UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Departament d'Arquitectura de Computadors

Tarjetas Gráficas y Aceleradores

CUDA – Propuestas de Proyectos

Beatriz Otero

Departament d'Arquitectura de Computadors

Facultat d'Informàtica de Barcelona

Universitat Politècnica de Catalunya

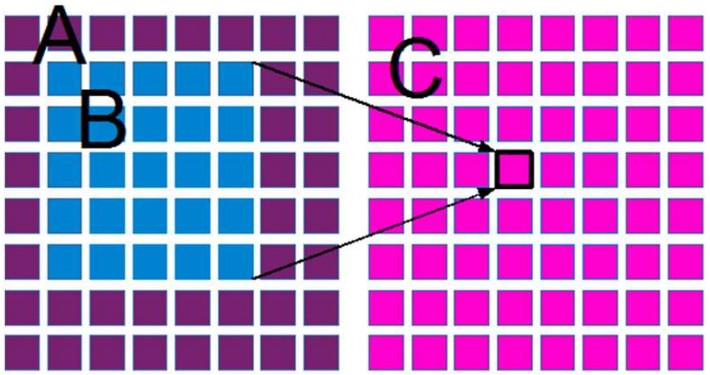
Imágenes borrosas: Matriz de convolución

Aplicar matriz de convolución B_{5x5} a cada posición de la matriz A_{NxN} para generar la matriz que contiene la imagen borrosa (matriz C_{NxN}).

- Se utiliza en el procesamiento de imágenes para mejorar/empeorar la imagen
- Dada una imagen (matriz A_{8x8}) aplicamos a cada elemento de la matriz A la matriz B (generada a partir de un filtro gaussiano) como se indica a continuación:

$$C_{[j][i]} = \sum_{m=0}^{4} \sum_{n=0}^{4} B_{[m][n]} \cdot A_{[i+m-2][j+n-2]} \qquad con\ 0 \le i \le 8\ y\ 0 \le j \le 8$$

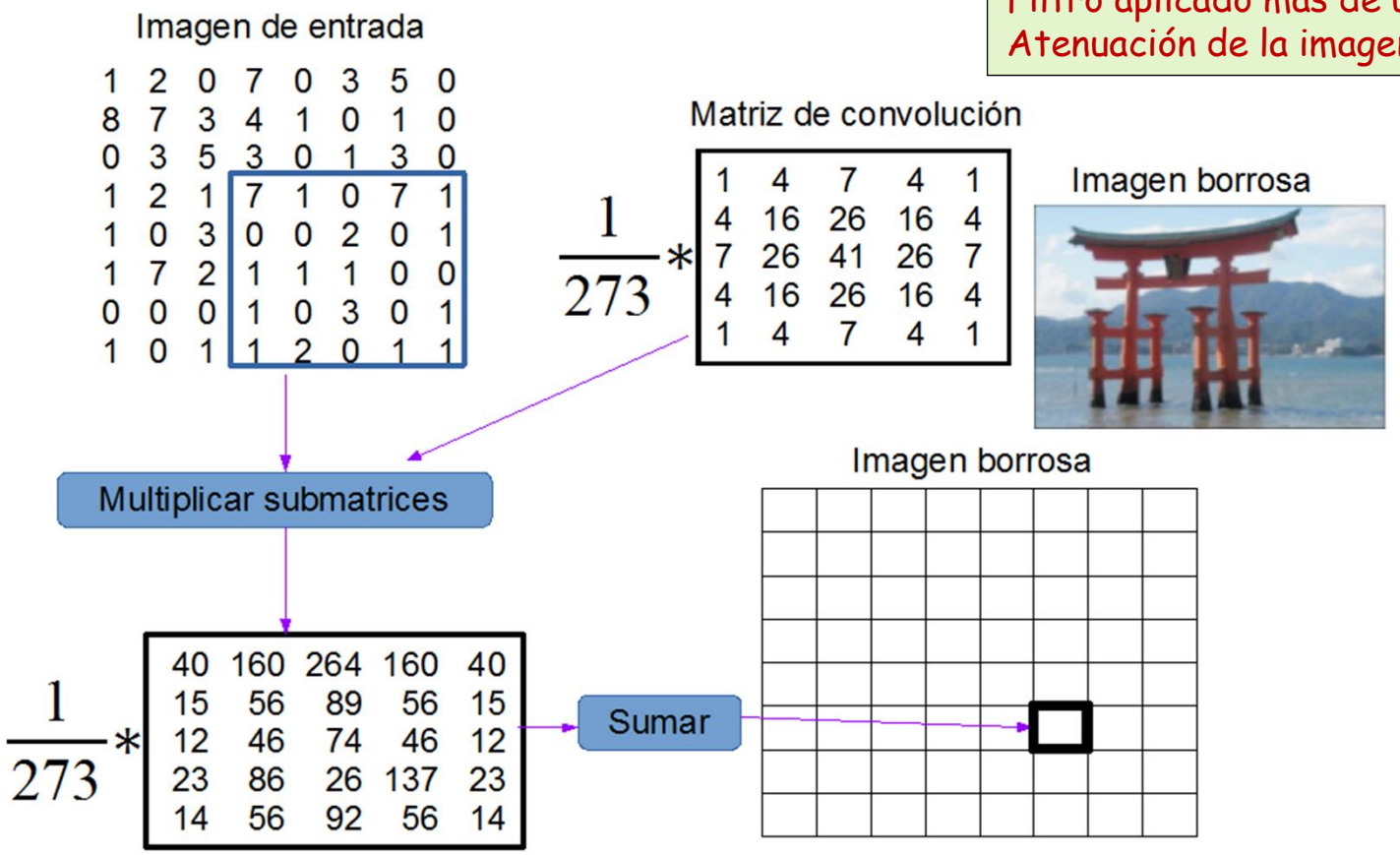
- En este caso obtenemos la matriz C_{8x8} que es la matriz borrosa de la matriz A_{8x8}.



