

# Generación y Visualización en Ciudades



Miguel A. Astor R.

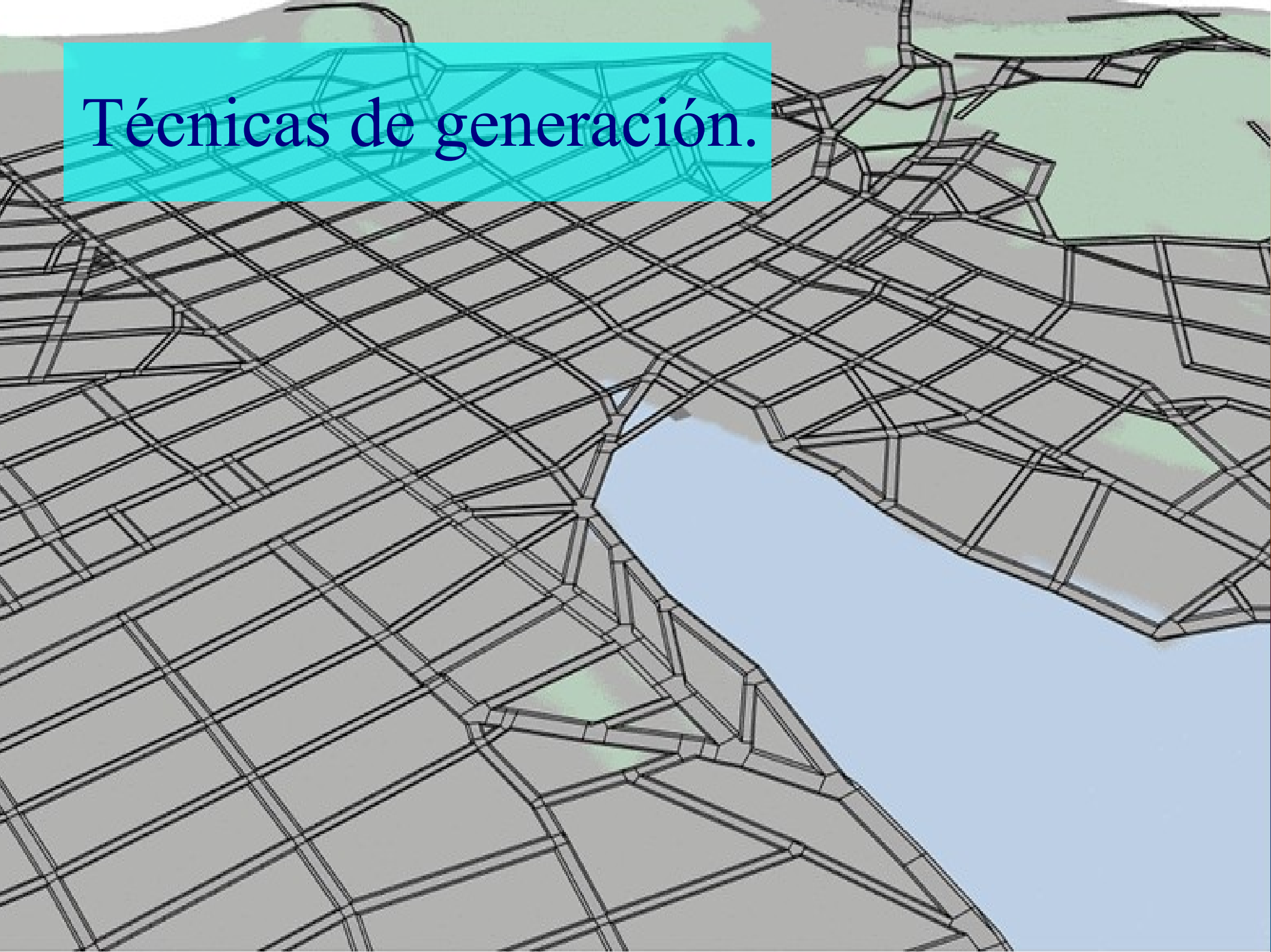
# Contenido.

- Introducción.
- Tecnicas de generación:
  - Generación del plano de la ciudad.
  - Generación de la vialidad.
  - Generación de edificaciones.
  - Generación de texturas.
- Tecnicas de despliegue:
  - *Hierarchical Occlusion Culling.*
  - *View Frustum Filling.*
- Aplicaciones y conclusiones.

# Introducción.

- La creación de entornos virtuales rurales o urbanos implica la producción y visualización de enormes cantidades de datos.
- Las soluciones se adaptan de problemas similares:
  - Creación procedimental de planos urbanos, carreteras, autopistas, edificaciones y texturas.
  - Reconstrucción de datos a partir de información tomada de entornos reales.
  - Aplicación de técnicas de *culling*.
  - Uso de diferentes niveles de detalles.

