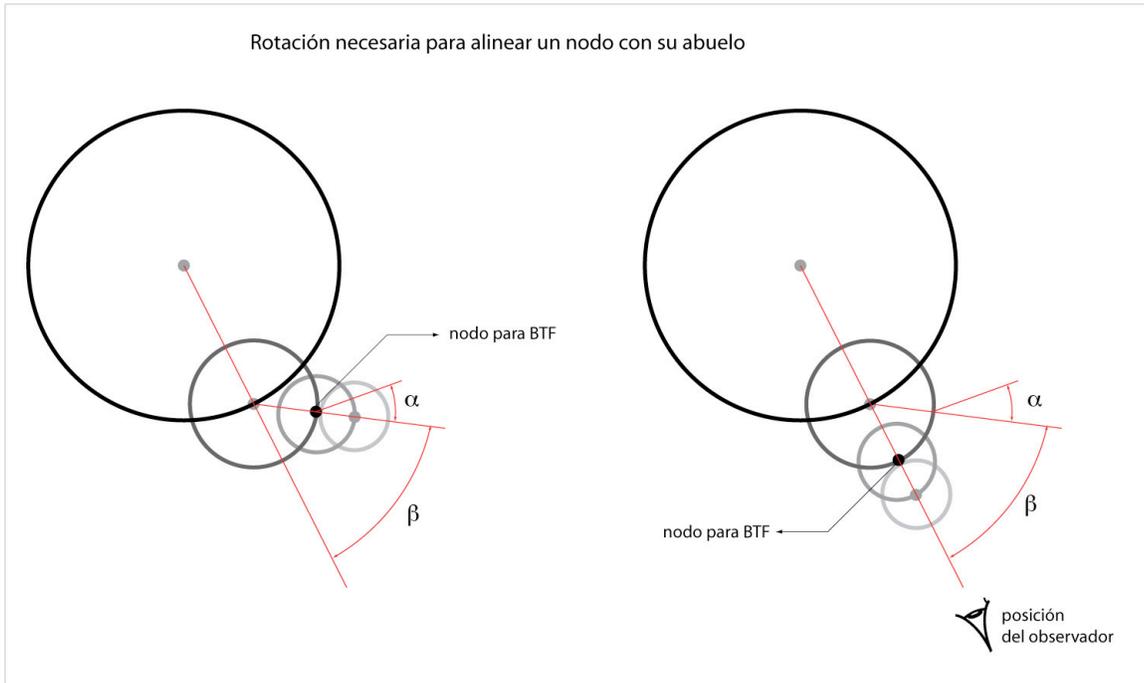


Figura 28. Posiciones necesarias para que con un BTF un nodo se alinee con su abuelo.

Conclusiones y recomendaciones:

- Fue necesario optimizar los algoritmos de generación y posicionamiento de los conos con el fin de minimizar los la posibilidad de choques en el momento de su generación.
- Los árboles con ninguna posibilidad de choques son inviables debido al uso muy ineficiente del espacio disponible para la visualización.
- 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.
- En caso de que se dé choques ocasionados por secuencias de BTFs el choque se daría en un nivel inferior al plano del observador.

Aportes y alcances:

El proyecto representa una muestra algunas posibilidades de mejoramiento de este paradigma, así como explora otras en las que el estado actual del algoritmo es adecuado. A partir de esta experiencia y su retroalimentación el sistema puede ser usado para mejorar otras aplicaciones de este mismo proyecto.

Bibliografía

[B00] R. Boardman. **Bubble Trees. The Visualization of Hierarchical Information Structures. Conference on Human Factors in Computing Systems.** CHI '00 Imperial College. 2000

[BCS04] T. Bladh, D. Carr, and J. Scholl. **Extending Tree-Maps to Tree Dimesions: A Comparative Study.** Springer Verlag.2004

[CK1995] Y. Carriere, R. Kazman. **Interacting with Huge Hierarchies: Beyond Cone Trees.** Obtenido el 4 de marzo del 2007 de la base de datos de la IEEEXplore.

[CK1996] 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 IEEEXplore.

[CMS99] S. Card, J.D. Mackinlay, B. Schneiderman. **Readings in Information Visualization. Using Vision to Think.** Morgan Kaufmann Publishers. San Diego, CA.1999.

[BEW03] R. Becker, S. Eick, A. Wilks. **Visualizing Network Data.** En [CMS99]

[FPF88] 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.

[HM06] 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

[JS91] 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.

- [KY93] 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.
- [M97] T. Munzner. **Laying Out Large Directed Graphs in 3D Hyperbolic Space**. Stanford University. IEEE Symposium on Information Visualization, Phoenix .October 20-21 1997,
- [MB95] 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
- [MCR95] J. D. Mackinlay, S. K. Card, G.G. Robertson. **Cone Trees: Animated 3D Visualizations of Hierarchical Information**. UIR-R-1995
- [MCR95] 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.
- [R03] Ruby M. K. **Information Visualization for Financial Analysis** Tesis de Maestría en Ciencias de la Computación. Department of Computer Science, University of Durham. 2003
- [RC94] **The Table Lens: Merging Graphical and Symbolic Representations** in an Interactive Focus+Context Visualization for Tabular Information. Proceedings of CHI'94, ACM Conference of human factors in computing Systems, New York, 1994.
- [WW99] J. Wijk, H. Wetering. Cushion Treemaps: **Visualization of Hierarchical Information**. Eindhoven University of Technology. 1999.
- [YS2001] F. Yidoung, D. Shihai. **Network Information Visualization Based on Multi-Resolution Animated Cone Trees**. Peking University. 2001

DOCUMENTO II

Cumplimiento de objetivos:

Los objetivos originales del proyecto eran:

Objetivo general: Diseñar e implementar mejores algoritmos que permitan un posicionamiento, dimensionado y rotación dinámica de los conos en un árbol de representación de información jerarquizada

El objetivo general fue alcanzado en todos sus aspectos al haber mejorado el posicionamiento y dimensionado inicial minimizando la posibilidad de choques. A su vez se encontró que la rotación dinámica no es necesario mejorarla pues se considera que no contribuye de modo significativo a la posibilidad de choques.

Objetivos específicos:

- a. Mejorar el dimensionado de los conos actuales (disminuir espacio tridimensional ocupado). Logrado.
- b. Mejorar el posicionamiento radial de los nodos en las bases de los conos (disminuir espacio tridimensional ocupado). Logrado.
- c. Minimizar la cantidad de veces en que el algoritmo permite el traslape de los conos en niveles inferiores de la estructura. Logrado
- d. Mejorar el algoritmo de rotación dinámica de los conos (tiempo de ejecución). No fue necesario

Limitaciones y problemas encontrados:

El proceso se llevó a cabo como se planeó. Quizás la única limitación fue el excesivo tiempo que se requirió para obtener el equipo necesario aun cuando se contaba con el presupuesto debidamente aprobado.

Observaciones generales y recomendaciones:

Se recomienda continuar con investigaciones que exploren el campo de la visualización tridimensional de datos pues se considera una rama importante para disciplinas como simulación y modelaje de conocimientos en todos los campos de las ingenierías.