Filtro Gaussiano:
- Usado para reducir el ruido en una imagen o quitar detalles relevantes.
- Usaremos un filtro gaussiano de 5 puntos.

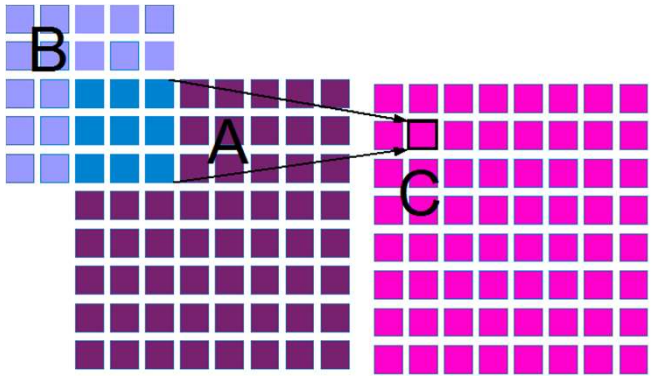
Ejemplo: Cómo generar imagen borrosa?

Filtro aplicado mas de una vez.
Atenuación de la imagen original.



Imágenes borrosas: Matriz de convolución

- Los elementos de la matriz B que queden fuera de la matriz A tienen valor 0.

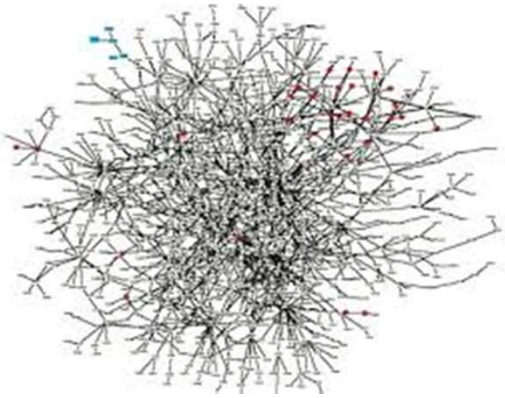


- ¿Qué hay que hacer en el proyecto de TGA?
  - Versión simple: 1 thread para calcular cada elemento de la matriz C
  - Versión mejorada que utilice la shared memory, trabaje por tiled y sincronice los threads en un bloque.
  - Para ambos versiones se requiere estudiar el rendimiento kernel para diferentes tamaños imágenes: 128x128, 512x512, 3072x3072, 4096x4096