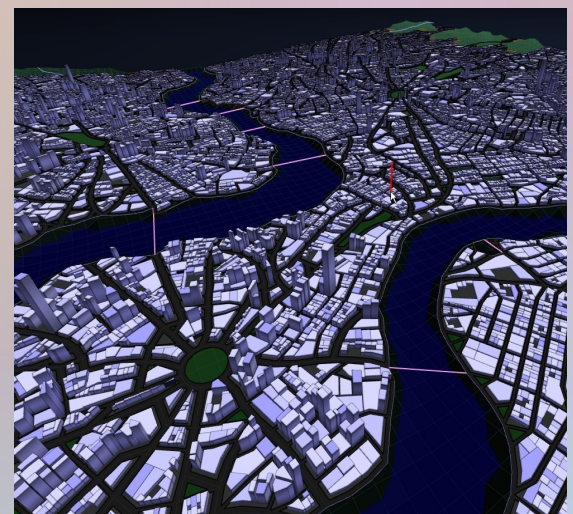
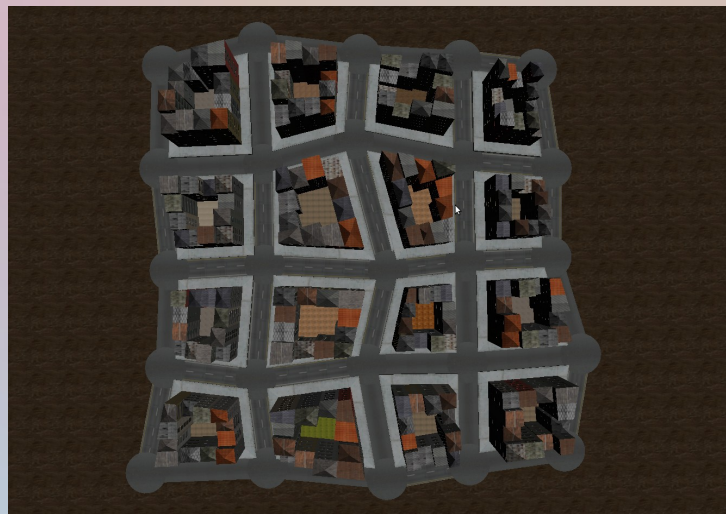
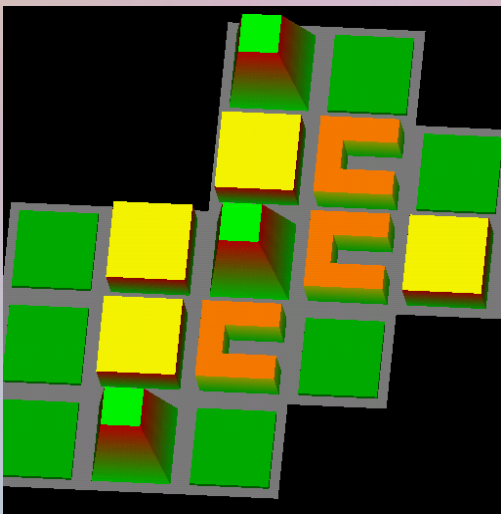
# Técnicas de generación.





# Técnicas de generación.

- Generación del plano de la ciudad.
  - Mallas regulares.
  - Mallas irregulares.
  - Reglas de producción parametrizadas. Gramáticas y *L-Systems*.



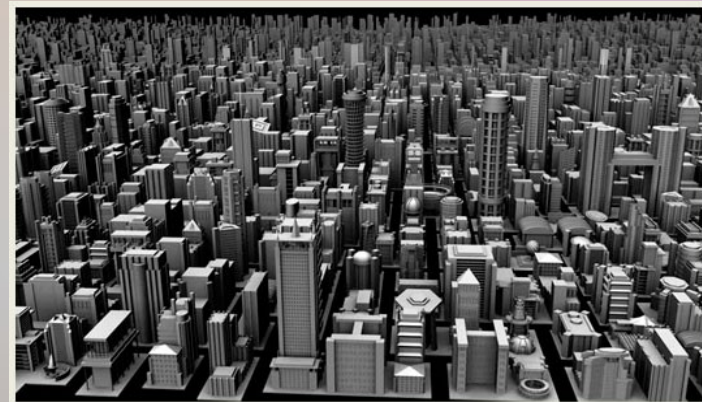
# Técnicas de generación.

- Mallas Regulares.
  - Se utiliza un *grid* regular. Puede deformarse siempre y cuando se parta de celdas iguales.
  - El resultado es muy poco realista.
  - Puede adaptarse facilmente para generar ciudades “infinitas” de forma rapida.



# Técnicas de generación.

- Mallas irregulares.
  - Puede obtenerse unificando celdas de un *grid* regular, generando *quad-trees* aleatorios, o subdividiendo anillos irregulares concéntricos.
  - La estructura del mallado puede ser muy evidente, especialmente en tomas aéreas.





