Ampliación de Informática Gráfica

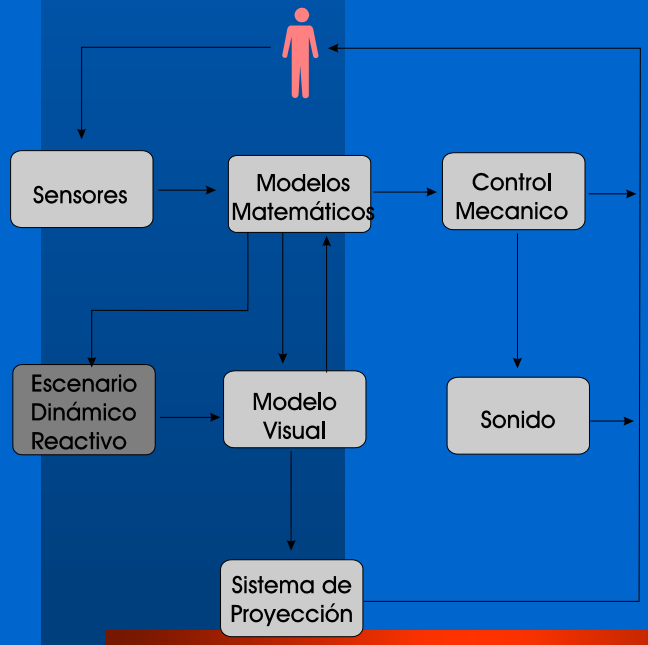
Tema 6. Gráficos Tiempo-Real

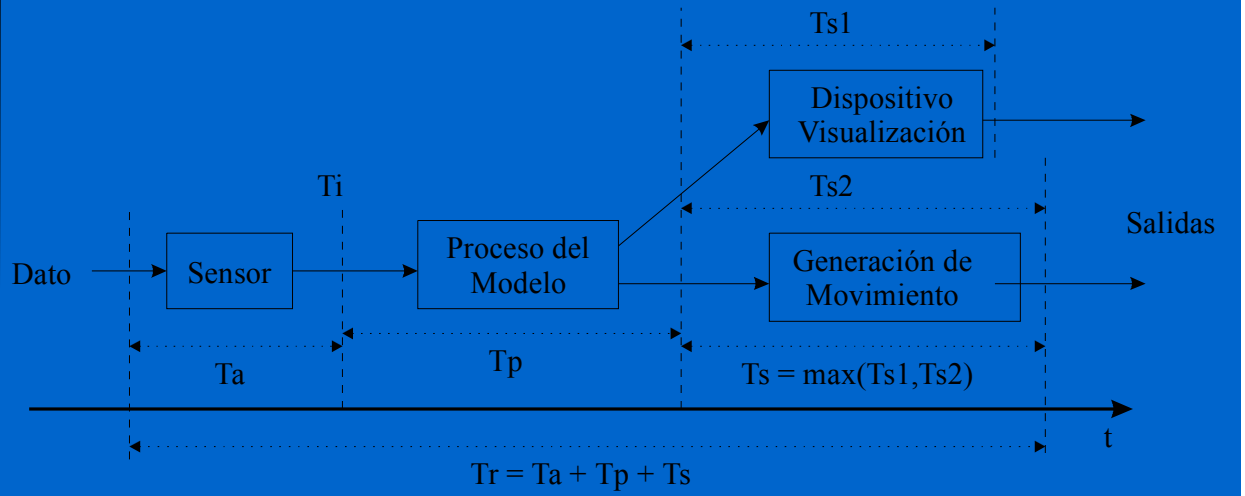
Índice del Tema.

- Introducción a los gráficos Tiempo-Real
- Factores que Influyen en el coste del Proceso de Visualización.
- Técnicas de Optimización del Coste de Visualización:
  - Técnicas de eliminación de partes no visibles.
  - Técnicas de Selección del Nivel de detalle.
- Estructuras de datos avanzadas para gráficos tiempo Real.
- Técnicas de control de carga en los gráficos TR.

Gráficos y Simulación Tiempo-Real.

- **Concepto de Simulación MIL y TR.**
  - **Simulación MIL implica Man-In-the-Loop.**
  - **Tenemos además restricciones Temporales: Tiempo de Respuesta del Sistema.**





Gráficos y Simulación Tiempo-Real.

- **Por tanto tenemos que tener en cuenta:**
  - **Frecuencia de Salida.**
  - **Frecuencia de Cuadro / Frecuencia de refresco.**
  - **El tiempo aparente y tiempo físico.**
  - **Sincronización entre estímulos/subsistemas**

Factores en el Coste de Visualización.

