Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computación
Curso: Análisis de Algoritmos
Profesor: José Carranza-Rojas
Valor: 10%
Proyecto en parejas

Semestre I, 2017

Proyecto 1 – Van Gogh Evolucional

Figure 1 - Si Van Gogh hubiese conocido a Hayao Miyazaki

Vincent Van Gogh fue un pintor holandés de alto renombre y de las figuras más incónicas del post-modernismo y de la historia del arte occidental. Sufrió de constantes ataques psicóticos y problemas mentales, problemas de depresión e irónicamente durante su vida no tuvo éxito. Finalmente decidió suicidarse a sus apenas 37 años disparándose en el pecho.

Inspirandos en sus delirios, definimos este proyecto que consiste en crear un algoritmo genético parametrizable que aproxime paulatinamente una imagen meta en tonos de grises, comenzando con una generación inicial formada por imágenes con pixels aleatorizados. Inicialmente las imágenes se verán totalmente sin sentido pero paulatinamente se aproximarán a la imagen meta dada. Deberá ser programado en Python.

La población inicial consiste en una serie de k imágenes generadas aleatoriamente. Eventualmente cada nueva generación consiste en una serie de k individuos que van teniendo un indice mayor de similitud con la imagen meta. El índice de similitud en este caso provee una forma de valorar a los individuos más aptos. Los genes de cada individuo (imagen candidata) consisten en sus pixels. En otras palabras, el índice de similitud entre imágenes es lo mismo que la función de adaptabilidad del algoritmo genético. Se incluirá un documento a estilo paper con el análisis algoritmico de varias funciones de similitud así como experimentos que empíricamente muestran el comportamiento de los algoritmos genéticos usando las distintas funciones de similitud.

Funciones de Adaptabilidiad

Se tendrán que programar 3 funciones de adaptabilidad (y analizar dos), que en este caso son funciones de similitud entre imágenes, que se calculan entre cada imagen candidata y la imagen meta, para obtener a los individuos más y menos aptos.

1. Similitud Euclideana: Una de las más sencillas y básicas formulas para comparar dos vectores. La ecuación está dada por:

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

   Donde p y q son dos vectores n-dimensionales de tamaño n (en el caso de las imágenes sería de tamaño m x n, filas x columnas). Tipicamente este algoritmo es O(m x n).
2. Otro algoritmo existente de similitud: deben buscar algún algoritmo existente de similitud que sea útil entre imágenes. Deben programarlo y además sacar su complejidad algorítmica usando O grande. Deben

incluir estos detalles en el documento (escrito en latex) junto a las referencias respectivas de el o los papers donde encontraron el algoritmo. Los detalles del documento se espcifican en una sección posterior de este documento.

3. Su propio algoritmo de similitud: tendrán que pensar en su propio algoritmo de similitud entre imágenes, programarlo, calcular su complejidad algoritmica en O grande, y lógicamente describirlo en el documento en latex.

Documento en Latex / PDF

Deben crear en latex (y entregar los fuentes de latex) un documento donde describen el análisis de los distintos algoritmos de simillitud implementados (la euclideana no hace falta). El documento debe seguir el template de la IEEE para publicaciones en ingeniería el cual pueden encontrar aquí:

https://www.ieee.org/documents/IEEEtran.zipLlamada Inicial del Programa

El documento además tendrá un análisis comparativo (experimentos) entre las disintas implementaciones de funciones de adaptabilidad. Éste debe hacerse respecto a número de generaciones y porcentaje de similitud alcanzada por el individuo más apto de cada generación. Idealmente debería verse como cada generación incrementa el índice de similitud respecto a la imagen meta y se podría observar cúal algoritmo tiende a dar los mejores resultados y después de cuántas generaciones eso sucede. Se recomienda usar matlibplot para generar los gráficos.

Parámetros del Programa

Se debe paramatrizar lo siguiente:

- Tamaño de población.
- Probabilidad de cruce entre dos individuos.
- Porcentaje de genes a mutar.
- Porcentaje de individuos menos aptos a mantener para garantizar variabilidad y evitar poblaciones degeneradas.
- Ruta de la imagen meta en disco.

Salida del Programa

La salida es una imagen grande que tiene concatenadas las mejores imágenes candidatas (más aptas) de generaciones que están en mútiplos de 10% (empezando de 0%) de todas las generaciones. De esta manera siempre se tendrá una imagen formada por 10 imágenes, donde la primera de la izquierda es puro ruido, y se puede ir viendo como poco a poco se empiezan a acercar a la imagen meta. La última de la derecha debería ser muy parecida a la imagen meta.

Librerías

Instituto Tecnológico de Costa Rica Semestre I, 2017
Escuela de Ingeniería en Computación
Curso: Análisis de Algoritmos
Profesor: José Carranza-Rojas
Valor: 10%
Proyecto en parejas

Se recomienda usar varias librerías de Python para agilizar el proceso de desarrollo del software:

1. Numpy: librería para manipulación de vectores n-dimencionales. Permite manipular imágenes como matrices, por ejemplo.
2. PIL: librería que se puede usar para leer y escribir imágenes, y es compatible con numpy.
3. Matplotlib: Librería útil para generar gráficos y para unir varias fotos en una misma (aunque esto se puede lograr con numpy también).

Puntos Extra

Se darán puntos extra si se usan fotos a color y el análisis incluye formalmente el hecho de que hayan 3 canales de color.

Evaluación

Tarea	Puntaje Máximo
Programación de Algoritmo Genético, con cruces, mutaciones, y demás	45
Imagen compuesta resultante (que se vea el proceso evolutivo)	10
Distancia Euclideana (Programación)	5
Otra distancia existente (Programación)	5
Analisis de otra distancia existente	5
Distancia propia inventada (Programación)	10
Análisis de distancia propia	10
Experimentos y discusión	10
Manejo de color (programado y analizado)	5