Tarea 2

CC5213 – Recuperación de Información Multimedia

2 de Octubre de 2017

El objetivo de esta tarea es analizar la relación entre efectividad y eficiencia para distintos índices al resolver búsquedas aproximadas del vecino más cercano.

Para distintos índices y conjuntos de descriptores debe resolver un **Similarity Join del Vecino Más Cercano Aproximado**, midiendo la efectividad y eficiencia lograda según distintos niveles de aproximación de la búsqueda. Los índices que deberá evaluar son:

- **Randomized KD-Tree** para distintos números de árboles (`trees`).
- **K-Means Tree** para distintos números de centroides por nivel (`branching`).
- **Linear Scan** o algoritmo de fuerza bruta que se utilizará como línea base de efectividad y eficiencia.

Puede encontrar una implementación de estos índices en la librería FLANN (*Fast Library for Approximate Nearest Neighbors*). Esta librería está incluida en OpenCV aunque se recomienda utilizar el último release disponible en el repositorio GitHub[1].

Un experimento consistirá en seleccionar un par de conjuntos de vectores Q y R, construir un índice con los vectores de R y resolver $|Q|$ búsquedas del vecino más cercano usando cierto valor para el parámetro de aproximación (`checks`). Para cada experimento definiremos los siguientes indicadores:

- **Efectividad**: La fracción de las consultas cuyo vecino más cercano encontrado coincide con el encontrado por Linear Scan.
- **Eficiencia**: La fracción del tiempo utilizado en resolver las búsquedas en comparación con el tiempo requerido por Linear Scan.

Ejemplo: Dado un par de conjuntos Q y R suponga que $|Q|=500$ y que Linear Scan demora 30 segundos en resolver las 500 búsquedas del NN. Si un índice (para cierto parámetro de aproximación) resuelve las mismas búsquedas en 12 segundos y en 400 de ellas obtiene el mismo resultado que Linear Scan entonces obtiene una efectividad de 0.8 y eficiencia de 0.4.

Notar que por definición Linear Scan obtiene efectividad 1 y eficiencia 1, y que el óptimo ideal corresponde a efectividad 1 y eficiencia 0.

Calculando la efectividad y eficiencia obtenida por un índice al usar distintos valores de aproximación en la búsqueda se puede obtener la **curva de efectividad versus eficiencia** del índice. Con las curvas obtenidas por todos los distintos índices

[1] https://github.com/mariusmuja/flann

evaluados elabore un **gráfico de efectividad versus eficiencia** para el par de conjuntos *Q* y *R*.

Las curvas de efectividad versus eficiencia dependen de los conjuntos *Q* y *R* utilizados en los experimentos. Encontrará distintos conjuntos de vectores para realizar la evaluación en la dirección:

http://juan.cl/cc5213-2017/

Estos conjuntos de vectores corresponden a descriptores visuales y acústicos obtenidos a partir de los videos de televisión utilizados en la Tarea 1. Escoja varios pares de conjuntos *Q* y *R* de entre los descriptores ahí publicados.

Con los resultados obtenidos de la comparación de efectividad y eficiencia de los distintos índices y descriptores deberá elaborar un informe en PDF con formato de paper[2]. El informe debe contener Introducción, Descripción de los algoritmos a evaluar, Experimentos (incluyendo los gráficos de efectividad versus eficiencia sobre los distintos descriptores), Conclusiones y Referencias. Considere las siguientes preguntas en la discusión de resultados:

- ¿Qué índice obtiene la mejor performance? ¿Con qué parámetros (`trees`, `branching`)?
- ¿Depende la performance de los índices del descriptor usado? ¿depende del número de dimensiones?
- ¿Qué sucede con la performance de los índices cuando se desea maximizar la efectividad o la eficiencia?
- ¿Puede predecir la performance que obtendrá un índice antes de realizar las búsquedas?

Entrega:

- Debe entregar todos los códigos fuentes, el informe en PDF y un archivo `readme.txt` con las instrucciones de compilación y ejecución de la tarea. No incluya datos de prueba.
- La ejecución de su tarea debe generar los mismos resultados que están en el informe.
- El plazo máximo de entrega es el **Martes 17 de Octubre a las 23:59** por U-Cursos. No se aceptarán tareas atrasadas.
- La tarea es ***individual***.
- La tarea la puede implementar en alguno de los lenguajes soportados por FLANN: Python o C++.
- Se evaluará el orden del código fuente, la claridad del archivo `readme.txt`, los resultados de los experimentos y las conclusiones obtenidas.

[2] http://www.acm.org/publications/proceedings-template-16dec2016