

Extractor de subtítulos incrustados: Trabajo Final de Procesamiento Digital de Imágenes

María A. Paez, Leonardo N. Arato y Darién J. Ramírez

Trabajo práctico final de “Procesamiento Digital de imágenes”, II-FICH-UNL.

Resumen—En este trabajo se propone un método automático para la extracción de subtítulos incrustados en videos utilizando técnicas del procesamiento digital de imágenes. El proceso consiste en primer lugar en analizar los fotogramas del archivo de entrada y determinar si cuentan o no con la presencia de subtítulos. Con esa información se establecen los tiempos para crear un archivo *srt*. Finalmente se utiliza el reconocedor óptico de caracteres *tesseract OCR* para obtener una aproximación al texto original incrustado en el fotograma.

Palabras clave—extractor, subtítulos, *srt*, *tesseract*, OCR.

I. INTRODUCCIÓN

ES común encontrar videos con subtítulos incrustados los cuales se caracterizan por estar pegados sobre la imagen que representa al fotograma y no pueden ser desactivados, es decir, forman parte de la imagen y no se encuentran en formato texto. Se dan situaciones donde se posee un archivo de video con contenido remasterizado (mejorado en cuanto a calidad y resolución) pero el cual carece de subtítulos, sin embargo, se dispone de la versión original (no remasterizada e incluso con una baja calidad) la cual se encuentra subtitulada pero teniendo los subtítulos pegados impidiendo la posibilidad de utilizarlos en la versión remasterizada. El propósito de este trabajo es realizar la conversión *imagen-texto* de manera automática para crear un archivo de subtítulos que sea reutilizable facilitando el tedioso trabajo de establecer los tiempos de los subtítulos de manera manual.

II. DETECCIÓN DE SUBTÍTULOS

Se analizan los fotogramas del video para determinar la presencia de subtítulos en cada uno de ellos. La región de interés para la búsqueda es el cuarto inferior de la imagen.

$$\frac{Q}{f_1(p^k(t))} + \frac{Q}{f_2(p^k(t))} + \dots + \frac{Q}{f_n(p^k(t))}$$

III. MATRICES DE RELACIÓN

Las matrices de relación entre ciudades contienen los costos de las diferentes características de evaluación de caminos. Para este trabajo en particular se definieron cuatro matrices: distancias, peajes, tráfico y hospedaje. A nivel general estas matrices contienen información correspondiente a las aristas de un grafo, es decir, el costo representado en la celda i, j de las matrices representa el costo de ir de la ciudad A a la ciudad B y como son simétricas la celda j, i representa el costo de ir de la ciudad B a la ciudad A .

Existen ciudades que no tienen una conexión física entre ellas. En estos casos la celda i, j de las matrices tienen un valor no válido de costo. Esto es para que las hormigas no puedan optar por tomar un camino inexistente entre ciudades y se adapten a la estructura del problema planteado y representado por las matrices de relación. Por

```

1  $t = 0, \sigma_{ij}(t) \leftarrow U(0, \sigma_0)$ 
2 Ubicar  $N$  hormigas en el nodo origen
3 repetir
4   para cada hormiga  $k = 1, 2, \dots, N$ 
5      $p^k(t) = \emptyset$ 
6     repetir
7       • seleccionar el próximo nodo según probabilidad
          
$$p_{ij}^k(t) = \begin{cases} \frac{\sigma_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{u \in N_i^k} \sigma_{ui}^\alpha(t) \eta_{ij}^\beta(t)} & ; j \in N_i^k \\ 0 & ; e.o.c. \end{cases}$$

8       • agregar un paso  $(i, j)$  al camino  $p^k(t)$ 
9       hasta alcanzar el destino
10      calcular la longitud del camino encontrado  $f(p^k(t))$ 
11    para cada conexión  $(i, j)$ 
12      • reducir por evaporación la cantidad de feromonas:  $\sigma_{ij}(t) \leftarrow (1 - \rho)\sigma_{ij}(t)$ 
13      • depositar feromonas proporcionalmente a la bondad de la solución
          
$$\Delta\sigma_{ij}^k(t) = \begin{cases} \frac{Q}{f(p^k(t))} & \text{global} \\ \frac{Q}{d_{ij}} & \text{uniforme} \\ \frac{Q}{d_{ij}} & \text{local} \end{cases}$$

          
$$\sigma_{ij}(t+1) = \sigma_{ij}(t) + \sum_{k/(i,j) \in p^k(t)} \Delta\sigma_{ij}^k(t)$$

14 hasta que todas las hormigas sigan el mismo camino
15 devolver el mejor camino

```

Fig. 1. Algoritmo: Sistema de Hormigas

otra parte, la diagonal de las matrices son nulas puesto que la distancia de una ciudad a si misma es nula. Aun así dado el funcionamiento del algoritmo, si se está parado en una ciudad, no se elegirá esa misma ciudad como posible camino.