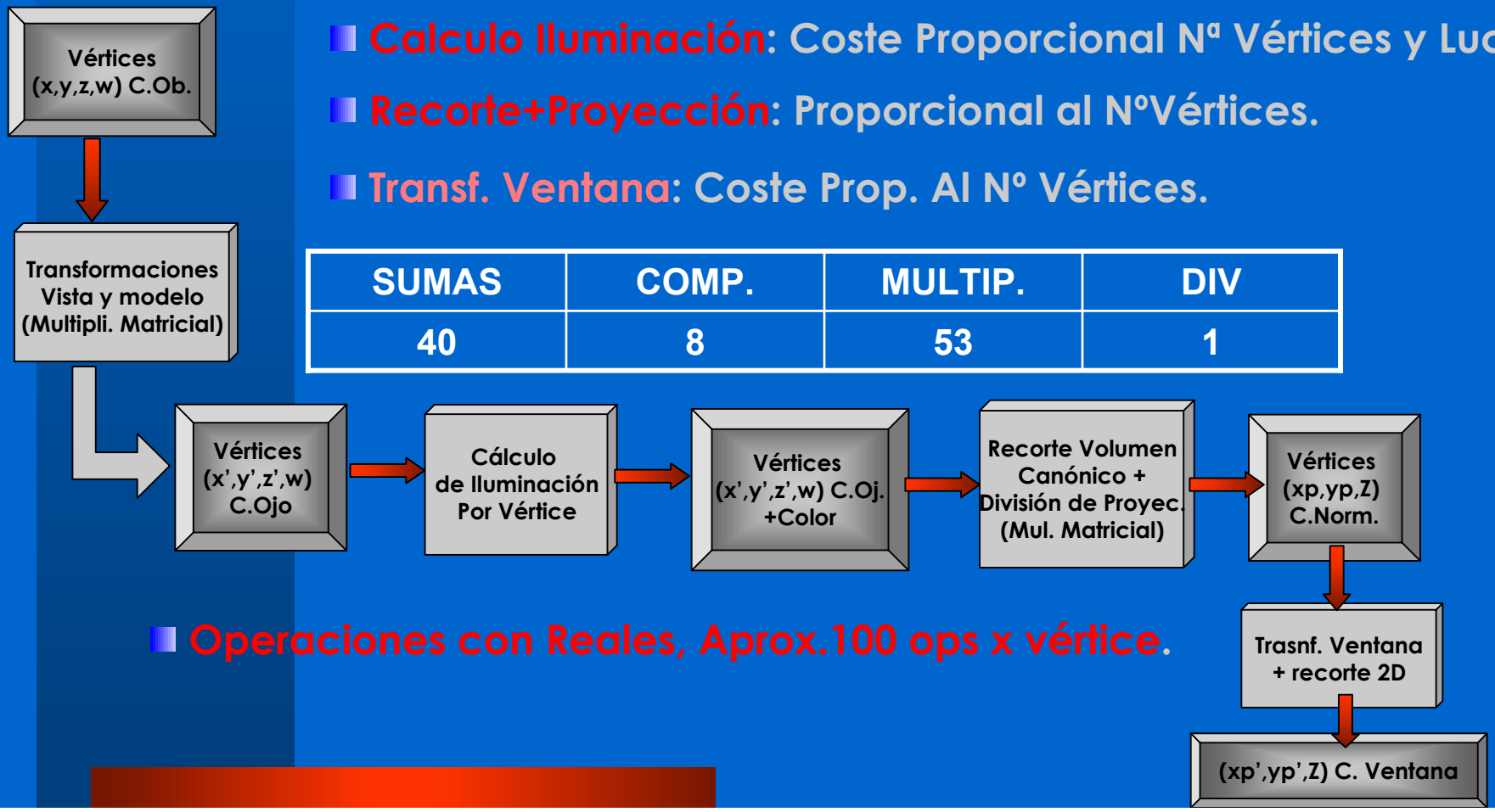
- Si recordamos la estructura de procesamiento para la generación de una imagen presentada en el tema 3, tenemos dos etapas principales:
  - Procesamiento de vértices. Trabajo realizado en coordenadas 3D.
  - Procesamiento de *pixels*. Trabajos realizados en coordenadas de imagen.

Factores en el Coste de Visualización.

- Factores que Influyen en el coste de la Transformaciones sobre Vértices. Revisión de la tubería de Vértices

- **Transf. VM**: Coste Proporcional al número de vértices
- **Calculo Iluminación**: Coste Proporcional Nª Vértices y Luces
- **Recorte+Proyección**: Proporcional al N°Vértices.
- **Transf. Ventana**: Coste Prop. Al Nº Vértices.

SUMAS	COMP.	MULTIP.	DIV
40	8	53	1

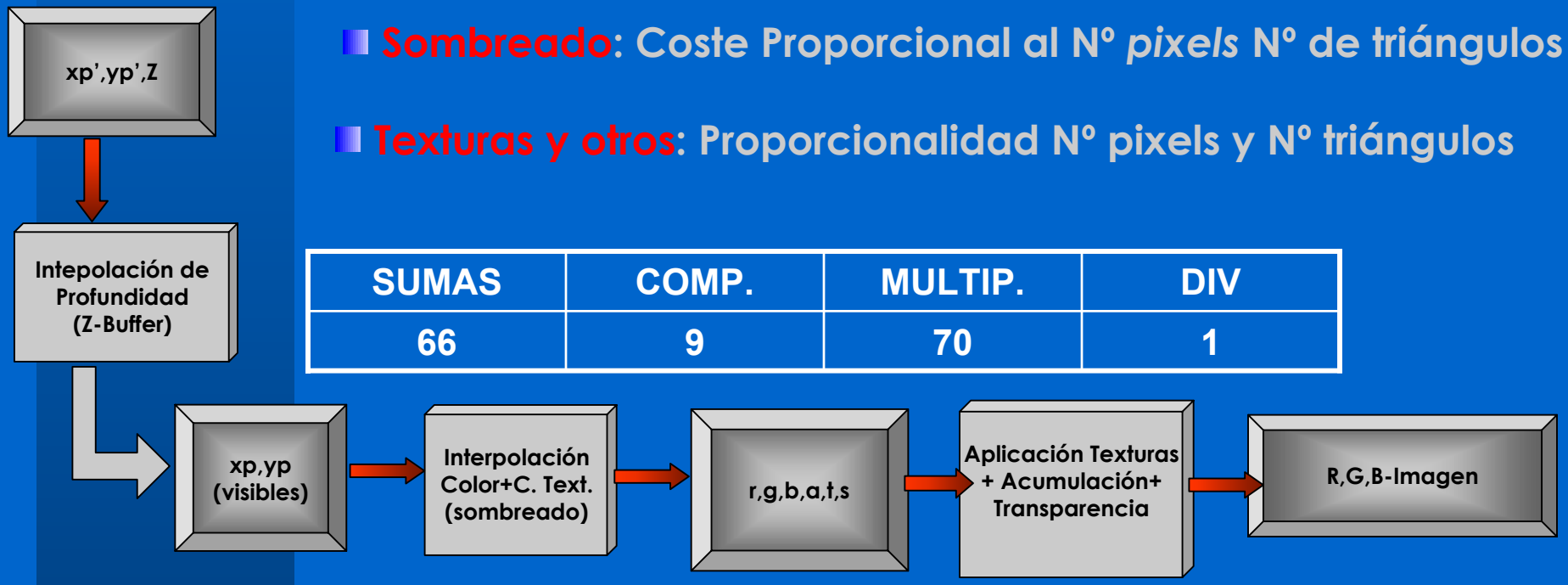


- Operaciones con Reales, Aprox.100 ops x vértice.