# Técnicas de generación.





# Técnicas de generación.

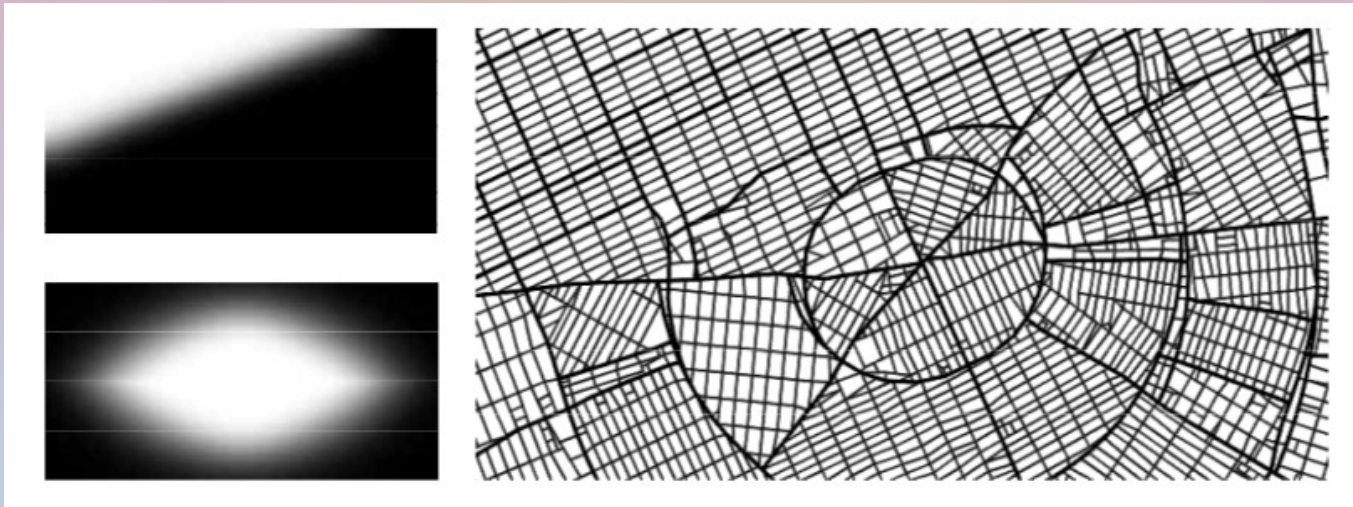
- Gramáticas y *L-Systems*.
  - Se utilizan para generar cadenas partiendo de un alfabeto, símbolos de sustitución y reglas de producción.
  - La cadena viene a representar series de carreteras o autopistas interconectadas.
  - Pueden generarse de manera aleatoria, o basarse en parámetros que determinan qué regla de producción aplicar.
  - Su complejidad crece muy rápidamente.

# Técnicas de generación.

- *Self-sensitive L-Systems* (CityEngine). Basado en dos conjuntos de reglas:
  - Globales:
    - Los caminos siguen un patron predefinido.
    - Los caminos buscan la menor elevación en el terreno.
    - Las autopistas conectan centros de alta densidad de población. Las carreteras conectan autopistas.
  - Locales:
    - Las carreteras no cruzan el agua. Las autopistas si.
    - Los caminos se rotan para adaptarse al terreno.
    - Nuevas carreteras tratan de unirse en las intersecciones.

# Técnicas de generación.

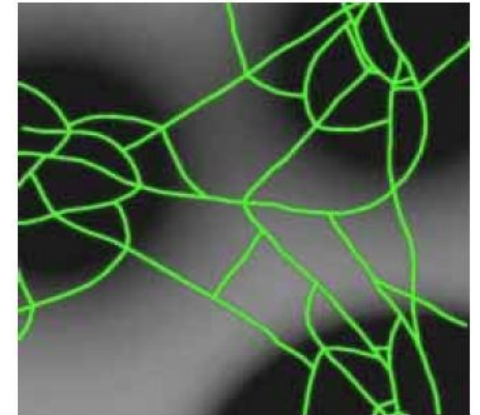
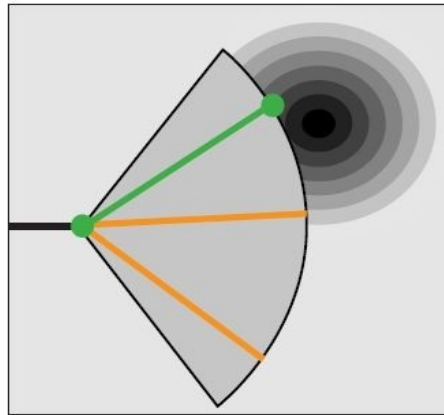
- Parámetros:
  - Alturas del terreno de la ciudad.
  - Densidad poblacional.
  - Distribución del agua y la vegetación en el terreno.
  - Patrones de carretera.