A continuación se muestran ejemplos de matrices de relación de 3×3 para visualizar mejor su estructura:

$$D = \begin{matrix} & \begin{matrix} Santa Fe & Esperanza & Recreo \end{matrix} \\ \begin{matrix} Santa Fe \\ Esperanza \\ Recreo \end{matrix} & \begin{pmatrix} 0 & -1 & 17 \\ -1 & 0 & 22 \\ 17 & 22 & 0 \end{pmatrix} \end{matrix}$$

Fig. 2. Matriz de distancias

La fila 1 de D (Figura 2) se lee como:

- Santa Fe tiene distancia nula consigo misma.
- No hay una ruta que conecte Santa Fe con Esperanza.
- Santa Fe tiene una ruta que la conecta a Recreo cuya distancia es de 17 km. También es posible verlo como que el costo para viajar desde Santa Fe a Recreo o desde Recreo a Santa Fe es 17.

Los posibles valores para la matriz de peajes (Figura 3) (sin contar la diagonal y las posiciones inválidas) son los números $0, 1, \dots, 5$. El cero indica no hay una estación de

$$P = \begin{matrix} & \begin{matrix} Santa Fe & Esperanza & Recreo \end{matrix} \\ \begin{matrix} Santa Fe \\ Esperanza \\ Recreo \end{matrix} & \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 2 \\ 0 & 2 & 0 \end{pmatrix} \end{matrix}$$

Fig. 3. Matriz de peajes

peaje en la ruta que une esas dos ciudades determinadas. Los números restantes no indican el precio del peaje si no un indicador de costo.

- 1: Muy económico.
- 2: Económico.
- 3: Intermedio.
- 4: Costoso.
- 5: Muy costoso.

$$T = \begin{matrix} & \begin{matrix} Santa Fe & Esperanza & Recreo \end{matrix} \\ \begin{matrix} Santa Fe \\ Esperanza \\ Recreo \end{matrix} & \begin{pmatrix} 0 & -1 & 27 \\ -1 & 0 & 27 \\ 27 & 27 & 0 \end{pmatrix} \end{matrix}$$

Fig. 4. Matriz de tránsito

Los valores en la matriz de tránsito (Figura 4) indican una cantidad de minutos aproximada en la que es posible recorrer la distancia teniendo en cuenta el tráfico habitual de la ruta.

$$H = \begin{matrix} & \begin{matrix} Santa Fe & Esperanza & Recreo \end{matrix} \\ \begin{matrix} Santa Fe \\ Esperanza \\ Recreo \end{matrix} & \begin{pmatrix} 0 & -1 & 9 \\ -1 & 0 & 1 \\ 9 & 1 & 0 \end{pmatrix} \end{matrix}$$

Fig. 5. Matriz de hospedaje

Los valores en la matriz de hospedaje (Figura 5) indican la cantidad de lugares con alojamiento que se encuentran en el camino entre las dos ciudades.

Todas las matrices de relación tienen en común que son simétricas, la diagonal es nula y que las posiciones inválidas se corresponden, el resto de las celdas son valores propios del tipo de matriz en cuestión.

Entonces el deseo de moverse de la hormiga ya no será sólo la inversa de la distancia si no que será la suma de las inversas de cada característica aplicada. Lo mismo ocurre a la hora de depositar feromonas en base a la bondad de la solución.

