

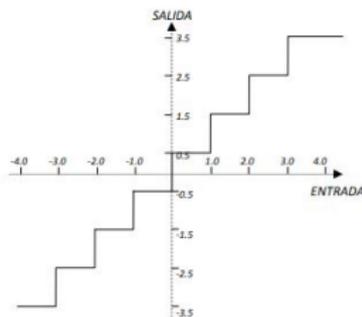
Lección 5: Cuantización Vectorial

Jesús Savage
Carlos Rivera

28 de marzo de 2022

Cuantización Vectorial

La cuantización vectorial es una generalización de la cuantización escalar.



En lugar de usar un solo escalar para cuantizarlo se utiliza todo un vector para ser cuantizado.

Cuantización Vectorial

Un cuantizador vectorial Q , de dimensión k y tamaño N , es una transformación de un vector del espacio euclidiano de dimensión R^k en un conjunto finito C que contiene N salidas o puntos de reproducción llamados vectores de código (codevectors), donde:

$$Q : R^k \rightarrow C$$

$$C = \{\underline{y}_1, \underline{y}_2, \dots, \underline{y}_n\}$$

$$\underline{y}_i \in R^k \forall i \in J = \{1, 2, \dots, N\}$$

Cuantización Vectorial

El conjunto C es llamado el alfabeto (CodeBook) y tiene un tamaño N , lo que significa que tiene N elementos distintos, donde cada uno de ellos es un vector en R^k

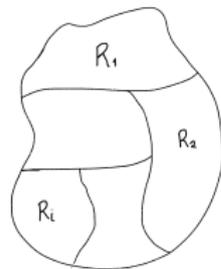
Asociado a cada punto del cuantizador vectorial hay una partición de R^k en N regiones.

La i -ésima región es definida por:

$$R_i = \{\underline{x} \in R^k; Q(\underline{x}) = \underline{y}_i\}$$

$$\cup R_i = R^k$$

$$R_i \cap R_j = \emptyset \text{ para } i \neq j$$



Cuantización Vectorial

Un cuantizador vectorial puede ser descompuesto en dos operaciones, la codificación vectorial y la decodificación vectorial.

$$E : R^k \rightarrow J$$

y

$$D : J \rightarrow R^k$$

La operación total de un cuantizador vectorial puede ser interpretado como:

$$Q(\underline{x}) = D(E(\underline{x}))$$

Cuantización Vectorial

Para un alfabeto dado, una partición óptima es la que satisface la condición del vecino más cercano, para cada región R_i , todos los puntos más cercanos al vector de código \underline{y}_i que a cualquier otro código de vector deberá ser asignado a la región R_i .

$$R_i \subset \{\underline{x} : d(\underline{x}, \underline{y}_i) \leq d(\underline{x}, \underline{y}_j)\} \quad \forall j \neq i$$

esto en

$$Q(\underline{x}) = \underline{y}_i$$

solo si

$$d(\underline{x}, \underline{y}_i) \leq d(\underline{x}, \underline{y}_j)$$

$$d(\underline{x}, Q(\underline{x})) = \min d(\underline{x}, \underline{y}_i)$$

Donde $d(\underline{x}, \underline{y}_i)$ es una función que mide que tan parecidos son los vectores $\underline{x}, \underline{y}_i$

Medidas de Distorsión

Para una medida de distorsión de error cuadrático medio, $E(|\underline{x} - \underline{y}|^2)$ es minimizado cuando \underline{y} es la media de \underline{x}

$$f(\underline{y}) = E\{|\underline{x} - \underline{y}|^2\} = E\{(\underline{x}^2 - 2\underline{x}\underline{y} + \underline{y}^2)\} = E\{\underline{x}^2\} - 2\underline{y}E\{\underline{x}\} + \underline{y}^2$$

$$\frac{df(\underline{y})}{d\underline{y}} = -2E\{\underline{x}\} + 2\underline{y} = 0$$

$$\Rightarrow \underline{y} = E\{\underline{x}\}$$

Entonces:

$$Cent(R_i) = E(\underline{x} | \underline{x} \in R_i)$$

Cuantización Vectorial

Si cada una de las \underline{x} en R_j tiene una función de densidad de probabilidad uniforme:

$$Cent(R) = \frac{1}{L} \sum_{i=1}^L \underline{x}_i$$

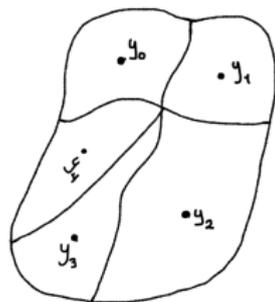
Para: $\underline{x}_i \in R_j, i = 1, 2, 3, \dots, L$