Factores en el Coste de Visualización.

- Factores que Influyen en el coste de las operaciones sobre *pixels*.

- **Z-buffer**: Coste Proporcional al Nº *pixels* Nº de triángulos
- **Sombreado**: Coste Proporcional al Nº *pixels* Nº de triángulos
- **Texturas y otros**: Proporcionalidad Nº pixels y Nº triángulos



SUMAS	COMP.	MULTIP.	DIV
66	9	70	1

- **Operaciones con enteros. Aprox 150 ops.**

Factores en el Coste de Visualización.

- Otro elemento importante es la Transferencia de datos entre la etapas de la *pipeline*.
  - En la tubería de vértices es proporcional al Nº de Vértices.
  - En la tubería de pixels es proporcional al Nº vértices y al Nº de *pixels* de la imagen final.
- También se deben considera las ordenes de configuración de los estados de las etapas de la *pipeline* que requieren accesos a memoria.
  - Matrices del Modelo.
  - Activación de Luces.
  - Propiedades de los materiales.
  - Imágenes de textura.

Factores en el Coste de Visualización.

- Por tanto a la hora de optimizar el coste podemos atacar el problema mediante:
  - Limitación el número de Vértices considerados.
    - ***Reduce el coste de la* pipeline *de vértices y de paso mejora prestaciones de la de* pixels *(menos triángulos en las primeras fases)***
    - ***Reduce las transferencias en pipeline de vértices.***
  - Limitación el número de *Pixels* considerados.
    - ***Mejora los tiempos de rellenado.***
    - ***Acelera las operaciones de texturado.***
    - ***Reduce las transferencias en la pipeline de* pixels.**
  - Reducir las transferencias de datos en la *pipeline*
- Todo esto sin cambiar la calidad final de la imagen generada.
- Cuidar la configuración de la *pipeline*, importante evitar cuellos de botella, ya que producen una degradación general en la arquitectura.

Posible Ahora veremos las Técnicas más comunes

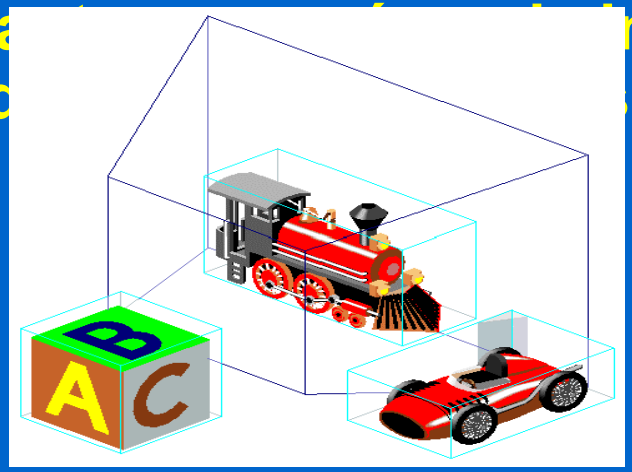
Complicado Requiere Soporte Hardware (DVR)

Ordenación adecuada del Envió de datos a la Pipeline

Lo importante será controlar qué partes de la escena y en qué orden se mandan a la *pipeline* en cada momento

DETERMINACIÓN DE "QUÉ PARTES"

- Elemento de Partida: Queremos una imagen que refleje la escena representada por los modelos 3D.
- La imagen final de la escena depende de donde situemos la cámara en cada instante y del ángulo que deseemos cubrir. Se pueden dar las siguientes situaciones:
  - Objetos que caen fuera de mi campo visual.
  - Objetos que aún estando en mi campo visual son tapados por otros objetos.
  - Objetos que están más cerca de la cámara y ocupan por ta magen y objetos que c . Percibimos más detalles



DETERMINACIÓN DE "QUÉ PARTES"

- **A partir del análisis previo podemos describir dos técnicas básicas de reducción de partes de la escena:**
  - **Selección de objetos en función de su visibilidad**
    - ***Eliminación de objetos fuera del campo de visual definido por la pirámide de visión. Métodos de recorte por visibilidad o culling.***
    - ***Eliminación de objetos ocluidos. Métodos de precálculo de visibilidad.***
  - **Selección del nivel de detalle de los objetos.** Técnicas de gestión dinámica de LOD (*Level-of-Detail*)

