

Vigilando el museo

JOSÉ MANUEL DÍAZ MORENO

Departamento de Matemáticas, Universidad de Cádiz
CASEM, Campus del Río San Pedro
11510 Puerto Real, Cádiz, España
josemanuel.diaz@uca.es

1. EL PROBLEMA

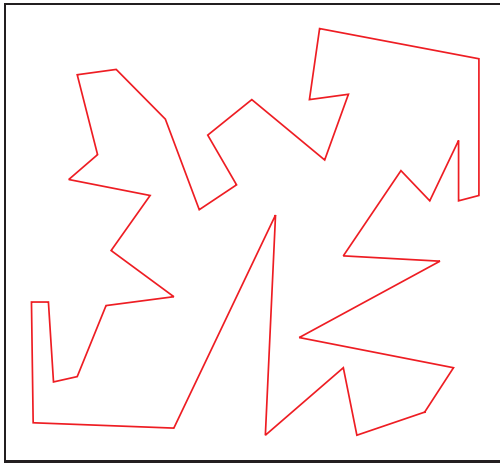


Figura 1: Planta de un museo

Para construir una aplicación computacional que resuelva el problema se necesita, en primer lugar, programar algunas funciones geométricas simples. Con este artículo se incluye un archivo con la codificación en *Mathematica* y un *Notebook* de ejemplo que el lector interesado puede descargarse).

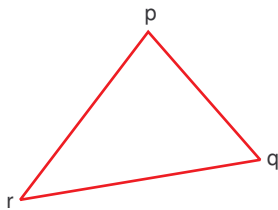
Consideremos un museo con planta poligonal simple de n lados (las paredes). La figura 1, a la izquierda, es un ejemplo –algo extravagante, desde luego– tomado de [6]. Se intuye inmediatamente que situando una cámara de vigilancia en cada esquina (vértice), el museo estará completamente vigilado. Pero ¿se puede hacer con un número menor de cámaras?

El *teorema de la galería de arte* establece que siempre hay alguna forma de situar $\lfloor n/3 \rfloor$ cámaras de manera que cualquier punto de la galería esté a la vista de al menos una de ellas.

La primera demostración de este elegante teorema se debe a V. Chvátal (1975) y una prueba posterior debida a S. Fisk permite, como una bella aplicación de la geometría computacional, la construcción de un algoritmo eficiente para situar las cámaras de vigilancia.

2. ALGUNOS PROBLEMAS SIMPLES

2.1. Dado un triángulo, determinar su orientación



Computacionalmente, un triángulo se representa mediante una *lista ordenada* de tres puntos del plano (los vértices):

$$T = (p, q, r).$$

Cuando los vértices se dan en el sentido inverso al movimiento de las agujas del reloj, decimos que el triángulo T tiene orientación positiva (+1); en caso contrario, T tiene orientación negativa (-1). Es importante observar, no obstante, que la orientación no es una propiedad intrínseca del triángulo, sino una propiedad de su representación computacional como lista *ordenada*.

Matemáticamente, la orientación de un triángulo (p, q, r) puede determinarse hallando el determinante

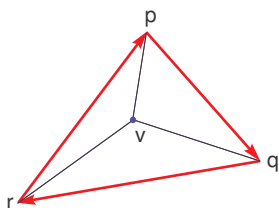
$$\begin{vmatrix} p_1 & p_2 & 1 \\ q_1 & q_2 & 1 \\ r_1 & r_2 & 1 \end{vmatrix},$$

y determinando su signo. Así, pues, la orientación del triángulo $T = (p, q, r)$ viene dada por

$$\text{signo} \left(\begin{vmatrix} p_1 & p_2 & 1 \\ q_1 & q_2 & 1 \\ r_1 & r_2 & 1 \end{vmatrix} \right).$$

(Los detalles de porqué esto funciona se dejan al lector).

2.2. Situación relativa de un punto v respecto a un triángulo



Dados dos puntos, p y q , denotamos por $R = (p, q)$ la recta *dirigida* desde p a q . Es inmediato que si la orientación del triángulo $T = (p, q, v)$ es positiva, entonces el punto v está a la *izquierda* de R .

Como consecuencia, si el triángulo $T = (p, q, r)$ tiene orientación positiva, un punto v es interior a T si está a la izquierda de la recta (p, q) , está a la izquierda de la recta (r, q) y está a la izquierda de la recta (p, r) .