Cuantización Vectorial

Una de las condiciones de optimilidad es que para una partición dada $\{R_i : i = 1, \dots, N\}$ los vectores de código satisfacen

$$y_i = \text{cent}(R_i)$$

$$\text{cent}(R_i) = \min E\{d(\underline{x}, \underline{y}) \mid \underline{x} \in R_i\}$$



Cuantización Vectorial

En las comunicaciones digitales el codificador se encarga de codificar un vector \underline{x} a un vector \underline{y}_i el cual es parecido a \underline{x} . El índice i del vector seleccionado es transmitido (binario) al receptor, donde este realiza un procedimiento de búsqueda en la tabla y genera la reproducción \underline{y}_i . El objetivo principal en el diseño de un cuantizador vectorial es el de encontrar el alfabeto, una partición y una regla de decisión que maximiza la medida del desempeño total considerando la secuencia entera de vectores a ser codificados por el cuantizador.

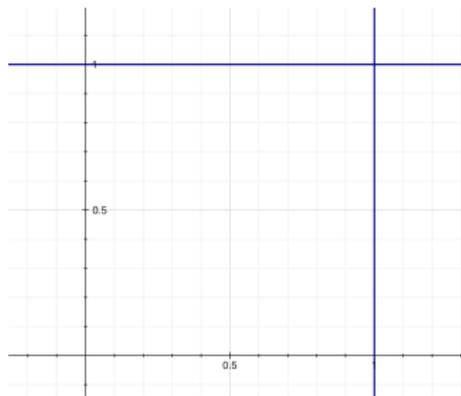
$$D = E \{d(\underline{x}, Q(\underline{x}))\} = \int d(\underline{x}, Q(\underline{x})) f_{\underline{x}}(\underline{x}) d\underline{x}$$

$f_{\underline{x}}(\underline{x})$ es la función de probabilidad conjunta del vector \underline{x} y la integral es una integral múltiple sobre un espacio de k dimensiones.

$$D = E\{d(\underline{x}, Q(\underline{x}))\} = \sum d(\underline{x}_i, Q(\underline{x}_i)) q_x(\underline{x}_i)$$

Ejemplo

Objetivo: Encontrar un centroide $\underline{x}_c = (x_c, y_c)$ que disminuya el error cuadrático para vectores de tamaño 2 con una función uniforme conjunta entre 0 y 1:



$$d = \iint |\underline{x} - \underline{x}_c|^2 f(\underline{x}) d\underline{x}$$

Ejemplo

Para una función de densidad conjunta $f(\underline{x})$ uniforme $\underline{x} = (x, y)$

$$\begin{aligned}d &= \int_0^1 \int_0^1 [(x - x_c)^2 + (y - y_c)^2] f(x, y) dy dx \\&= \frac{1}{1} \int_0^1 \int_0^1 [(x - x_c)^2 + (y - y_c)^2] dy dx \\&= \int_0^1 [(x - x_c)^2 y + \frac{1}{3}(y - y_c)^3] \Big|_0^1 dx \\&= \int_0^1 [(x - x_c)^2 + \frac{1}{3}(1 - y_c)^3 + \frac{1}{3}y_c^3] dx \\&= \frac{1}{3}(x - x_c)^3 + \frac{1}{3}(1 - y_c)^3 x + \frac{1}{3}y_c^3 x \Big|_0^1 \\&= \frac{1}{3}(1 - x_c)^3 + \frac{1}{3}(1 - y_c)^3 + \frac{1}{3}y_c^3 + \frac{1}{3}x_c^3 = g(x_c, y_c)\end{aligned}$$