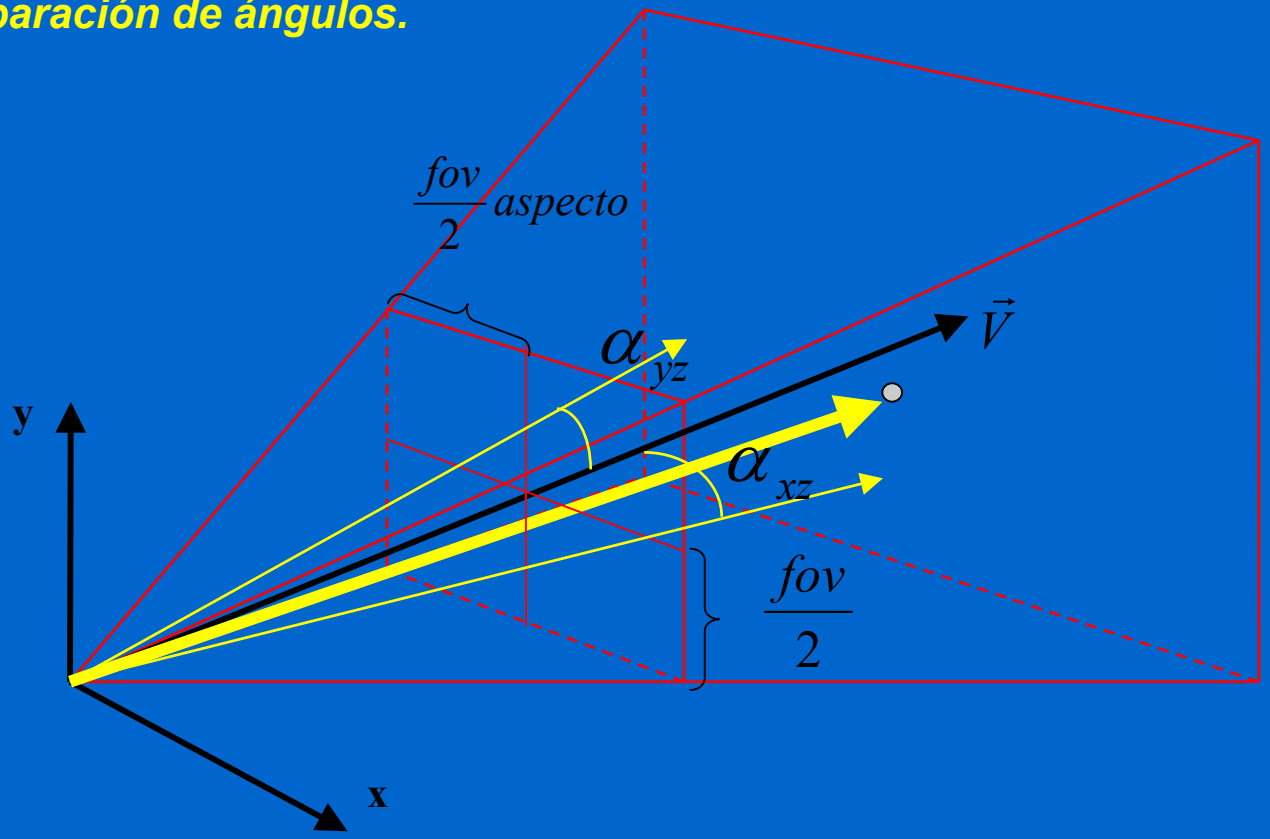
MÉTODOS DE RECORTE POR VISIBILIDAD

- **Método dinámico, evaluación en todos los *frames*, dependencia con el punto de vista.**
- ***El método produce ahorro solamente a la tubería de vértices, ya que los vértices serían eliminados en el proceso de recorte contra el volumen canónico .***
- **Método de grano grueso, evaluación por "objeto" y no por vértice, sería un coste excesivo.**
- **Método aproximado, evaluación basada en estimación sobre volúmenes envolventes.**
  - *Cajas Envolventes.* Mejor Precisión
  - *Esferas envolventes.* *Mejor Coste*

MÉTODOS DE RECORTE POR VISIBILIDAD

– Cálculo simple determinación del ángulo formado por los puntos que definen la envolvente y el punto de visión cae dentro o de los ángulos de visión. Operaciones involucradas:

- ***Determinación del Vector en los extremos de la envolvente.***
- ***Determinación del producto escalar en yz y xz***
- ***Comparación de ángulos.***



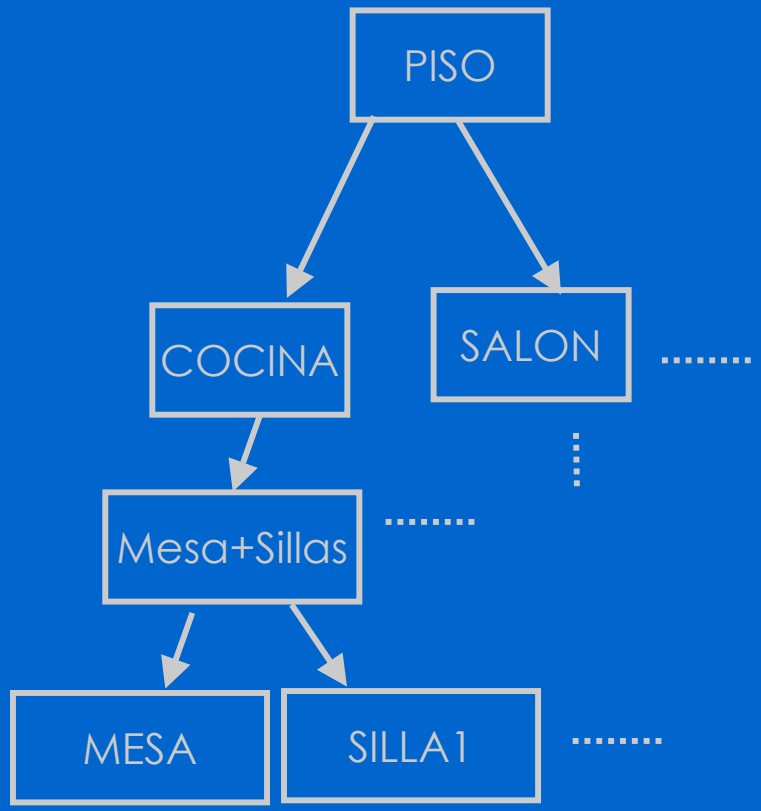
MÉTODOS DE RECORTE POR VISIBILIDAD

- Aquí vemos un ejemplo de implementación en código C.

```
char test_recorte_visibilidad(float *pvista,float *prrg,float *bbx[3],int nvbbx,
                              float cosfovv, float cosfovh)
{
        float VecVistayz[2];
        float VecVistaxz[2];
        float aux[2],aux2[2];

        VecVistayz[0]=pref[1]-pvista[1];
        VecVistayz[1]=VecVistaxz[1]=pref[2]-pvista[2];
        VecVistaxz[0]=pref[0]-pvista[0];

        normaliza(VecVistayz);
        normaliza(VecVistaxz);

        for(i=0;i<nvbbx;i++)
        {
                aux[0]=bbx[i][1]-pvista[1];
                aux[1]=aux2[1]=bbx[i][2]-pvista[2];
                aux2[0]=bbx[i][0]-pvista[0];
                normaliza(aux);
                if((aux[0]*VecVistayz[0]+aux[1]*VecVistayz[1])>cosfovv &&
                        (aux2[0]*VecVistaxz[0]+aux2[1]*VecVistaxz[1]))>cosfovh))
                                return(0);


        }

        return(1);
}
```

MÉTODOS DE RECORTE POR VISIBILIDAD

- El método de recorte por visibilidad puede mejorarse mediante la utilización de una jerarquía de volúmenes envolventes que minimice el número de test de recorte a realizar.
- Esta jerarquía puede realizarse en fase de montaje de la escena y suele apoyarse en las estructuras de organización de escenas que veremos en otros de los puntos de este tema.