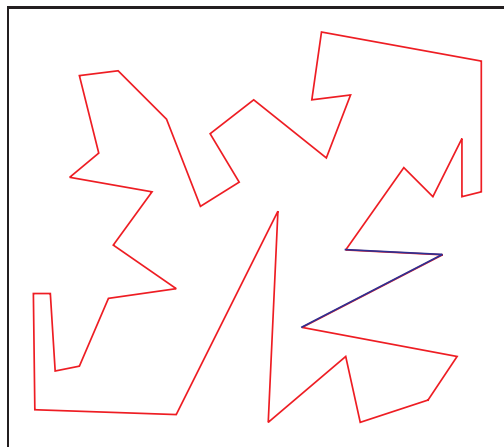
El inteligente lector comprenderá que, *mutatis mutandi*, el procedimiento puede aplicarse también cuando el triángulo tiene orientación negativa.

2.3. Dado un polígono simple, hallar un vértice convexo.

De modo similar a la representación de un triángulo, un polígono simple se representa computacionalmente mediante una *lista ordenada* de vértices

$$P = (p_1, p_2, \dots, p_n).$$

Que cualquier polígono simple, P , tiene al menos un vértice convexo, es decir, un vértice para el cual, el ángulo interior es menor que π , es inmediato (¿por qué?).



Lo que ya es menos evidente es cómo determinar algorítmicamente un vértice convexo de P . Desde luego, para ello se pueden usar las herramientas tradicionales de la geometría, pero el procedimiento siguiente es, a mi juicio, mucho más ingenioso y eficiente.

Dado un polígono simple $P = (p_1, p_2, \dots, p_n)$, si

$$p_i \leq p_k, \quad (k = 1, \dots, n)$$

con el orden lexicográfico entre los puntos, entonces p_i es un vértice convexo (¿puede el astuto lector determinar por qué?). Recordemos que el orden lexicográfico entre dos puntos del plano $p = (p_1, p_2)$ e $q = (q_1, q_2)$ se define como sigue: $p \leq q$ si $p_1 < q_1$ o si $p_1 = q_1$ y $p_2 \leq q_2$.

2.4. Orientación de un polígono

De manera análoga a los triángulos, decimos que el polígono simple

$$P = (p_1, p_2, \dots, p_n).$$

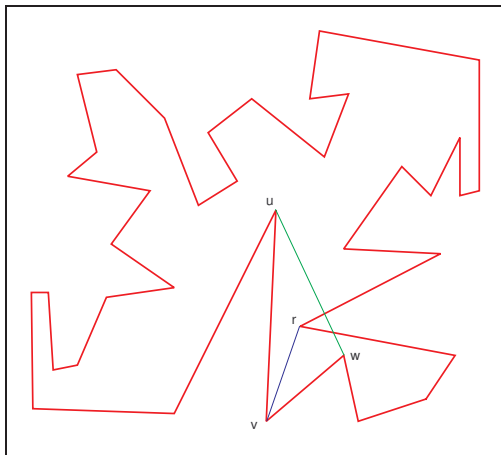
tiene orientación positiva si la lista de sus vértices están en el sentido de las agujas del reloj; en caso contrario, diremos que P tiene orientación negativa. Desde luego, con esta definición, la determinación algorítmica de la orientación de un polígono no parece ser una tarea sencilla. Bien, un poco de reflexión muestra que, en realidad, para determinar la orientación de un polígono, basta con hallar la orientación del triángulo que forman un vértice convexo y los dos vértices adyacentes al mismo; así pues, dado un polígono P , el siguiente algoritmo determina su orientación.

1. Determinése un vértice convexo, p_i , de P .
2. Sean

$$q = \begin{cases} p_{i-1} & \text{si } i > 1 \\ p_n & \text{si } i = 1 \end{cases} \quad \text{y} \quad r = \begin{cases} p_{i+1} & \text{si } i < n \\ p_1 & \text{si } i = n \end{cases}$$

3. Hállese la orientación del triángulo $T = (q, p_i, r)$.

2.5. Dado un polígono, hallar una diagonal interior



Que cualquier polígono simple, P , tiene, al menos, una diagonal interior no es una propiedad trivial. Informalmente, una prueba es: sea v un vértice convexo de P , llamemos u y w a los vértices anterior y posterior respectivamente y considérese el triángulo $T = (u, v, w)$. Si T no contiene ningún otro vértice del polígono, entonces el segmento (u, w) es una diagonal interior de P ; en caso contrario, elíjase de entre los vértices interiores a T aquel que está más lejano del segmento (u, w) y llamémosle r . Es relativamente fácil comprobar que el segmento (v, r) es una diagonal interior.