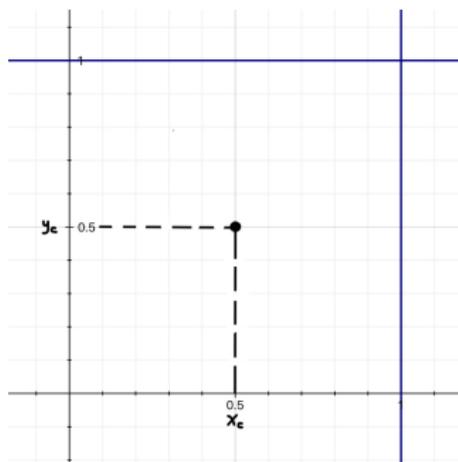
Ejemplo

Para encontrar los puntos x_c y y_c óptimos minimizando $g(x_c, y_c)$

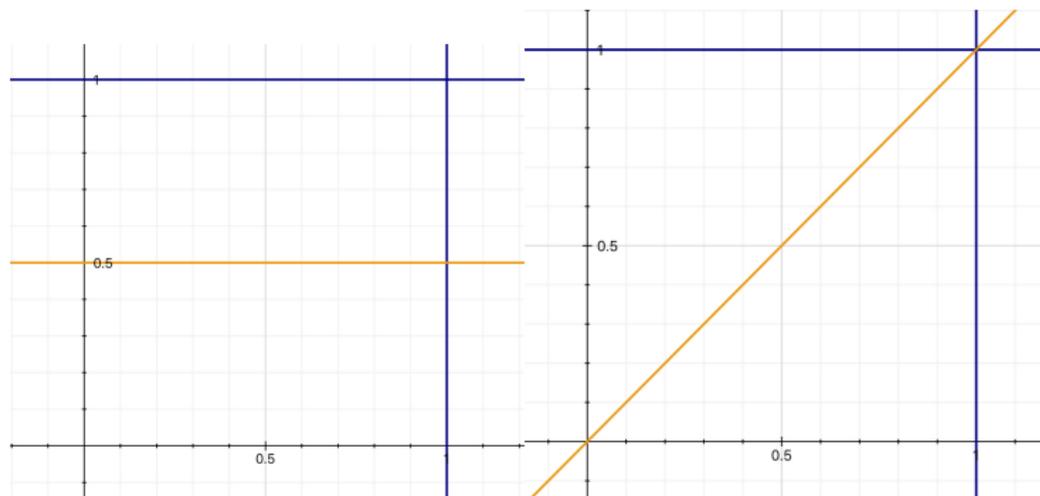
$$\begin{aligned}\frac{\delta g(x_c, y_c)}{\delta x_c} &= -(1 - x_c)^2 + x_c^2 \\ &= -1 + 2x_c - x_c^2 + x_c^2 = 0 \\ x_c &= \frac{1}{2}\end{aligned}$$

$$\begin{aligned}\frac{\delta g(x_c, y_c)}{\delta y_c} &= -(1 - y_c)^2 + y_c^2 \\ &= -1 + 2y_c - y_c^2 + y_c^2 = 0 \\ y_c &= \frac{1}{2}\end{aligned}$$

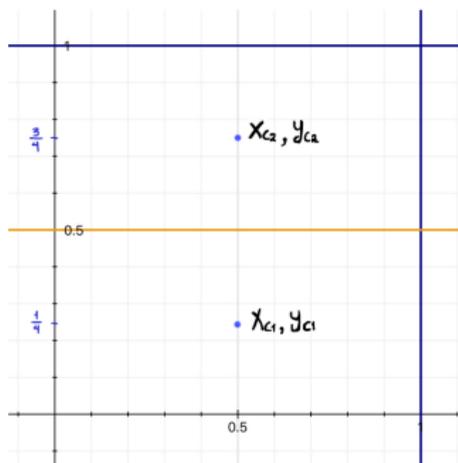
Ejemplo



¿Cómo seguir haciendo particiones?



Caso 1



$$d = \iint |\underline{x} - \underline{x}_c|^2 f(\underline{x}) d\underline{x}$$

Caso 1

$$\begin{aligned}d_1 &= \int_0^1 \int_0^{\frac{1}{2}} ((x - x_c)^2 + (y - y_c)^2) f_1(x, y) dy dx \\&= \frac{1}{2} \int_0^1 \left[(x - x_c)^2 y + \frac{1}{3} (y - y_c)^3 \right] \Big|_0^{\frac{1}{2}} dx \\&= \frac{1}{2} \int_0^1 \left[(x - x_c)^2 \frac{1}{2} + \frac{1}{3} \left[\frac{1}{2} - y_c \right]^3 + \frac{1}{3} y_c^3 \right] dx \\&= \frac{1}{2} \left[\frac{1}{6} (x - x_c)^3 + \left[\frac{1}{3} \left[\frac{1}{2} - y_c \right]^3 + \frac{1}{3} y_c^3 \right] x \right] \Big|_0^1 \\&= \frac{1}{2} \left[\frac{1}{6} (1 - x_c)^3 + \left[\frac{1}{3} \left[\frac{1}{2} - y_c \right]^3 + \frac{1}{3} y_c^3 \right] + \frac{1}{6} x_c^3 \right] = g_1(x_c, y_c)\end{aligned}$$