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) + \sum_{\forall k/(i,j) \in p^k(t)} \Delta\sigma_{ij}^k(t)$$

donde

$$\Delta\sigma_{ij}^k(t) = \frac{Q}{f_D(p^k(t))} + \frac{Q}{f_P(p^k(t))} + \frac{Q}{f_T(p^k(t))} + \frac{Q}{f_H(p^k(t))}$$

Como se puede notar las proporciones de éstas matrices son muy diferentes entre sí. Por lo tanto, para poder realizar una comparación válida entre ellas a la hora de hacer sus

aportes a la matriz de feromonas, serán normalizadas. Para esto se toma el valor máximo de cada matriz y se las divide por dicho valor obteniendo así matrices normalizadas con valores que van entre cero y uno.

IV. ALGORITMO GENÉTICO

Además del algoritmo de hormigas también se ha desarrollado un algoritmo genético para encontrar el camino óptimo entre dos ciudades de modo de poder hacer una comparación entre los resultados arrojados por ellos.

```

1 inicializar(población)
2 mejor_fitness=evaluar(población)
3 mientras (mejor_fitness < fitness_requerido)
4     selección=seleccionar(población)
5     población=cruciarymutar(selección)
6     mejor_fitness=evaluar(población)
7 fin_mientras
    
```

Fig. 6. Algoritmo genético

Para un ejemplo donde se consta de 7 ciudades el material genético se encuentra codificado de la siguiente manera:

fenotipo 5 → 1 → 4 → 2 → 3
 genotipo 101 → 001 → 100 → 010 → 011

Cada número en el fenotipo está asociado a una ciudad. Su codificación genética consiste en la representación binaria del número que se asocia a la ciudad. El primer elemento representa la ciudad de origen y el último elemento representa la ciudad de destino. Los elementos intermedios representan las ciudades por las que hay que pasar para llegar del origen al destino.

Para determinar el *fitness* se convierte el genotipo a fenotipo (Tabla I), se mide y suma la distancia entre la ciudades y finalmente se calcula la inversa del valor del camino.

Fenotipo	0	1	2	3	4	5	6	7
Genotipo	000	001	010	011	100	101	110	111

TABLA I

CONVERSIÓN DE GENOTIPO A FENOTIPO.

El método de selección usado es el método de ventanas. Se ordenan los individuos según su *fitness* (de mejor a peor) y se van definiendo ventanas cada vez más pequeñas sacando un individuo (progenitor) al azar para cada una. Las ventanas van dejando de lado a los peores individuos.

Para realizar las mutaciones no se tocan los bits más significativos correspondientes al origen ni los menos significativos correspondientes al destino, se toman los bits de los destinos intermedios al azar. Existe una probabilidad *p* de que exista una mutación. De los bits del destino intermedio seleccionado se modifica un bit al azar y se cambia su valor (cero por uno o uno por cero). Tras la mutación se debe comprobar si la ciudad generada es válida revisando las conexiones con el nodo anterior y el siguiente. Si no lo es, se vuelve a repetir el proceso de mutación hasta generar una válida.

Para realizar la cruce se elige una ciudad al azar entre las intermedias como punto de cruce. Entonces el material genético anterior a ese punto de cruce de uno de los padres se combina con el material genético del otro padre a partir

e incluyendo el valor del punto de cruza. Si el camino no es válido se vuelve a repetir la cruza hasta obtener uno válido.

$$101001 \overset{pc}{100} 010011 \rightarrow 101001001010011$$

$$101 \overset{pc}{001} 010011 \rightarrow 101100010011$$

El tamaño de la población es constante pero el tamaño de los genotipos es variable puesto que depende de la longitud del camino. El método de remplazo utilizado durante la reproducción es el elitismo que implica agregar a la siguiente generación al individuo con mejor *fitness* de la generación actual.

V. MÉTODO DE BÚSQUEDA COSTO UNIFORME

Este método ha sido implementado, así como el algoritmo genético, para realizar comparaciones de resultados con el algoritmo de las hormigas.

En el método de búsqueda costo uniforme todos los nodos de profundidad d en el árbol de búsqueda se expanden antes que los nodos de profundidad $d + 1$. Además esa expansión es siempre primero para el nodo de menor costo, medido por una función que evalúa el *costo del camino* $g(n)$. Su implementación es una lista ordenada de menor a mayor costo. Cuando se verifican ciertas condiciones se garantiza que la primer solución que se encuentra es la de mínimo costo. La restricción es que el costo de un camino nunca debe decrecer al avanzar en su desarrollo.