Algorítmicamente: dado un eneágono

$$P = (p_1, p_2, \dots, p_n)$$

orientado positivamente, el siguiente procedimiento obtiene

una diagonal interior.

1. Determinése un vértice convexo p_i de P .
2. Sean

$$u = \begin{cases} p_{i-1} & \text{si } i > 1 \\ p_n & \text{si } i = 1 \end{cases} \quad \text{y} \quad w = \begin{cases} p_{i+1} & \text{si } i < n \\ p_1 & \text{si } i = n \end{cases}$$

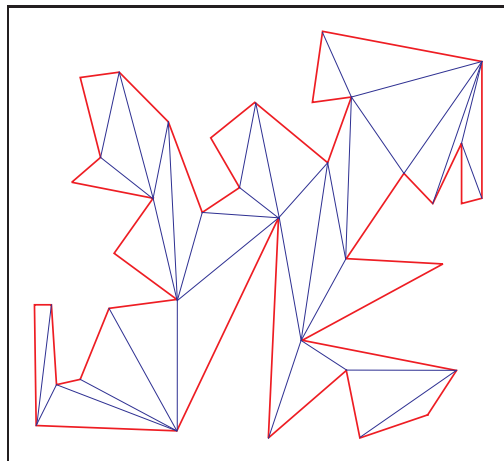
3. Hállese el subconjunto Q de los vértices de P que son interiores al triángulo $T = (u, p_i, w)$. Es decir,

$$Q = \{p_k \in P : p_k \text{ es interior a } T\}$$

4. Si $Q = \emptyset$, entonces (u, w) es la diagonal buscada.
5. En caso contrario, hállese el vértice $r \in Q$ cuya distancia al segmento (p_i, w) es mayor. El segmento (v, r) es la diagonal interior buscada.

3. COMIENZA EL JUEGO

3.1. Triangulación de un polígono



Con la ayuda de los algoritmos anteriores se puede abordar un famoso e importante problema en geometría computacional: la triangulación de un polígono. El término triangulación tiene aquí un sentido evidente. No obstante, conviene hacer notar que la triangulación se produce uniendo vértices del polígono.

Un resultado clásico establece que cualquier eneágono puede ser triangulado trazando $n - 2$ diagonales interiores.

La demostración habitual de este resultado –y el algoritmo que de ella se deduce– procede por recursión: el polígono original se escinde en dos piezas mediante una diagonal interior; a su vez cada una de las dos piezas menores se triangulan recursivamente.

Como el lector puede fácilmente imaginar, al final del proceso de recursión, lo que queda es triangular un trián-

gulo. Bueno, esta triangulación es evidente ¿verdad?.

Más precisamente, dado un eneágono

$$P = (p_1, p_2, \dots, p_n)$$

el siguiente algoritmo obtiene una triangulación del mismo.

1. Defínase

$$T(P) = P$$

si P es un triángulo.

2. En caso contrario, hállese $T(P)$ recursivamente mediante:

a) Sean p_i y p_k tal que (p_i, p_k) es una diagonal interior de P (suponemos, sin pérdida de generalidad, que $i < k$).

b) Sean

$$A = (p_i, p_{i+1}, \dots, p_k) \quad \text{y} \quad B = P \setminus A$$

c) Hállese

$$T(P) = T(A) \cup T(B)$$

3. $T(P)$ es la triangulación buscada.

