

# Cálculo Aproximado de Diagramas de Voronoi utilizando Autómatas Celulares

Christian Trefftz

Departamento de Informática y Sistemas

Universidad EAFIT

Medellín Colombia

ctrefftz@sigma.eafit.edu.co

## Abstract

Se presenta un modelo que utiliza autómatas celulares para resolver en forma aproximada el problema de los diagramas de Voronoi.

## 1 Introducción

El problema de construir diagramas de Voronoi [1, 9] se puede describir así: Dado  $E^d$  (un espacio euclideo de  $d$  dimensiones) y un conjunto de puntos de entrada (sitios o generadores), encuentre el conjunto de puntos en el espacio que están mas cerca de cada generador.

Si  $d = 2$ , entonces el problema se convierte en la “baldozinación” (tiling) de un plano. Los diagramas de Voronoi en el plano tiene aplicaciones en muchos campos diferentes: Gráficas computarizadas [2], segmentación por texturas [6] y navegación de robots [8].

Las primeras soluciones [4, 5] a este problema se basaron en encontrar las ecuaciones de las líneas que están a distancias iguales entre cada par de puntos,

calcular los interceptos entre estas líneas y recortar (clip) apropiadamente. Una nueva alternativa se basa en la técnica de barrido del plano [3]. Una línea barre el plano y las líneas que forman el diagrama se van calculando a medida que se va barriendo el plano.

Para ciertas aplicaciones, una aproximación gráfica es suficiente para propósitos prácticos. Al usuario le interesa una imagen que le suministre información acerca de la estructura de la "baldozización". Una aproximación gráfica se puede generar en la pantalla de un computador.

## 2 Una primera aproximación

Si una aproximación es suficiente, se puede usar la definición de Diagrama de Voronoi para colorear los pixels de la imagen de acuerdo con el generador que esté mas cerca, es decir, se asigna un color diferente a cada generador, y, para cada pixel, se calcula distancia desde ese pixel a todos los generadores. Al pixel se le asigna el color del generador mas cercano. Si el pixel queda a la misma distancia de dos generadores diferentes, este pixel se encuentra en la frontera entre dos regiones del plano y se colorea negro. La imagen resultante es una aproximación del diagrama de Voronoi en el plano.

En el siguiente algoritmo se asume que cada pixel y los generadores tienen coordenadas  $x$  y  $y$  que han sido escaladas apropiadamente del dominio del problema al sistema gráfico del computador que se está usando. En este proceso se introducen errores de redondeo, pero como se dijo anteriormente, solo se está calculando una aproximación.

La forma general del algoritmo es la siguiente:

```
begin  
  Lea las coordenadas de los generadores  
  Para cada pixel en la pantalla  
    Calcule la distancia entre el pixel y cada uno de los generadores  
    Asigne al pixel el color del generador mas cercano  
  FinPara  
  Muestre la imagen  
end
```

Este algoritmo se presta para ser implantado en máquinas paralelas en las que cada área de la pantalla se pueden asignar a un procesador diferente. Treftz, Szakas y McKinley realizaron una implantación en un MasPar MP-1. Los detalles se pueden encontrar en [7].

Si uno considera cada pixel de la pantalla como una celda de un autómata celular, este algoritmo se puede considerar, en un sentido amplio, como una aplicación de autómatas celulares. El lector argumentará que este algoritmo no corresponde realmente a un autómata celular porque el estado de las celdas del autómata no depende exclusivamente del estado anterior de cada celda y del estado de las celdas vecinas sino de unos datos que hay que difundir a todas las celdas participantes.

### 3 Una segunda versión

Si se quiere ser mas fiel al “espíritu” de los autómatas celulares, los nuevos estados de los autómatas se deben basar únicamente en el estado previo anterior y en los estados de los vecinos.

Si se sacrifica la inmediatez de la respuesta, aún es posible obtener una solución. En el momento inicial, únicamente los puntos generadores tienen color y todos los demás pixels estarán “en blanco”. En cada uno de los pasos siguientes, cada pixel realizará la siguiente operación: Si el pixel ya tiene asig-

nado un color, conserva su color. Si el pixel no tiene asignado un color, examine los colores de sus vecinos: Si ningún vecino tiene color, permanezca sin color; Si hay uno o mas vecinos con el mismo color, adopta ese color; si hay vecinos con colores diferentes, este pixel se encuentra en una frontera entre regiones y se colorea negro.

La forma general de esta nueva versión del algoritmo es la siguiente:

```
begin
  Cada generador tiene un color diferente
  Repetir hasta que no haya cambios
    Para cada pixel
      Si no tiene color asignado
        Examine los vecinos
        Si no hay vecinos con color,
          permanezca sin color
        FinSi
      Si uno o mas vecinos tienen el mismo color,
        adopte ese color
      FinSi
      Si hay dos o mas vecinos con diferentes colores,
        adopte el color negro
      FinSi
    FinSi
  FinRepetir
  Muestre la imagen
end
```

## 4 Conclusiones y Trabajo Futuro

Se ha descrito una aplicación de los autómatas celulares para obtener una solución aproximada al problema de los diagramas de Voronoi.

La experiencia con este algoritmo sugiere que los autómatas celulares son un modelo conceptual muy apropiado para ciertas aplicaciones en computadores paralelos. Planeamos explorar la factibilidad de utilizar modelos de autómatas celulares en redes de estaciones de trabajo.

## Agradecimientos

El autor desea expresar sus agradecimientos a B. Ostrowsky por una muy útil conversación acerca del tema.

## References

- [1] F. Aurenhammer, "Voronoi Diagrams - A survey of fundamental geometric data structures," *ACM Computing Surveys*, vol. 23, pp. 345-405, 1991.
- [2] David P Dobkin, "Computational Geometry and Computer Graphics", *Proceedings of the IEEE*, Vol. 80, No. 9, pp.1400-1411, 1992.
- [3] F. P. Preparata y M. Shamos, "Computational Geometry: An Introduction", Springer, New York, 1985.
- [4] Green, P.J y R. Sibson, "Computing Dirichlet tessellations in the plane," *The Computer Journal*, 21(2), 168-173, 1978.
- [5] Kokichi Sugihara y Masao Iri, "Construction of the Voronoi diagram for "One Million" Generators in Single-Precision Arithmetic", *Proceedings of the IEEE*, Vol. 80, No. 9, pp.1471-1484, 1992.

- [6] Mihran Tuceryan y Anil K. Jain, "Texture Segmentation Using Voronoi Polygons," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 2, pp.211-216, 1992.
- [7] Christian Treftz, Joseph Szakas y Philip McKinley, "A Parallel Algorithm for Graphically Approximating Voronoi Diagrams", Technical Report ACS-81, Department of Computer Science, Michigan State University, 1993.
- [8] Ana María Trejos y Juan David Velásquez, "Diseño de una Arquitectura para un Robot Autónomo Navegante", Proyecto de Grado, Departamento de Informática y Sistemas, Universidad EAFIT, 1993.
- [9] G. Voronoi, "Nouvelles applications des parametres continus a la theorie des formes quadratiques," *Journal fur die reine und angewandte Mathematik*, vol. 133, pp.97-178, 1907