VI. RESULTADOS

En la tabla II se muestran los tiempos de ejecución en segundos obtenidos con los tres algoritmos planteados para distinta cantidad de ciudades. El algoritmo de hormigas fue ejecutado con 7 hormigas en cada caso y el coeficiente de evaporación ρ fue fijado en 0,8. Al algoritmo genético se le estableció una población de 8 individuos en todos los casos y fue ejecutado hasta alcanzar las 10 generaciones. Los tiempos se corresponden a una sola ejecución del algoritmo y sólo se ha tenido en cuenta la distancia.

Dimensión de la matriz	Hormigas	Genético	Costo Uniforme
7×7	0.36	2.72	0.32
10×10	1.13	5.31	3.72
15×15	6.45	9.71	35.33
20×20	13.11	18.7	243.42

TABLA II
COMPARATIVA DE TIEMPOS DE EJECUCIÓN.

La figura ?? muestra una gráfica de como va aumentando el tiempo de ejecución al aumentar la cantidad de ciudades involucradas.

A continuación se comparan los porcentajes de aciertos entre el algoritmo de hormigas y genético para 50 iteraciones en una matriz de 7×7 (desde Rincón a Esperanza) con $\rho = 0,8$ y 7 hormigas. Para el algoritmo genético se establece una población de 8 individuos con 10 generaciones (Tabla III).

Los tiempos de ejecución obtenidos en la tabla III considerando sólo la distancia fueron de 17,03 segundos y 122,65 segundos respectivamente. Cabe aclarar que el resultado real del algoritmo, tanto de hormigas como

Matrices involucradas	Hormigas	Genético
Distancia	70	62
Distancia + Peaje	62	58
Distancia + Peaje + Hospedaje	64	54
Distancia + Peaje + Hospedaje + Tráfico	60	50

TABLA III
PORCENTAJES DE ACIERTOS DEL CAMINO ÓPTIMO EN CUANTO A LAS CARACTERÍSTICAS.

genético, es aquel camino que mayor ocurrencias tuvo.

La tabla IV muestra los porcentajes de acierto del camino óptimo variando la cantidad de hormigas, el coeficiente de evaporación y las características involucradas.

Matrices involucradas	7 Hormigas		14 Hormigas	
	$\rho = 0,4$	$\rho = 0,8$	$\rho = 0,4$	$\rho = 0,8$
D	54	70	72	76
D + P	62	76	80	88
D + P + H	60	72	74	78
D + P + H + T	64	76	76	82

TABLA IV
PORCENTAJES DE ACIERTOS DEL CAMINO ÓPTIMO PARA DISTINTOS VALORES DE N Y ρ .

Los tiempos de ejecución para cada combinación de características han sido similares tomando valores aproximados de 18, 71, 36 y 112 segundos respectivamente.

VII. CONCLUSIONES

El algoritmo de costo uniforme para una representación pequeña obtiene el camino optimo en el menor tiempo pero para una mayor representación se hace cada vez mas pesado y su alcance queda limitado a la potencia del hardware disponible. De todas formas, dada la estructura de las matrices de relación, que no debe recorrer todos los caminos si no sólo donde existe un camino posible, hace que sea una buena opción para la búsqueda del camino óptimo.

El algoritmo genético obtiene el camino optimo en un tiempo intermedio y este no depende del tamaño de la representación. A veces se obtienen caminos con bucles pero eso es algo que puede ser solucionado.

El algoritmo de hormigas obtiene resultados intermedios pero los tiempos de ejecución no se disparan tanto conforme aumenta la cantidad de ciudades pero esto es relativo a como se establezca la cantidad de hormigas y el coeficiente de evaporación de feromonas.

Algo común tanto en el algoritmo de hormigas como en el genético es que a medida que incrementa el número de características de evaluación, los porcentajes de acierto del camino óptimo disminuyen. Se puede inferir que demasiada información para determinar un camino hace que los resultados sean peores. Por esto es recomendable no abusar de la cantidad de características y sólo elegir aquellas fundamentales a la hora de determinar la ruta.

Una posible mejora para el futuro es cambiar la forma de representación de los datos de entrada. Levantar datos de una base de datos de mapas, o procesar una imagen para detectar nodos y distancias entre ellos.

REFERENCIAS

- [1] Rafael C. González and Richard E. Woods, *Digital Image Processing*. Prentice-Hall, 3rd. ed. (2008) - 2nd. ed. (2001).