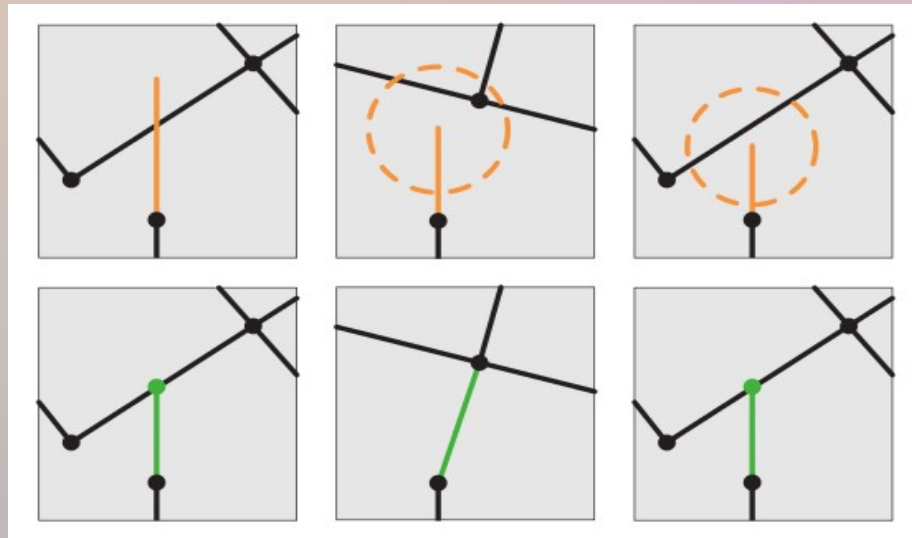


# Técnicas de generación.

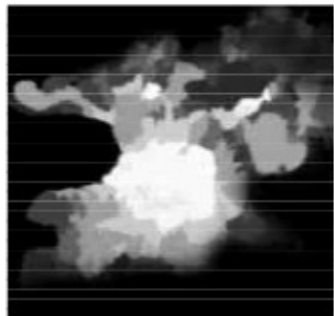
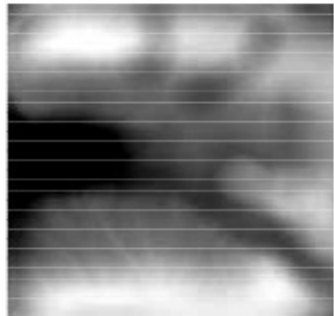
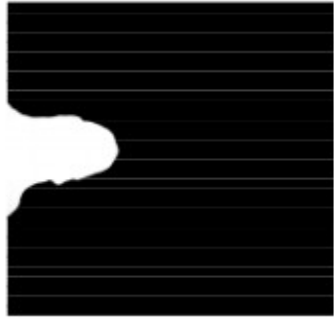
- Generación de autopistas:



- Generación de carreteras:

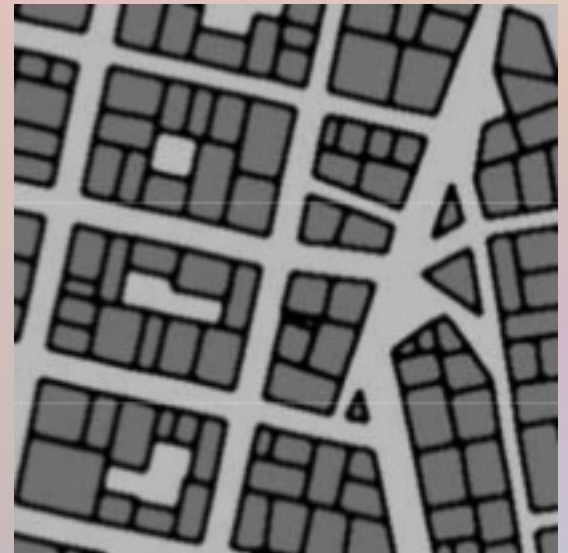
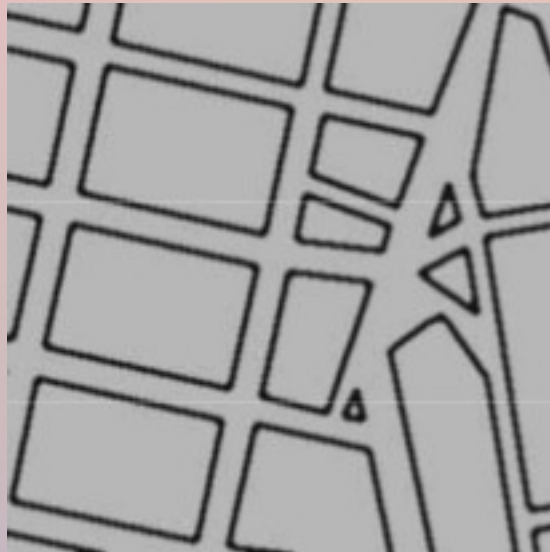
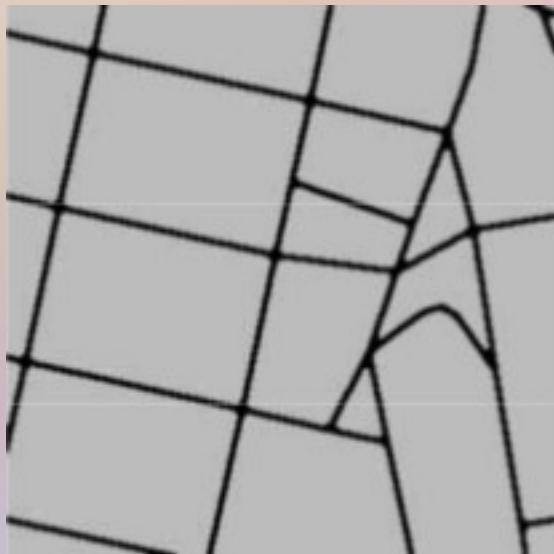


