2023

Difusión anisotrópica

EN IMAGEN Y NUBES DE PUNTOS 2D Y 3D

ALBERTO MORCILLO SANZ

Difusión anisotrópica

En computer vision, la difusión anisotrópica, es una técnica cuyo objetivo es reducir el ruido de la imagen sin eliminar partes significativas del contenido de la imagen, normalmente bordes, líneas u otros detalles que son importantes para la interpretación de la imagen.

Definición

Sea omega un subconjunto del espacio R^2

$$\Omega \subset \mathbb{R}^2$$

Tenemos una función escalar Phi definida en omega y en función del tiempo

$$\Phi(\cdot, t) : \Omega \to \mathbb{R}$$

En el caso de trabajar con imagen, omega serían el conjunto de puntos (x, y) que se encuentran entre [0, width] x [0, height] de la imagen.

La función escalar Phi cuando t = 0 sería la propia imagen, ya que para cada (x, y) le corresponde un escalar, que sería el color del pixel (el color está en escala de grises)

Con la ecuación de difusión podremos encontrar la imagen cuando t = 1, y después cuando t = 2, y así hasta t = n, siendo n el número de iteraciones totales. A medida que aumentan las iteraciones el blur (excepto en los bordes) va aumentando.

La ecuación de la difusión anisotrópica viene dada por:

$$\frac{\partial \Phi}{\partial t} = \nabla \cdot [c(\|\nabla \Phi\|)\nabla \Phi] = \nabla c \cdot \nabla \Phi + c(\|\nabla \Phi\|)\Delta \Phi$$

Siendo Δ el Laplaciano, ∇ el gradiente y ∇ • la divergencia

Si c fuera constante, por linealidad podríamos sacarlo fuera de la divergencia llegando a la ecuación de difusión isotrópica o más conocida como ecuación de calor (La derivada respecto del tiempo es igual al laplaciano). Siendo c constante, la difusión se aplicaría de forma uniforme y el resultado sería equivalente a un Gaussian Blur. Esto sería un problema ya que, si bien aplica un suavizado, “destruiría” los bordes. Siendo estos bordes, la separación entre distintas regiones.

Pietro Perona y Jitendra Malik en 1990 propusieron el coeficiente de difusión:

$$c(\|\nabla \Phi\|) = e^{-(\frac{\|\nabla \Phi\|}{k})^2}$$

Esta función sirve para detectar los bordes (es o se aproxima a 0 en los bordes) y depende de la norma del gradiente y del parámetro k.

El parámetro k sirve para detectar más o menos bordes.

Mientras que la función c no sea constante, la ecuación seguirá siendo anisotrópica. Por lo que podríamos utilizar otra función diferente a c. Pero para este propósito la función c es una buena opción.

En el caso de una imagen, el gradiente es un vector que nos indica la variación de color. Podemos interpretar la norma del gradiente como la cantidad de variación de color.

Si nos fijamos, como los bordes se encuentran entre regiones diferentes (colores diferentes) la variación de color es mayor que dentro de las regiones, donde los colores de los pixeles vecinos son iguales o muy parecidos.

Por tanto, en los bordes, la norma del gradiente es mayor, y la función c tiende a 0. Siendo la segunda parte de la igualdad de la ecuación 0, quedaría que la derivada respecto del tiempo de Phi en el borde es 0, lo que nos indica que la imagen no varía en esos puntos (borde en este caso) y es por esa razón por la que los bordes siempre se quedan como están y no se "destruyen".

En el caso de no estar en un borde, la norma del gradiente en estos puntos es menor y por tanto la función c no tiende a 0. De forma que la derivada respecto del tiempo ya no es cero. Ahora la imagen sí varía. Esta variación viene dada por la segunda igualdad de la ecuación de difusión anisotrópica, aplicando un Blur en estos puntos.