Caso 1

$$\frac{\delta g_1(x_c, y_c)}{\delta x_c} = \frac{1}{2} \left(-\frac{1}{2}(1 - x_c)^2 + \frac{1}{2}x_c^2 \right) = 0$$

$$= (1 - x_c)^2 - x_c^2 = 1 - 2x_c + x_c^2 - x_c^2$$

$$= 1 - 2x_c = 0$$

$$x_c = \frac{1}{2}$$

$$\frac{\delta g_1(x_c, y_c)}{\delta y_c} = -\left[\frac{1}{2} - y_c\right]^2 + y_c^2 = 0$$

$$-\frac{1}{4} + y_c - y_c^2 + y_c^2 = -\frac{1}{4} + y_c = 0$$

$$y_c = \frac{1}{4}$$

Mínimos

$$g_1(x_c, y_c) = \left(\frac{1}{2}, \frac{1}{4}\right)$$

$$\begin{aligned}g_1\left(\frac{1}{2}, \frac{1}{4}\right) &= \frac{1}{2} \left[\frac{1}{6} \left(1 - \frac{1}{2}\right)^3 + \frac{1}{3} \left[\frac{1}{2} - \frac{1}{4}\right]^3 + \frac{1}{3} \left(\frac{1}{4}\right)^3 + \frac{1}{6} \frac{1}{8} \right] \\&= \frac{1}{2} \left[\frac{1}{6} \cdot \frac{1}{8} + \frac{1}{3} \cdot \frac{1}{64} + \frac{1}{3} \cdot \frac{1}{64} + \frac{1}{6} \cdot \frac{1}{8} \right] \\&= \frac{1}{2} \left[\frac{1}{48} + \frac{1}{96} + \frac{1}{48} \right] = \frac{1}{2} \left[\frac{2}{96} + \frac{1}{96} + \frac{2}{96} \right] \\&= \frac{1}{2} \left[\frac{5}{96} \right] = \frac{5}{192} = 0.026041666\end{aligned}$$

Caso 1

$$\begin{aligned}d_2 &= \int_0^1 \int_{\frac{1}{2}}^1 \left[(x - x_c)^2 + (y - y_c)^2 \right] f_2(x, y) \, dx \, dy \\&= \frac{1}{2} \int_0^1 \left[(x - x_c)^2 y + \frac{1}{3} (y - y_c)^3 \right] \Bigg|_{\frac{1}{2}}^1 dx \\&= \frac{1}{2} \int_0^1 \left[\left((x - x_c)^2 \cdot 1 + \frac{1}{3} (1 - y_c)^3 \right) - \left((x - x_c)^2 \frac{1}{2} + \frac{1}{3} \left(\frac{1}{2} - y_c \right)^3 \right) \right] dx \\&= \frac{1}{2} \int_0^1 \left[\frac{1}{2} \left((x - x_c)^2 + \frac{1}{3} \left((1 - y_c)^3 - \left(\frac{1}{2} - y_c \right)^3 \right) \right) \right] dx \\&= \frac{1}{2} \left[\frac{1}{6} \left((x - x_c)^3 + \frac{1}{3} \left((1 - y_c)^3 - \left(\frac{1}{2} - y_c \right)^3 \right) x \right) \right] \Bigg|_0^1\end{aligned}$$

Caso 1

$$= \frac{1}{2} \left[\frac{1}{6} \left((1 - x_c)^3 + \frac{1}{3} \left((1 - y_c)^3 - \left(\frac{1}{2} - y_c \right)^3 \right) \right) + \frac{1}{6} x_c^3 \right] = g_2(x_c, y_c)$$

$$\frac{\delta g_2(x_c, y_c)}{\delta x_c} = \frac{1}{2} \left[-\frac{1}{2} (1 - x_c)^2 + \frac{1}{2} x_c^2 \right] = 0$$