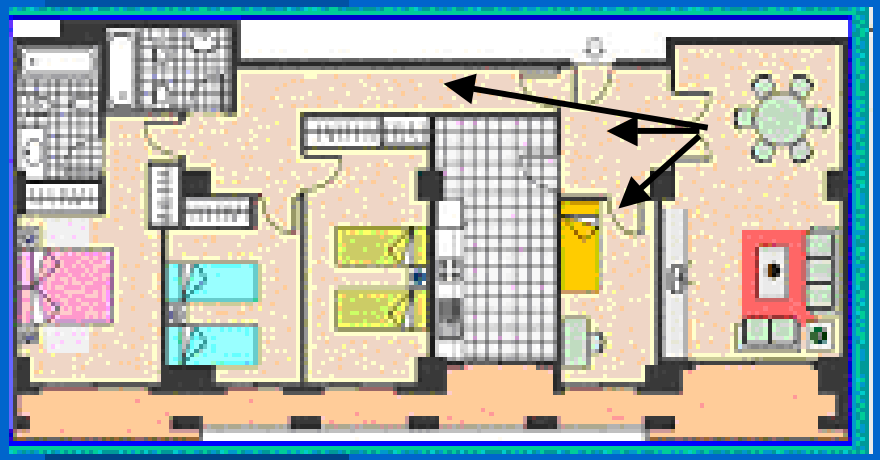


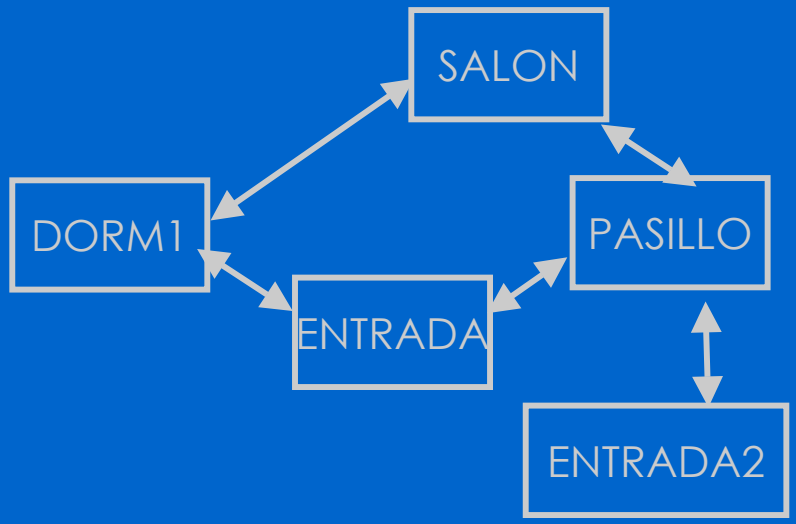
MÉTODOS PRECALCULO DE VISIBILIDAD

- Estos métodos como hemos indicado se utilizan para evitar el envió a la *pipeline* de objetos ocluidos por otros elementos de la escena. Esto afecta a:
  - ***tubería de vértices completa.***
  - ***Primera fase de la tubería de pixels, test de Z-buffer.***
- Se trata también de métodos gruesos aplicados sobre objetos y no sobre polígonos o vértices individuales.
- Se denomina de precalculo por que se realizan durante la preparación de la escena, ya que la determinación de oclusión en tiempo de ejecución puede resultar excesivamente costosa para el sistema (no obstante existe algunos métodos para realizarlo).
- Pese a ser métodos de precálculo, la oclusión depende de la visibilidad, por tanto estos métodos tendrán que generar estructuras que puedan ser evaluadas dinámicamente durante el tiempo de dibujado.
- El tipo métodos empleados se basan en modelos de partición espacial como los vistos en el tema 1. A estos métodos se les asocian listas de pertenencia de objetos. El modelo de partición espacial a utilizar se selecciona en función del tipo de escena.

MÉTODOS PRECALCULO DE VISIBILIDAD

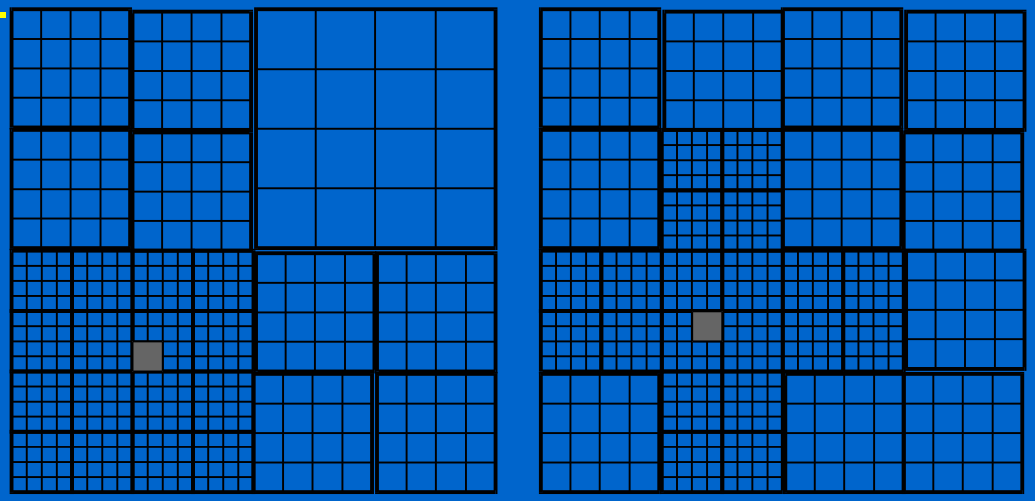
- Métodos basados en Partición Espacial de Celdas Adyacentes.
  - Son útiles en entornos de visualización de edificios o entornos cerrados con dependencia.
  - La descomposición en celdas se basa en la distribución propia de habitaciones y pasillos del edificio.
  - Tipo de estructura generada: gráfo de adyacencias, donde cada nodo tiene asociados los elementos contenidos.