# Técnicas de generación.



# Técnicas de generación.

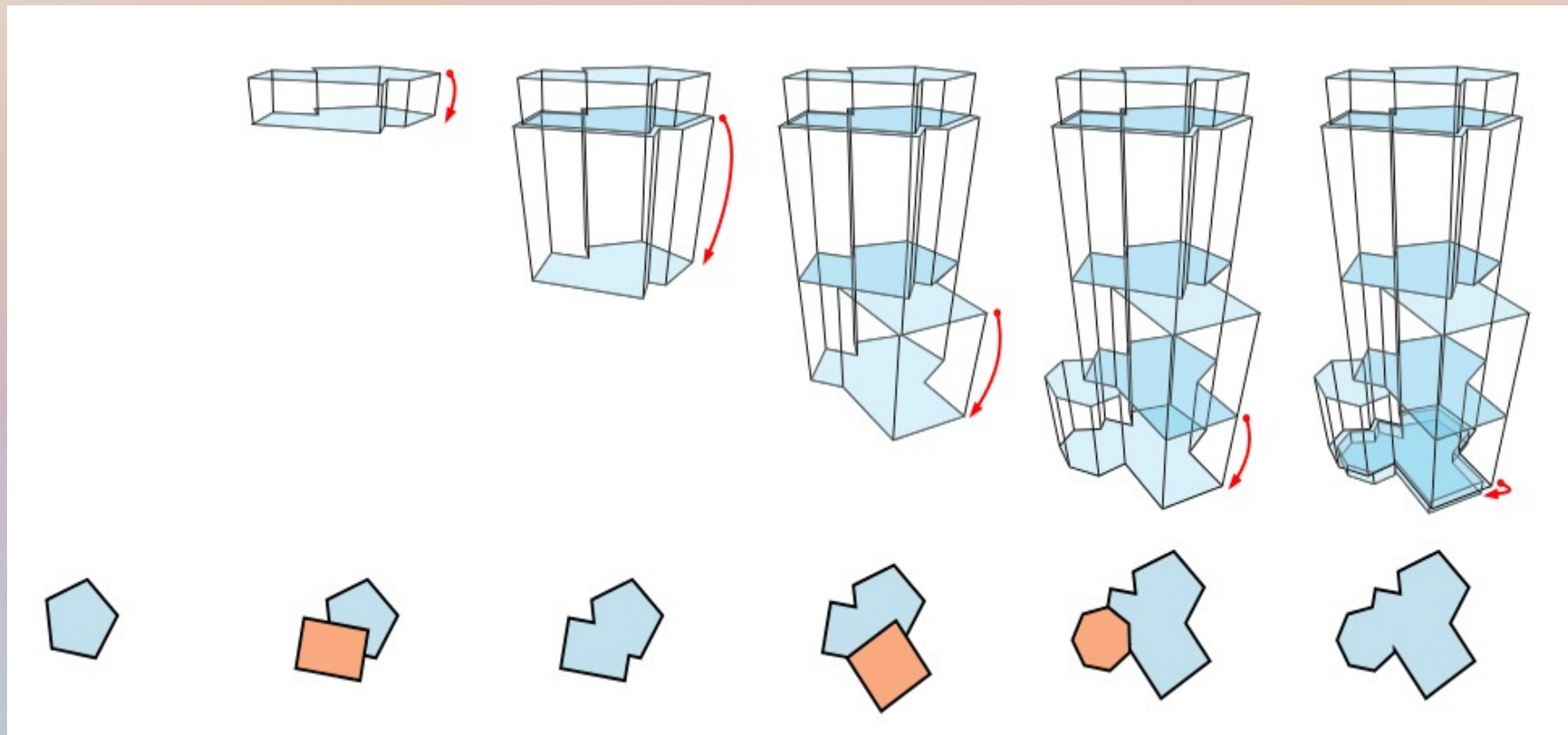
- Generación de la vialidad:
  - Dilatación de las aristas del mallado.





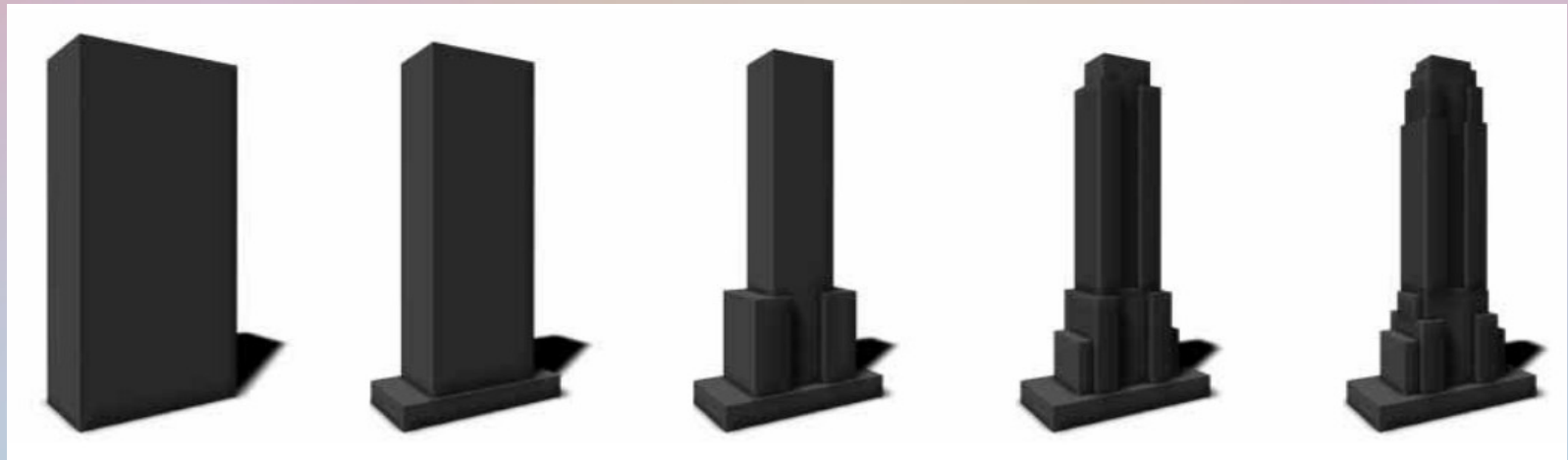
# Técnicas de generación.

- Generación de las edificaciones:
  - Extrusión iterativa de planos de planta.