Link a BD de imágenes: http://www.imageprocessingplace.com/root_files_V3/image_databases.htm
https://homepages.cae.wisc.edu/~ece533/images/

Centralidad en redes complejas

- Dado un grafo G, existen dos métricas importantes que se requieren calcular: la centralidad y la comunicabilidad de un vértice.
  - La centralidad de un vértice es una medida que indica lo importante que es el vértice en el grafo.
  - La comunicabilidad del vértice i es aproximadamente igual a la suma de todos los vértices con los que se comunica (misma fila).
- Las redes complejas se representan utilizando matrices de adyacencia, que describen la conexión de los vértices en el grafo. Generalmente es una matriz dispersa almacenada en el formato disperso Compressed Sparse Row format (CSR).
- Las redes complejas que analizaremos se derivan de los siguientes modelos de generación de grafos aleatorios: Barabási–Albert (conexión preferencial) y Watts–Strogatz (small-world).



Ejemplo Small-world:
Interacción proteína de la levadura

La centralidad de un vértice constituye la base del *PageRank* de Google ya que determina la relevancia de los documentos (o páginas web) indexados por este motor de búsqueda.

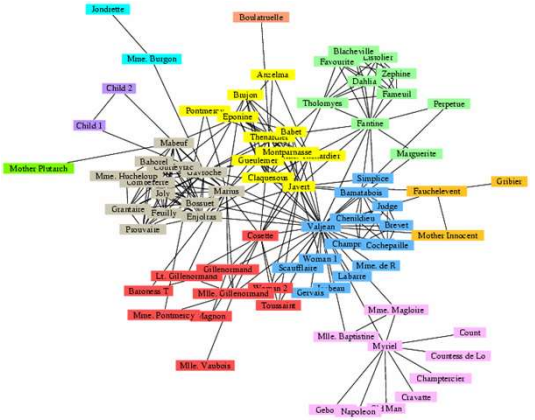
Centralidad en redes complejas: Algoritmo de Lanczos

- **La centralidad se determina al encontrar el autovector asociado al autovalor mas grande. El algoritmo de Lanczos aproxima los autovalores de una matriz.**
- **¿Qué hay que hacer en el proyecto de TGA?**
  - **NO** es necesario implementar **TODO** el algoritmo de Lanczos, simplemente realizar la implementación de algunas de sus operaciones matriciales tales como:
    - ✓ Cálculo de la norma L_2 de un vector
    - ✓ Producto matriz de adyacencia dispersa por vector
    - ✓ Producto de un escalar por un vector
    - ✓ Producto de un escalar por una matriz de adyacencia dispersa
    - ✓ Cálculo del error de 2 vectores (usar norma L_{inf})

TFG puede ser la implementación de todo el algoritmo de Lanczos y su evaluación en otras redes.

Centralidad en redes complejas: Monte Carlo

- Es un método determinístico que permite aproximar expresiones complejas costosas de determinar. Requiere el uso de un generador de números aleatorios.
- **¿Qué hay que hacer en el proyecto de TGA?**
  - Existe un código en C del algoritmo. Incorpora trabajo con matrices dispersas almacenadas en el formato CSR (Compressed Sparse Row format). El programa determina con bastante probabilidad cuál es el nodo con mayor centralidad en una red compleja y cuál es la comunicabilidad total de la red.
  - Los números aleatorios se deben generar en la GPU.
  - Sólo se requiere hacer el kernel en CUDA para determinar el vector de centralidades.



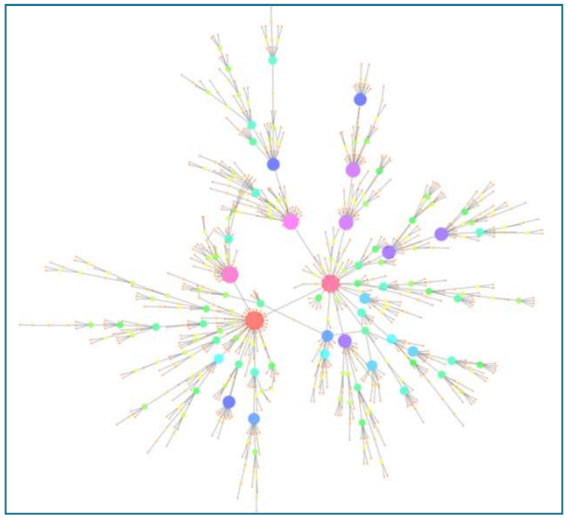
Ejemplo de Small-world:
Red social de los miserables

Aprenderás a utilizar funciones de la librería CUDART, entre otras y a trabajar con redes complejas de gran tamaño. Las redes utilizadas se proveen y están relacionadas con redes sociales.