MÉTODOS PRECALCULO DE VISIBILIDAD

- Métodos en Particiones Jerárquicas y Áreas de Definición de la Representación
  - Suelen emplearse en espacios abiertos, visualización de terrenos.
  - Las oclusiones no siguen un patrón regular y se deben a la existencia de montañas, casas etc.
  - Es usual utilizar como base las descomposición general del terreno en árboles de tipo *quadtree*, asociado a los nivel del *quadtree* determinados objetos en función del a visibilidad. El ADR el estado del árbol adecuado para cada posición del observador.



MÉTODOS SELECCIÓN DE NIVEL DE DETALLE

- Como dijimos se basan en la pérdida de detalles de los objetos en función de su tamaño aparente en la escena.
- Necesitan la definición de varios niveles de complejidad en los objetos, por tanto suponen un incremento en el coste de almacenamiento.
- Elementos a considerar:
  - Técnicas de decisión dinámica del nivel de detalle.
  - Modelos de transición entre niveles de detalle.

MÉTODOS SELECCIÓN DE NIVEL DE DETALLE

- Decisión Sobre el Cambio de Nivel de detalle.
  - Basado en la distancia.

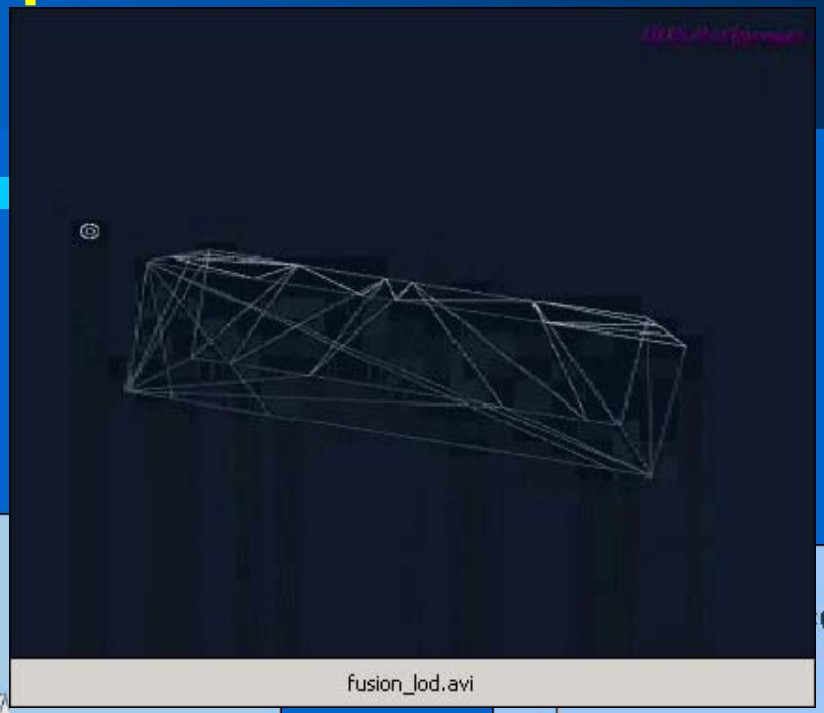
  - Basado en diferencia entre niveles y aspecto aparente. Requiere la estimación de la diferencia visual entre niveles y la estimación del espacio ocupado por el modelo en la pantalla en un momento dado.
  - Basado en capacidades del sistema. Selección para mantener un número de cuadros/s establecido.

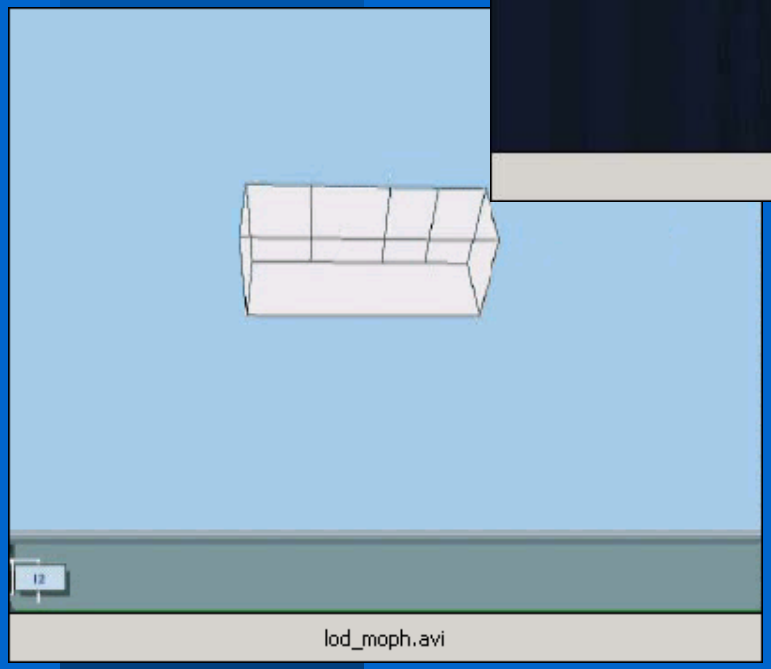
MÉTODOS SELECCIÓN DE NIVEL DE DETALLE

- **Modelos de Transición Entre Niveles de Detalle.**
  - **Modelo de cambio directo. Es el más simple pero también que más se aprecia, si no se cuidan los criterios de cambio y los modelos disponibles.**
  - **Modelo de fundido. Se realiza una transición suave basada en el cambio del valor de transparencia entre los modelos. Problema durante la transición tenemos dos modelos y los costes adicionales de fundido.**
  - **Modelos de *Morphing*. Se basa en la transición continua entre las formas de los dos niveles de detalle. Se emplean las técnicas descritas en el tema 5.**
  - **Modelos multiresolución de transición continua. Disponemos un modelo de indexación poligonal que permite devolver el nº deseado de vértices en cada instante.**