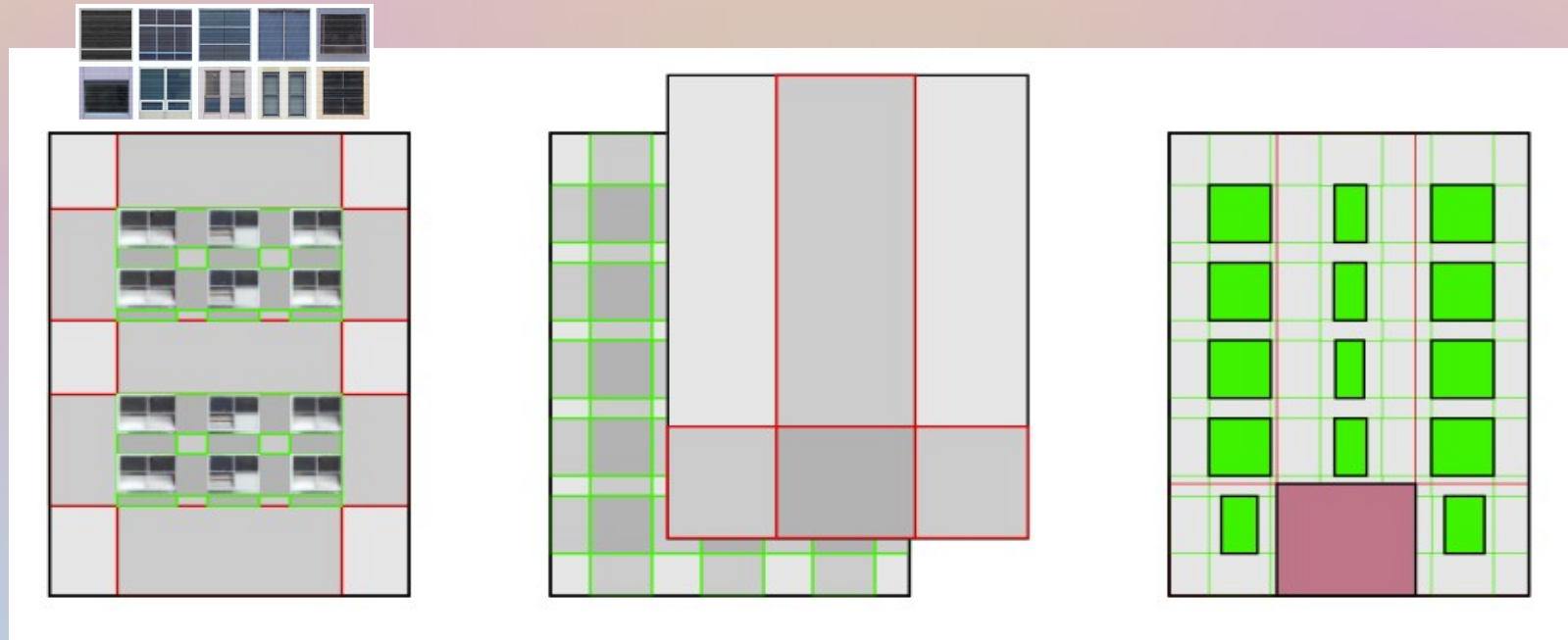
# Técnicas de generación.

- Subdivisión aleatoria del *Bounding Box* (*L-Systems*):
  - Las reglas de producción son transformaciones geométricas y extrusiones.
  - Los parámetros son la altura máxima de los edificios por zona de la ciudad y el tipo de edificio por zona.
  - Los niveles de detalle son automáticos.



# Técnicas de generación.

- Generación de las texturas.
  - Subdivisión de las caras del edificio en *grids* diferentes.
  - Combinación de texturas pequeñas en una mayor.