Centralidad en redes complejas: Aproximación series

- La centralidad de los nodos de una red compleja puede aproximarse por los elementos diagonales de la matriz e^A obtenidos al aproximar esta función matricial por la siguiente serie:

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!}$$

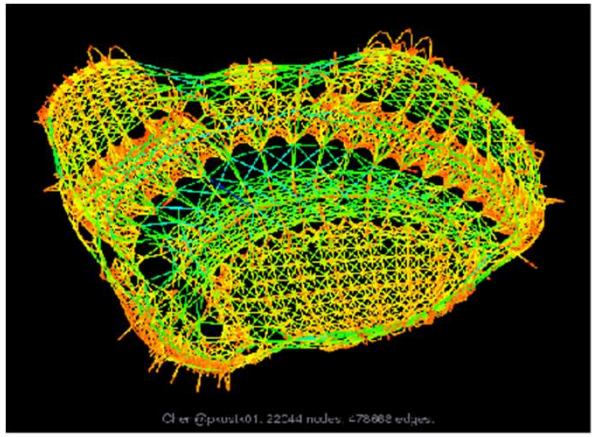


Extraído de Wikipedia: Red 1000 nodos generada por el Modelo de Barabási-Albert

- **¿Qué hay que hacer en el proyecto de TGA?**
  - Realizar el producto de dos matrices dispersas.
  - Sumar matrices dispersas.
  - Calcular el factorial de un número real.

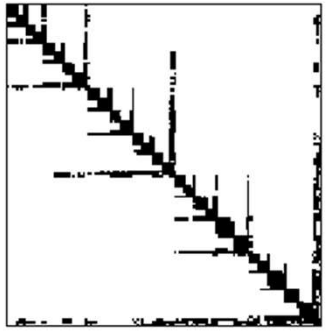
Formatos de almacenamiento: Matrices dispersas

- Matrices muy grandes con un gran porcentaje de elementos con valor 0.
- Se requiere una forma de almacenar estas matrices de forma de forma que garantice mínimos requerimientos de memoria y de cálculo.
- Existe una gran cantidad de formatos de almacenamiento, algunos de ellos ya han sido pensados para utilizar las prestaciones de la GPU. La librería CULA Sparse tiene implementados los formatos: CSR, CSC, Indexing, entre otros.
- Existen muchos problemas que requieren de estos formatos para almacenar los datos de la matriz asociada al problema que se requiere resolver.



Extraído de: http://www.cise.ufl.edu/research/sparse/matrices/Chen/pkustk01.html
Modelo del Conservatorio del Jardín Botánico de Beijing.

22044x22044 con 979380 elementos cero



Formatos de almacenamiento: Matrices dispersas

¿Qué hay que hacer en el proyecto de TGA?

- Implementar la operación matricial producto matriz por vector utilizando como mínimo 3 de los siguientes formatos de almacenamiento para la matriz A:
- Slice ELL, SELL-C, BRO-ELL, BRO-COO, BRO-HYB.
  - ✓ Revisar los siguientes artículos:
    - An architecture-aware technique for optimizing sparse matrix-vector multiplication on GPUs (2013). http://ac.els-cdn.com/S1877050913003396/1-s2.0-S1877050913003396-main.pdf?_tid=366bd536-1f90-11e7-b71e-00000aacb35d&acdnat=1492009170_cfba72344a716174fc23c7933a52c679
    - Implementing a Sparse Matrix Vector Product for the SELL-C/SELL-C-σ formats on NVIDIA GPUs (2014). http://www.icl.utk.edu/sites/icl/files/publications/2014/icl-utk-772-2014.pdf
    - Accelerating Sparse Matrix-Vector Multiplication on GPUs using Bit-Representation-Optimized Schemes (2013). https://www.comp.nus.edu.sg/~wongwf/papers/SC13.pdf
- Evaluar el rendimiento de la operación matricial en la GPU de cada formato de almacenamiento implementado. Comparar con los tiempos obtenidos en la CPU/GPU para diferentes tipos de matrices. Incluir métricas de rendimiento tales como: speedup, ancho de banda, tiempo de ejecución.