MÉTODOS SELECCIÓN DE NIVEL DE DETALLE

- **Ejemplos de funcionamiento.**





Estructuras de Datos Para Visualización 3D-TR

- **Estructuras de datos para gestión de simulación 3D TR:**
  - **Grafos aciclicos dirigidos.**
  - **Recorrido o *traversal* del grafo se obtiene como resultado un conjunto de primitivas con las transformaciones del modelo acumuladas.**
  - **Los nodos sirven para organizar la escena, aplicar transformaciones, toma de decisiones, indicar geometría, propiedades visuales de los objetos, etc.**

Estructuras de Datos Para Visualización 3D-TR

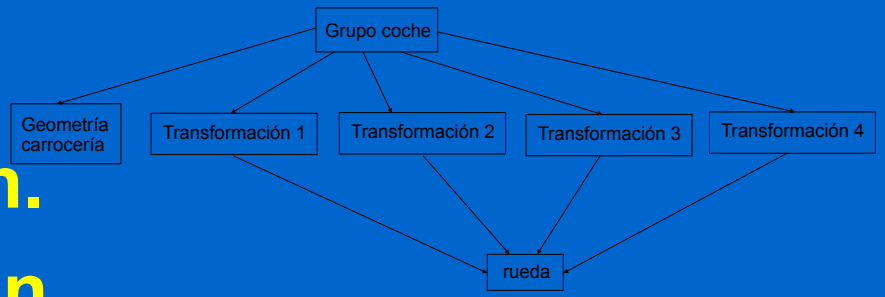
- **Categorías de Nodos:**
  - **Nodos Visuales.**
  - **Nodos de Operación.**
  - **Nodos de Transformación.**
  - **Nodos de Retroalimentación..**

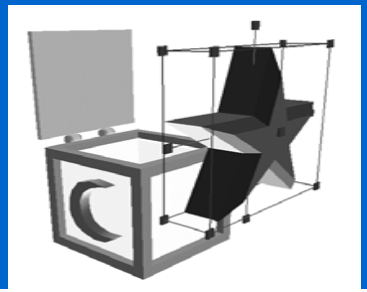
Estructuras de Datos Para Visualización 3D-TR

- **Nodos Visuales: Se trata de los nodos terminales del grafo.**
  - **Nodos de Geometría.**
  - **Nodos de Propiedades.**
- **Nodos de Operación: Gestionan el recorrido del grafo.**
  - **Nodos de agrupación: se utilizan a nivel organ.**
  - **Nodos de Recorte y selección por Visibilidad.**
  - **Nodos de nivel de detalle.**
    - **Centro**
    - **Rango de distancias**
  - **Nodos de aplicación (switch)**

Estructuras de Datos Para Visualización 3D-TR

- **Nodos de Transformación: Producen cambios de localización, tamaño etc en sus hijos.**
  - **Nodos de Transformación Afín: Transformaciones del modelo.**
  - **Nodos de Articulación**
  - **Nodos para Animación.**
  - **Nodos de Manipulación.**



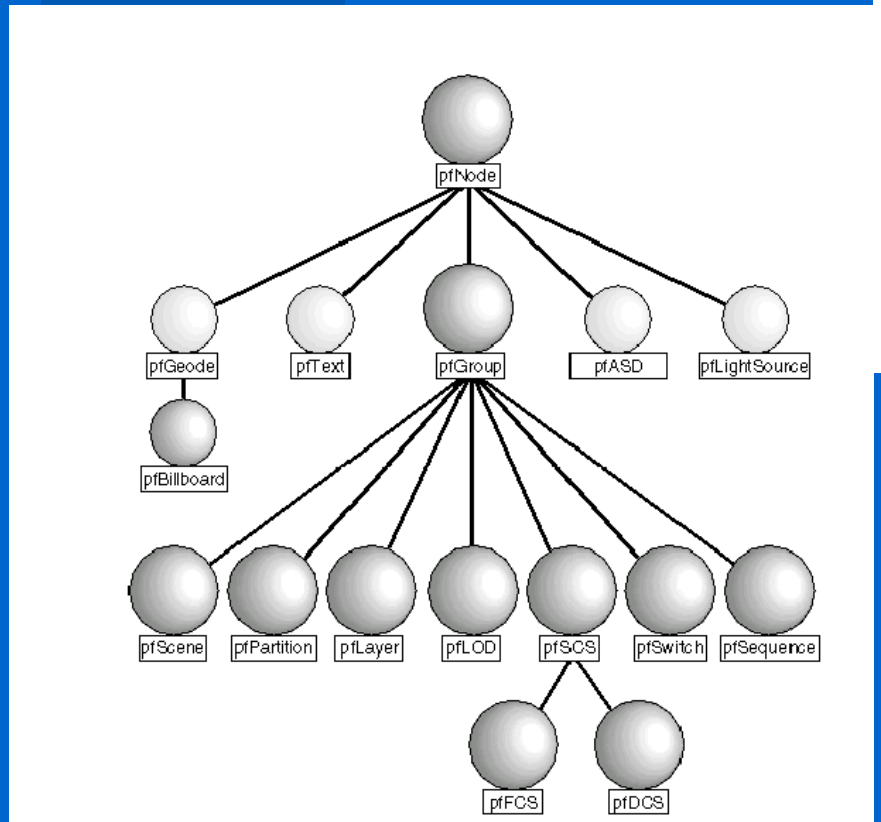


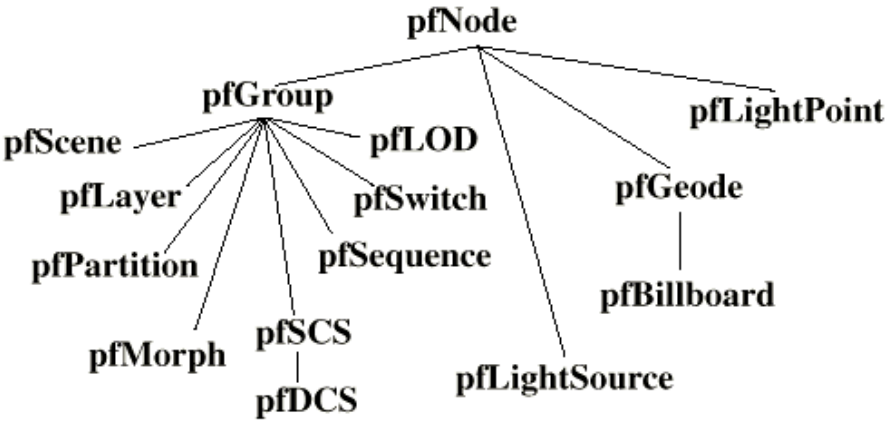
Estructuras de Datos Para Visualización 3D-TR