# Técnicas de despliegue.



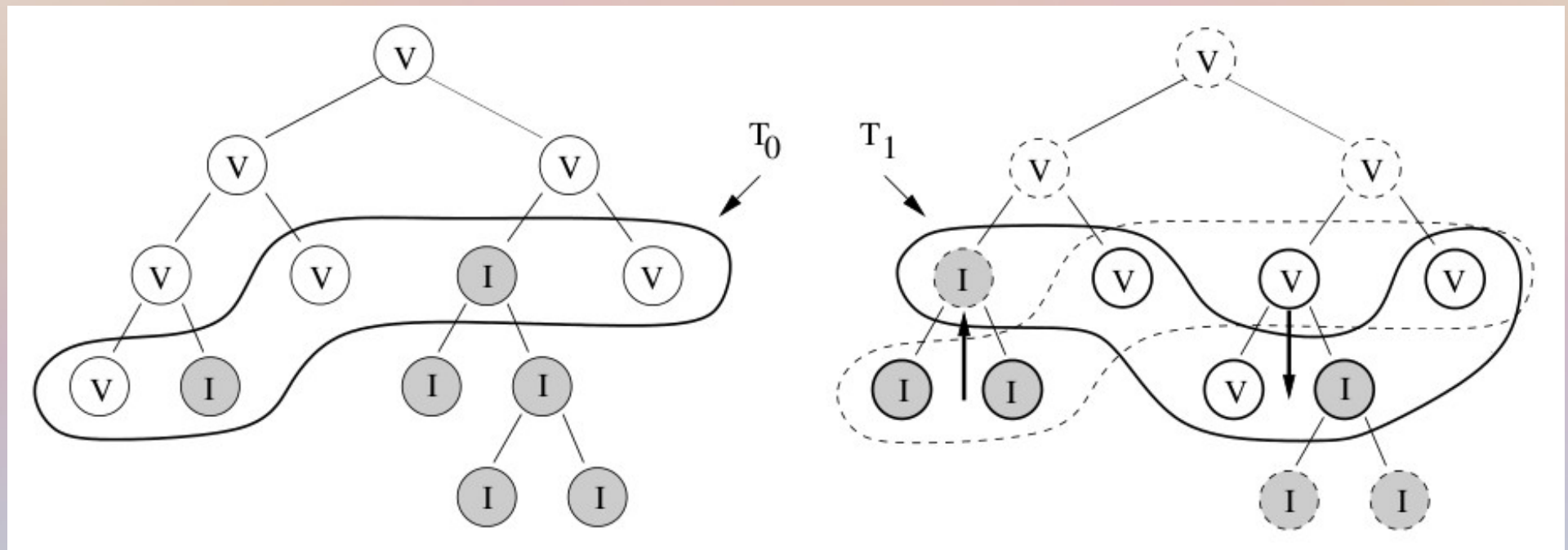
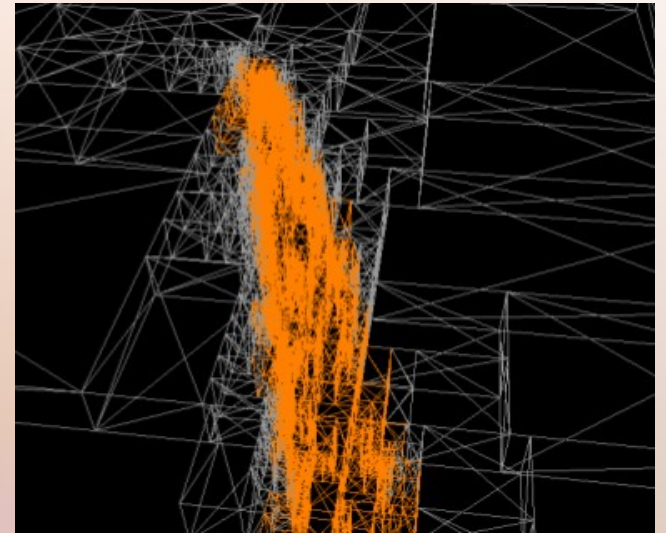
# Técnicas de despliegue.

- *Occlusion Culling* jerárquico.
  - La escena se organiza en bloques dentro de alguna estructura jerárquica (ej. *K-D Tree*).
  - Para el caso de ciudades, la ciudad se divide en regiones según algún criterio. Las regiones se dividen en subregiones.
  - Se prueban las regiones que superan el frustum culling.
  - El árbol se recorre recursivamente hasta que se determina que un nodo es invisible, o es una hoja visible.

# Técnicas de despliegue.

- Algoritmo (por cada frame):
  - Examinar el estado de todas las pruebas incompletas.
  - Si la prueba esta lista determinar si el nodo es visible o no.
  - Si es visible, marcar sus hijos para prueba. Sino, marcar todo su subarbol como invisible.
  - Recorrer el conjunto terminal. Dibujar los nodos que estan marcados como visibles y los que eran visibles en el frame anterior.
  - Solicitar *occlusion queries* para los nodos terminales que no tengan *queries* pendientes.