Suavizado de nubes de puntos 2D con difusión anisotrópica

A diferencia de los píxeles, los puntos de una nube de puntos están dispersos por el espacio y no son continuos. Para solucionar esto, podemos emplear el siguiente algoritmo:

- Para poder aplicar la difusión anisotrópica se tratará de crear un Grid donde cada casilla contendrá los puntos correspondientes de la nube de puntos y así podremos aplicar la ecuación de forma similar a una imagen. El tamaño de cada casilla es un parámetro que hay que tener en cuenta ya que afecta directamente a los resultados del suavizado.

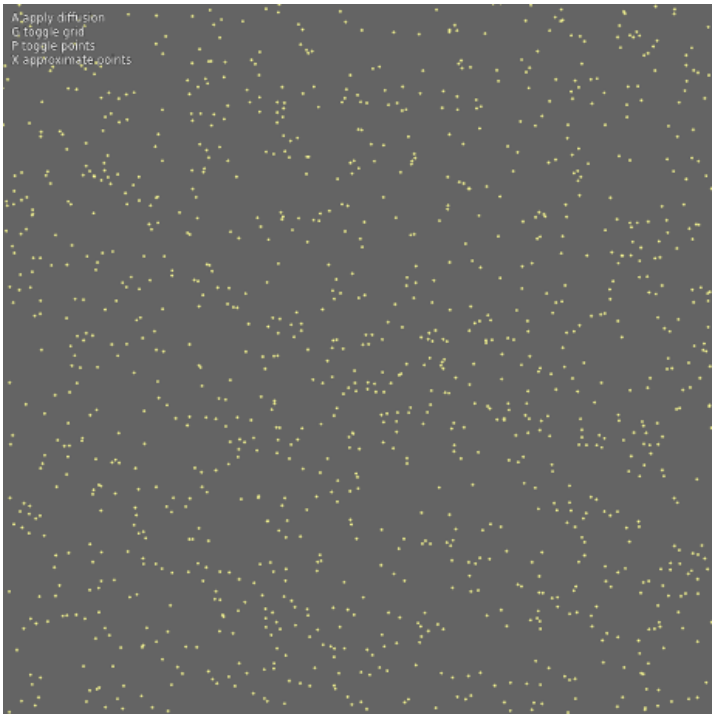

Figura 1. Nube de puntos 2D generada de forma aleatoria

- Cada punto irradia una temperatura. Si asumimos que todos los puntos irradian la misma temperatura (1, adimensional; para simplificar cálculos), la temperatura de cada casilla del Grid será la suma de temperaturas de los puntos que contiene; siendo

igual al número de puntos que contiene. De esta forma ya tenemos la función Phi, donde cada casilla (que podemos considerarla como un pixel en imagen), tiene asociado un escalar (en el caso de imagen, un color; en este caso, una temperatura):

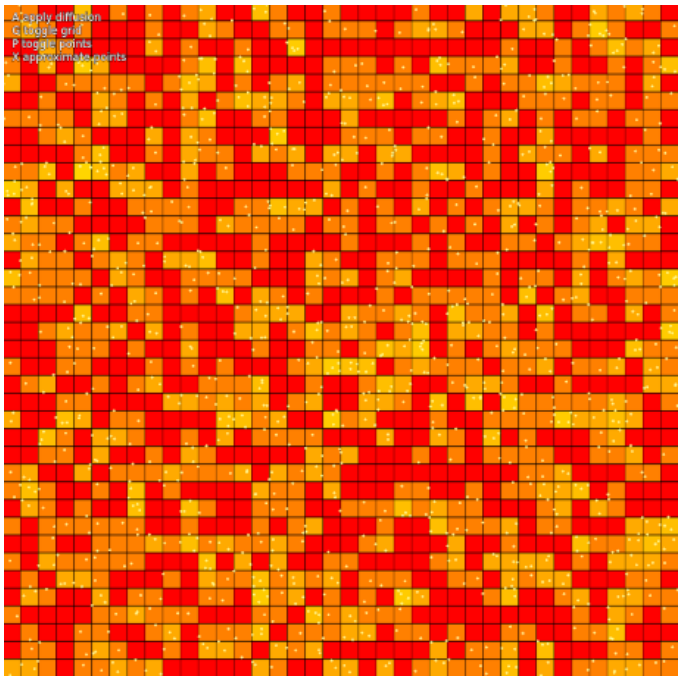

Figura 2. Grid y visualización de temperaturas

- Una vez teniendo las temperaturas, se aplica la ecuación para suavizar las temperaturas:

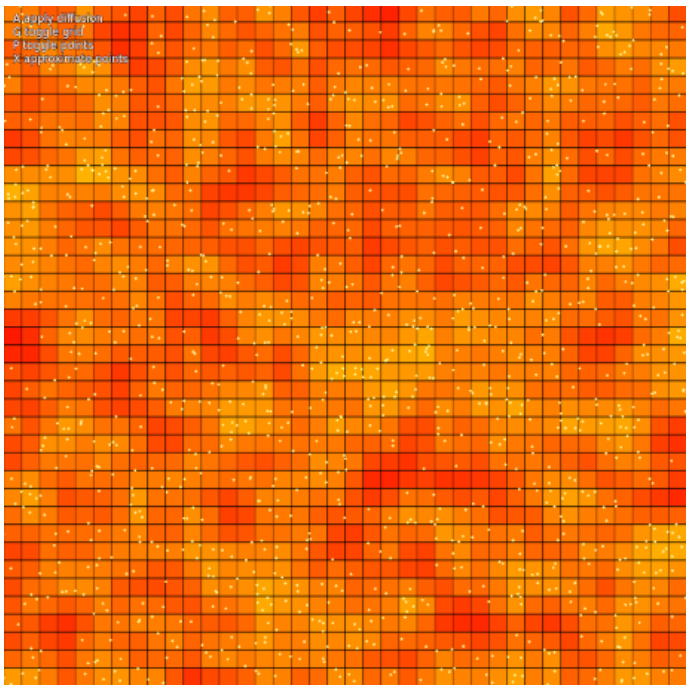

Figura 3. Grid después de la difusión anisotrópica

Como podemos observar, las zonas más cálidas (colores claros) tienen una mayor densidad de puntos que las zonas más frías. De esta forma tenemos un mapa con las regiones más pobladas.

- Como el ruido está principalmente en las zonas frías, queremos que los puntos de estas zonas se aproximen a las zonas más cálidas. Para ello, para cada casilla del Grid, se calculará el gradiente, y moveremos los puntos contenidos en la casilla en la dirección positiva del gradiente (la dirección positiva del gradiente apunta hacia un máximo, por tanto, apunta hacia una región más cálida). Este proceso es conocido como Gradient Ascent:

$$v_{n+1} = v_n + \gamma \nabla \Phi(v_n)$$

Y el gradiente viene dado por:

$$\nabla \Phi = (\frac{\partial \Phi}{\partial x}, \frac{\partial \Phi}{\partial y})$$

Siendo γ el "step-size", es decir, la cantidad que tiene que moverse cada punto en la dirección del gradiente. Es importante considerar valores no muy grandes para este parámetro para aumentar la precisión.

Cuantas más veces se repita este proceso, más próximos quedarán los puntos de la nube de puntos:

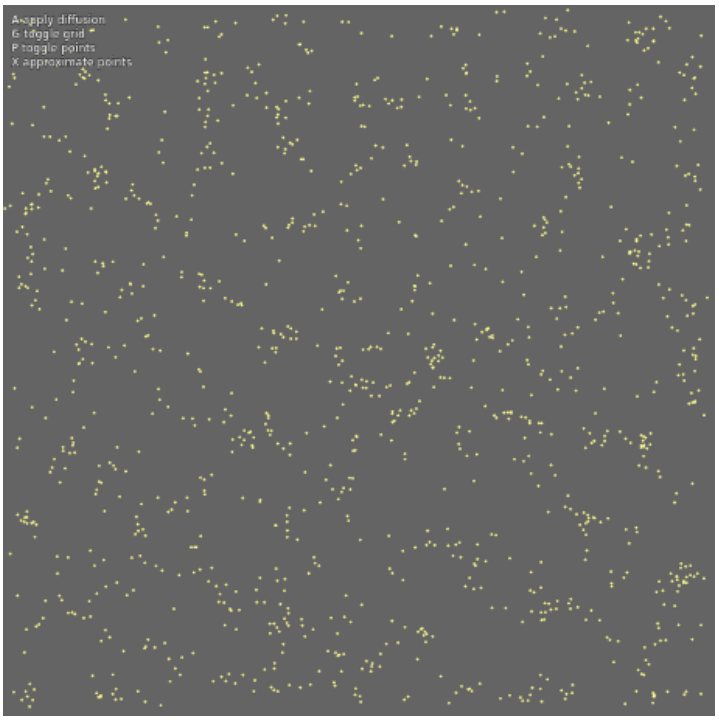

Figura 4. Nube de puntos 2D suavizada

Discretización de la ecuación de difusión anisotrópica

Para este propósito no tiene sentido encontrar soluciones de Phi de forma analítica, ya que lo que se busca es encontrar aproximaciones con un error mínimo con un ordenador. Los ordenadores no trabajan de forma continua tampoco entienden lo que es el infinito o lo que puede ser un infinitesimal. Los ordenadores trabajan de forma discreta.

Por tanto, se mostrará el proceso con el que convertimos la ecuación principal de continua a discreta.

Partimos de que la difusión anisotrópica está definida en R^2 y Phi una función continua.

$$\frac{\partial \Phi}{\partial t} = \nabla \cdot [c(\|\nabla \Phi\|)\nabla \Phi] = \nabla c \cdot \nabla \Phi + c(\|\nabla \Phi\|)\Delta \Phi$$

Y el coeficiente de difusión:

$$c(\|\nabla \Phi\|) = e^{-(\frac{\|\nabla \Phi\|}{k})^2}$$

Asumiendo que:

$$\nabla c \cdot \nabla \Phi + c(\|\nabla \Phi\|)\Delta \Phi \approx c(\|\nabla \Phi\|)\Delta \Phi$$

Tendríamos:

$$\frac{\partial \Phi}{\partial t} \approx c(\|\nabla \Phi\|)\left(\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2}\right)$$

Para discretizarlo utilizaremos diferencias finitas, que es un método comúnmente utilizado para calcular de manera numérica soluciones de ecuaciones diferenciales usando ecuaciones diferenciales finitas para aproximar las derivadas:

$$\frac{\partial \Phi}{\partial t} \approx \frac{\Phi_{ij}^{t+1} - \Phi_{ij}^{t}}{\Delta t}, \quad \frac{\partial^2 \Phi}{\partial x^2} \approx \frac{\Phi_{i+\Delta x,j}^{t} - 2\Phi_{i,j}^{t} + \Phi_{i-\Delta x,j}^{t}}{\Delta x^2},$$

$$\frac{\partial^2 \Phi}{\partial y^2} \approx \frac{\Phi_{i,j+\Delta y}^{t} - 2\Phi_{i,j}^{t} + \Phi_{i,j-\Delta y}^{t}}{\Delta y^2}$$

Como Δx = Δy = 1 ya que los píxeles en una imagen o las casillas de un Grid distan 1:

$$\frac{\partial^2 \Phi}{\partial x^2} \approx \Phi_{i+1,j}^{t} - 2\Phi_{i,j}^{t} + \Phi_{i-1,j}^{t}, \quad \frac{\partial^2 \Phi}{\partial y^2} \approx \Phi_{i,j+1}^{t} - 2\Phi_{i,j}^{t} + \Phi_{i,j-1}^{t}$$

$$\nabla_e \Phi_{i,j}^{t} = \Phi_{i+1,j}^{t} - \Phi_{i,j}^{t}, \quad \nabla_w \Phi_{i,j}^{t} = \Phi_{i-1,j}^{t} - \Phi_{i,j}^{t}$$

$$\nabla_n \Phi_{i,j}^{t} = \Phi_{i,j-1}^{t} - \Phi_{i,j}^{t}, \quad \nabla_s \Phi_{i,j}^{t} = \Phi_{i,j+1}^{t} - \Phi_{i,j}^{t}$$