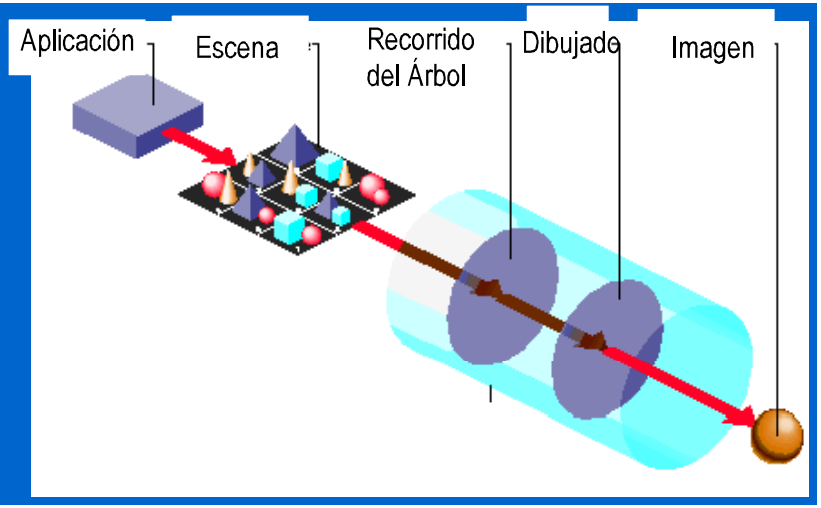
- **Nodos de Retroalimentación: Recogen información durante el recorrido del grafo, activan funciones, etc.**
  - **Nodos de Información**
  - **Nodos de Navegación**
  - **Nodos de Activación.**
  - **Nodos de Selección.**

Estructuras de Datos Para Visualización 3D-TR

- **Ejemplo IRIS Performer**





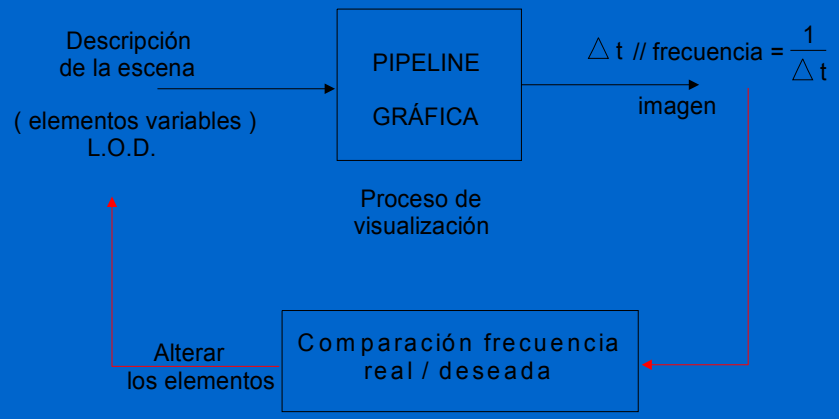


Control de la Frecuencia de Cuadro en gráficos TR

- **Control de la frecuencia de refresco.**
  - **La idea es que lo importante es mantener un numero de cuadros por segundo, por tanto hemos de aplicar estrategias de control jugando con los factores que tenemos: quitar calidad de los objetos.**
- **Estrategias de control:**
  - **Método Adaptativo:Bucle Cerrado.**
  - **Método Predictivo: Lazo Abierto.**

Control de la Frecuencia de Cuadro en gráficos TR

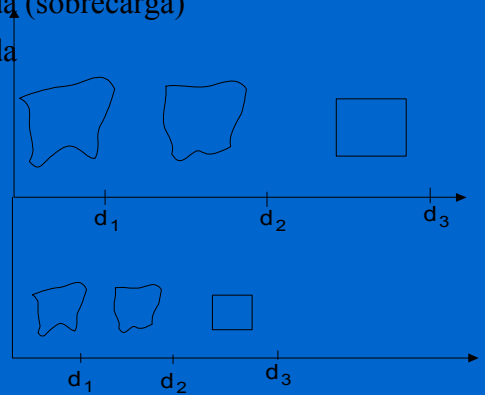
- **Método de control Adaptativo: Bucle clásico de control en lazo cerrado.**



- **Definición del factor de estrés: determinación crítica problemas de oscilación.**

$$f.e. = \frac{f_{deseada}}{f_{real}} \quad \begin{cases} f.e. > 1 & \rightarrow \text{frecuencia real menor que la deseada (sobrecarga)} \\ f.e. < 1 & \rightarrow \text{frecuencia real mayor que la deseada} \end{cases}$$

modulamos el rango de distancias $\rightarrow$ $d_i' = \frac{d_i}{f.e.}$



- **Independiente de la máquina**

Control de la Frecuencia de Cuadro en gráficos TR

- **Método de control Predictivo: Bucle abierto. Tratamos de enviar solo aquello que la máquina es capaz de pintar en el tiempo disponible.**
- **Estimación de coste / beneficio visual.**

$$\forall objeto\ i \begin{cases} C_{ij} \rightarrow \text{coste de la visualizacion para cada nivel de detalle j de i} \\ b_{ij} \rightarrow \text{'beneficio' visual que aporta ese LOD j del objeto i} \end{cases}$$

$$C_{ij} = \alpha \cdot n^{o}\ vertices + \beta \cdot area\ objeto$$

- **Maximizar el beneficio para un coste determinado: problema de la mochila discreta. NP-Completo.**

$$b = \sum_{i} b_{i,f(i)}$$

- **Depende de la máquina.**