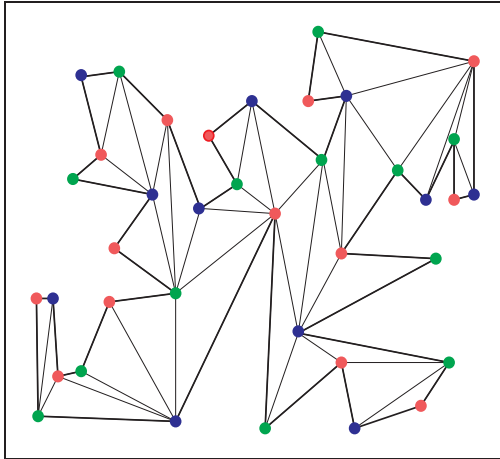
En la práctica computacional, la triangulación de un polígono se representa mediante índices. Esto es: si

$$(p_i, p_j, p_k)$$

es uno de los triángulos, la representación consiste en la lista de índices

$$(i, j, k).$$

3.2. Coloreado



que verifique que

$$|j - i| = 2$$

Una vez hallada tal oreja, llamémosle (u, v, w) , recursivamente se colorea el polígono triangulado que resulta tras el *desorejamiento*; esto es: el polígono que resulta de quitarle la oreja al original y al vértice v se le asigna el color que no aparece ni en u ni en w .

Algorítmicamente, dado un eneágono

$$P = (p_1, p_2, \dots, p_n)$$

el siguiente procedimiento colorea los vértices con tres colores (los designaremos como 1, 2, y 3). El resultado final es, entonces, una lista de colores, uno para cada vértice, en el mismo orden en que aparecen en P .

1. Sea Q el polígono P triangulado.
2. Defínase, si Q es un triángulo,

$$C(Q) = (1, 2, 3).$$

3. En caso contrario, procédase recursivamente en la forma siguiente:

- a) Selecciónese una oreja (u, v, w) de Q y defínase

$$Q' = Q \setminus \{v\}.$$

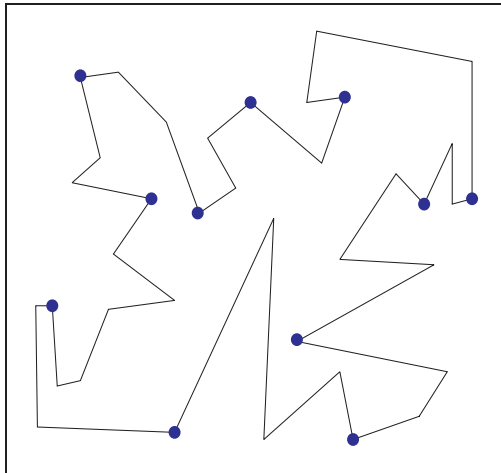
- b) Coloréese Q' ; es decir, hállese

$$C(Q').$$

- c) Insértese en $C(Q')$, en el lugar correspondiente al vértice v , el color complementario al que tienen u y w .

Como en los casos anteriores, el lector debe tener en cuenta que, por claridad, sólo se ha pretendido dar una descripción informal de los algoritmos. Desde luego, los detalles de la construcción práctica son algo más complejos, pero no mucho más como puede comprobar en los archivos que se adjuntan.

4. LA RESOLUCIÓN FINAL



En este punto, tras lo que el autor espera haya sido para el lector un divertido recorrido por la geometría computacional, la solución al problema inicial de cómo vigilar un museo aparece como trivial.

Para situar las cámaras de vigilancia del museo se triangula la planta y se colorea.

Con los vértices coloreados, se eligen aquellos que tengan el color con menor frecuencia. Puesto que todos los triángulos tienen los tres colores en sus vértices, un sólo color basta para vigilar el triángulo entero, de manera que todos los triángulos estarán vigilados. Conclusión: el museo queda vigilado en su totalidad.

Algorítmicamente, para un museo

$$P = (p_1, p_2, \dots, p_n)$$

el siguiente algoritmo sitúa los vigilantes de forma adecuada:

1. Triángulense el polígono.
2. Coloréese la triangulación.
3. Determinéense los vértices con color cuya frecuencia es menor.
4. Sitúese una cámara en cada vértice hallado.

REFERENCIAS

- [1] Eppstein, D.
Geometry in Action.
<http://www.ics.uci.edu/~eppstein/geom.html>
- [2] Fiume, E.
An Introduction to Scientific Symbolic and Graphical Computation.
A K Peters, 1995.
- [3] Maeder, R.
Programming in Mathematica.
Addison-Wesley, 1991.
- [4] Skiena, S.
Implementing Discrete Mathematics.
Addison-Wesley, 1990
- [5] Vardi, I.
Computational Recreations in Mathematics.
Addison-Wesley, 1991.
- [6] Wagon, S.
Mathematica in Action.
Springer-Verlag, 1999.
- [7] Wolfram, S.
Mathematica. A System for Doing Mathematics by Computer.
Addison-Wesley, 1991.