Por tanto:

$$\Delta \Phi = \left(\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2}\right) = \left[\left(\nabla_e \Phi_{i,j}^{t} + \nabla_w \Phi_{i,j}^{t}\right) + \left(\nabla_n \Phi_{i,j}^{t} + \nabla_s \Phi_{i,j}^{t}\right)\right]$$

Sustituyendo la aproximación de la derivada respecto del tiempo y la aproximación del laplaciano en la ecuación obtenemos:

$$\frac{\Phi_{i,j}^{t+1} - \Phi_{i,j}^{t}}{\Delta t} = c(\|\nabla\Phi\|)\left(\nabla_e\Phi_{i,j}^{t} + \nabla_w\Phi_{i,j}^{t} + \nabla_n\Phi_{i,j}^{t} + \nabla_s\Phi_{i,j}^{t}\right) =$$

$$= c_n\nabla_n\Phi_{i,j}^{t} + c_s\nabla_s\Phi_{i,j}^{t} + c_e\nabla_e\Phi_{i,j}^{t} + c_w\nabla_w\Phi_{i,j}^{t}$$

El coeficiente c en m, siendo m = n, s, e y w:

$$c_m = c(|\nabla_m\Phi_{i,j}^{t}|) = e^{\left(\frac{|\nabla_m\Phi_{i,j}^{t}|}{k}\right)^2}$$

Como el paso del tiempo viene dado por las iteraciones, asumimos que cada iteración es un momento nuevo por tanto asumimos que Δt = 1 y finalmente llegamos a la expresión:

$$\Phi_{i,j}^{t+1} = \Phi_{i,j}^{t} + \lambda\left(c_n\nabla_n\Phi_{i,j}^{t} + c_s\nabla_s\Phi_{i,j}^{t} + c_e\nabla_e\Phi_{i,j}^{t} + c_w\nabla_w\Phi_{i,j}^{t}\right)$$

Donde se ha añadido el parámetro λ (entre 0 y 1), para regular la cantidad de difusión.

Suavizado de nubes de puntos 3D con difusión anisotrópica

Se podría aplicar el mismo proceso que en 2D, pero voxelizando la nube de puntos y aplicando la ecuación en R^3. Donde antes utilizábamos las casillas del Grid, ahora utilizaríamos voxels.

Hay que tener en cuenta que el trabajo computacional en este caso es mayor y quizás sea necesario utilizar varios hilos de ejecución o incluso la GPU para poder suavizar nubes de puntos, ya que, si no, podría llevar demasiado tiempo.

Si hacemos el proceso de discretización en R^3, obtendríamos:

$$\Phi_{i,j,k}^{t+1} = \Phi_{i,j,k}^{t} + \lambda\left(c_n\nabla_n\Phi_{i,j,k}^{t} + c_s\nabla_s\Phi_{i,j,k}^{t} + c_e\nabla_e\Phi_{i,j,k}^{t} + c_w\nabla_w\Phi_{i,j,k}^{t} + c_f\nabla_f\Phi_{i,j,k}^{t} + c_b\nabla_b\Phi_{i,j,k}^{t}\right)$$

Donde:

$$\nabla_f\Phi_{i,j,k}^{t} = \Phi_{i,j,k-1}^{t} - \Phi_{i,j,k}^{t}, \quad \nabla_b\Phi_{i,j,k}^{t} = \Phi_{i,j,k+1}^{t} - \Phi_{i,j,k}^{t}$$

Referencias

Paper: https://ieeexplore.ieee.org/document/56205

Wikipedia: https://en.wikipedia.org/wiki/Anisotropic_diffusion