# Técnicas de despliegue.



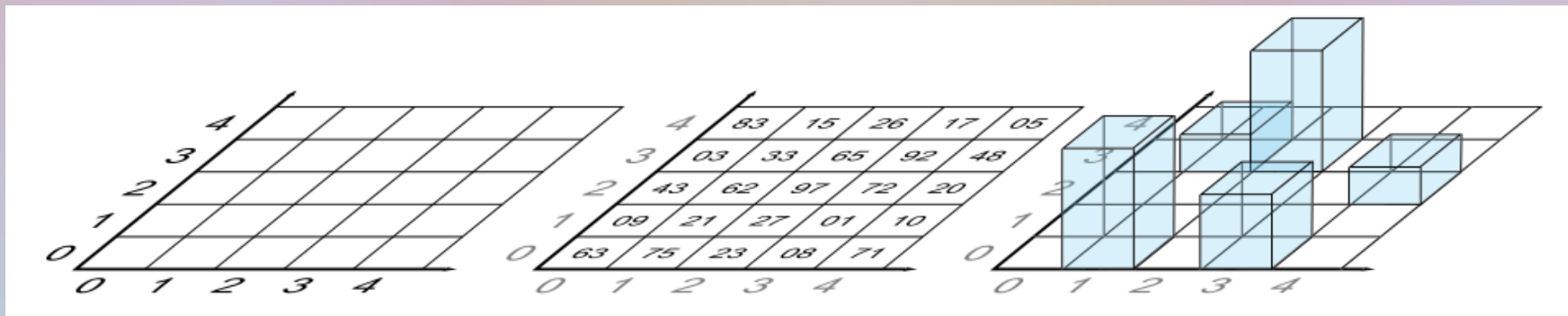


# Técnicas de despliegue.

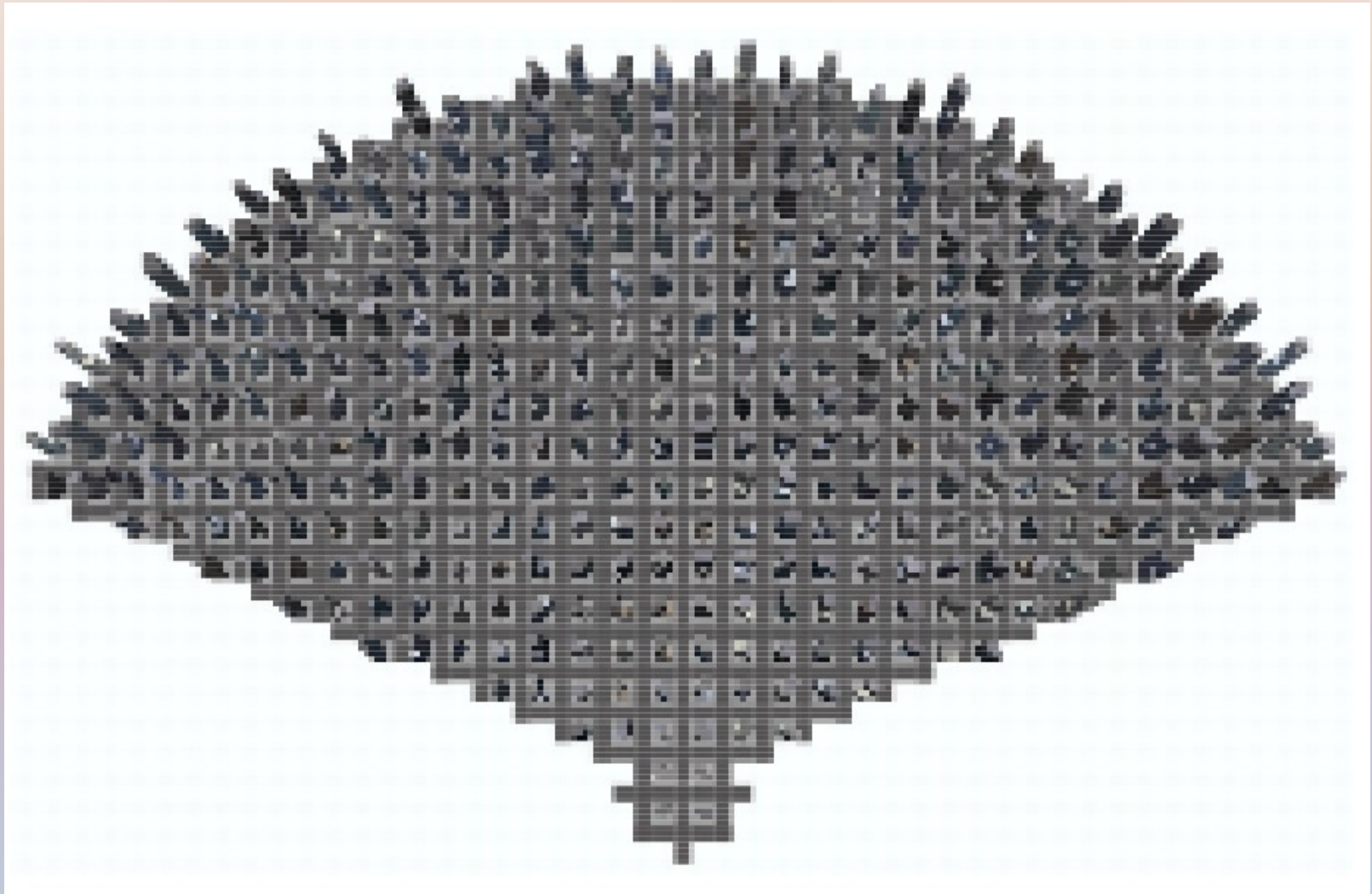
- *View Frustum Filling*:
  - Se utiliza para generar y navegar dentro de ciudades de dimensiones “infinitas”.
  - Basado en mallas uniformes.
  - Se genera solamente la parte de la ciudad que está dentro del frustum.
  - Para una misma ejecución la ciudad es determinística.
  - Se utiliza una caché de edificaciones con LRU como algoritmo de reemplazo.

# Técnicas de despliegue.

- La generación de la ciudad sigue estos pasos:
  - Se genera una semilla aleatoria para cada celda:
    - Se calcula una funcion hash de las coordenadas X y Z de la celda y se aplica un XOR con una semilla aleatoria de la ciudad.
  - Se usa la semilla de la celda para generar una serie pseudo aleatoria que determina su altura máxima y el proceso que se usará para generar el edificio.



# Técnicas de despliegue.



# Aplicaciones.

- Sistemas de información geográfica.
- Planificación urbana.
- Simulación en sismología.
- Cartografía.
- Escenarios para cine y juegos de video.



# Conclusiones.

- Los modelos de ciudades pueden ser reconstruidos a partir de datos reales:
  - Escaneo 3D.
  - Análisis de imágenes satelitales.
  - Ejemplo: La ciudad plegable de *Inception*.
- La complejidad de los modelos puede crecer muy rápidamente. Se necesita de algoritmos para cargar partes del modelo eficientemente junto a sus niveles de detalle.
- Suele haber apoyo de sistemas de bases de datos.

# Referencias

- Kelly, George y McCabe, Hugh: *A Survey of Procedural Techniques for City Generation*.
- Parish, Yoav I. H. y Müller, Pascal: *Procedural Modelling of Cities*.
- Müller, Pascal et al.: *Procedural Modelling of Buildings*.
- Greuter, Stefan et al.: *Real-Time Procedural Generation of 'Pseudo Infinite' Cities*.
- Beck, Michael: *Real-Time Visualization of big 3D City Models*.
- Bittner, Jirí et al.: *Coherent Hierarchical Culling: Hardware Occlusion Queries Made Useful*.