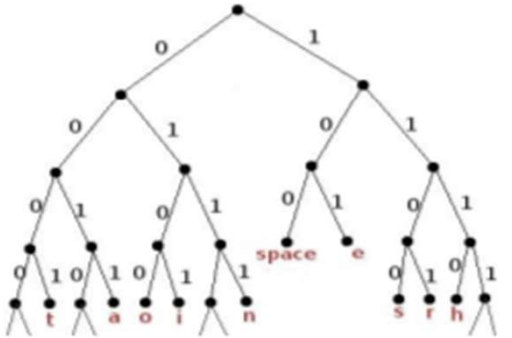
TFG puede ser la implementación de OTROS formatos y su evaluación en otras matrices.

Links a bibliotecas que contienen ejemplos de matrices dispersas:
http://math.nist.gov/MatrixMarket/
http://www.cise.ufl.edu/research/sparse/matrices/

Compresión de datos

Codificación Huffman (Compresión de imágenes)

- Cambiamos la representación de bits de cada byte con la esperanza de que el nuevo diccionario produce una cadena más corta de bits necesarios para almacenar la imagen. A diferencia de la representación ASCII de cada byte, nuestro nuevo diccionario podría utilizar representaciones de longitud de bits variables.
- La solución se basaba en el proceso de construir el árbol de abajo hacia arriba en vez de arriba hacia abajo. La idea de Huffman fue identificar los píxeles que aparecen con mayor frecuencia en una imagen y asignarles representaciones cortas. Los píxeles que ocurren con menor frecuencia en una imagen se les asigna representaciones largas.



J. Lezema. Codificación Huffman. Revista de educación matemática. Vol 32, No. 1, pp. 25-36, 2017
https://revistas.unc.edu.ar/index.php/REM/article/view/17191/16809

Compresión de datos

Formato JPEG (Compresión de fotografía)

- La compresión de una fotografía es la reducción de los datos digitales que no resultan necesarios e importantes. Esta compresión permite almacenar mayor número de imágenes al conseguir que los archivos resultantes no ocupen mucho espacio.

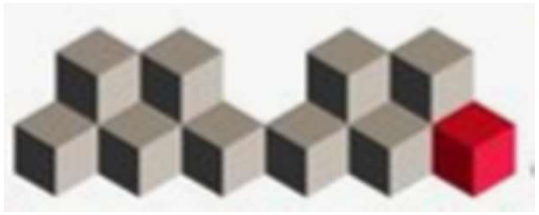
Imagen original. Imagen resultante.

J. Lezema. Compresión de imágenes: Formato JPEG. Revista de educación matemática. Vol 32, No. 2, pp. 23-34, 2017
https://revistas.unc.edu.ar/index.php/REM/article/view/18372/18232

Minería de criptomonedas

- Monedas virtuales que tienen capitalización en el mercado (Bitcoin (sistema de pagos P2P-2009), litecoin (2011), ripple (2013), dogecoin (2013), peercoin (2012))
- Minar criptomonedas es el proceso a través del cual las transacciones de criptomoneda se verifican y se ofrecen nuevas unidades.
  - El objetivo de los mineros es usar la potencia del PC y ponerlo al servicio de la red Bitcoin, para recopilar las últimas transacciones en bloques (es decir, conjuntos de transacciones verificadas). Haciendo esto se obtiene una recompensa: una cantidad fija de criptomoneda.

Minero agrupa transacciones nuevas de criptomoneda en un bloque



El bloque se codifica y se vincula a la cadena de bloques o *blockchain* existente.

El minero obtiene recompensa.

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Departament d'Arquitectura de Computadors

Tarjetas Gráficas y Aceleradores

CUDA – Propuestas de Proyectos

Beatriz Otero

Departament d'Arquitectura de Computadors

Facultat d'Informàtica de Barcelona

Universitat Politècnica de Catalunya