$$(1 - x_c)^2 - x_c^2 = 1 - 2x_c + x_c^2 - x_c^2$$

$$= 1 - 2x_c = 0$$

$$x_c = \frac{1}{2}$$

Caso 1

$$\frac{\delta g_2(x_c, y_c)}{\delta y_c} = \frac{1}{2} \left[-(1 - y_c)^2 + \left(\frac{1}{2} - y_c\right)^2 \right] = 0$$

$$-1 + 2y_c - y_c^2 + \frac{1}{4} - y_c + y_c^2$$

$$-\frac{3}{4} + y_c = 0$$

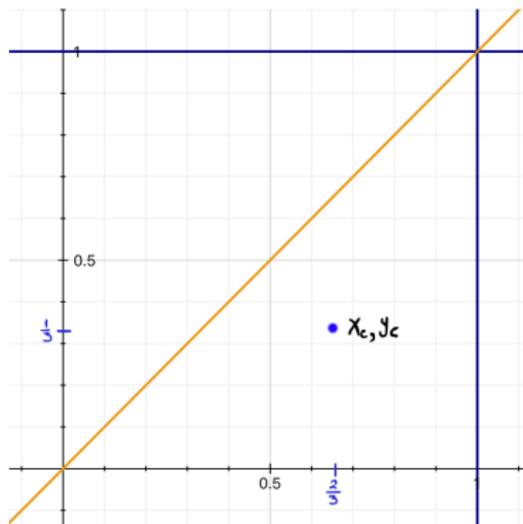
$$y_c = \frac{3}{4}$$

Mínimos

$$g_2(x_c, y_c) = \left(\frac{1}{2}, \frac{3}{4}\right)$$

$$\begin{aligned} g_2\left(\frac{1}{2}, \frac{3}{4}\right) &= \frac{1}{2} \left[\frac{1}{6} \left[\left(\frac{1}{2}\right)^3 + \frac{1}{3} \left(\left(1 - \frac{3}{4}\right)^3 - \left(\frac{1}{2} - \frac{3}{4}\right)^3 + \frac{1}{6} \cdot \left(\frac{1}{2}\right)^3 \right) \right] \right] \\ &= \frac{1}{2} \left(\frac{1}{6} \cdot \frac{1}{8} + \frac{1}{3} \cdot \frac{1}{64} - \frac{1}{3} \left(\frac{-1}{64} \right) + \frac{1}{6} \cdot \frac{1}{8} \right) = \frac{5}{192} \end{aligned}$$

Caso 2



$$d_3 = \int_0^1 \int_0^x ((x - x_c)^2 + (y - y_c)^2) f_3(x, y) dy dx$$

Caso 2

$$d_3 = \frac{1}{2} \int_0^1 \left[(x - x_c)^2 y + \frac{1}{3}(y - y_c)^3 \right] \Big|_0^x dx$$

$$= \frac{1}{2} \int_0^1 \left[(x - x_c)^2 x + \frac{1}{3}(x - y_c)^3 + \frac{1}{3}y_c^3 \right] dx$$

$$= \frac{1}{2} \int_0^1 \left[(x^2 - 2x x_c + x_c^2) x + \frac{1}{3}(x^3 - 3x^2 y_c + 3x y_c^2) + \frac{1}{3}y_c^3 \right] dx$$

$$= \frac{1}{2} \left[\frac{x^4}{4} - \frac{2}{3}x^3 x_c + \frac{x_c^2 x^2}{2} + \frac{1}{12}x^4 - \frac{x^3}{3}y_c + \frac{x^2}{2}y_c^2 - \frac{1}{3}y_c^3 x + \frac{1}{3}x y_c^3 \right] \Big|_0^1$$

Caso 2

$$\begin{aligned} &= \frac{1}{2} \left[\frac{1}{4} - \frac{2}{3}x_c + \frac{x_c^2}{2} + \frac{1}{12} - \frac{y_c}{3} + \frac{y_c^2}{2} - \frac{1}{3}y_c^3 + \frac{1}{3}y_c^3 \right] \\ &= \frac{1}{2} \left[\frac{4}{12} - \frac{2}{3}x_c + \frac{x_c^2}{2} - \frac{y_c}{3} + \frac{y_c^2}{2} \right] \\ &= g_3(x_c, y_c) \end{aligned}$$

Caso 2

$$\frac{\delta g_3(x_c, y_c)}{\delta x_c} = -\frac{2}{3} + \frac{2x_c}{2} = 0$$

$$x_c = \frac{2}{3}$$

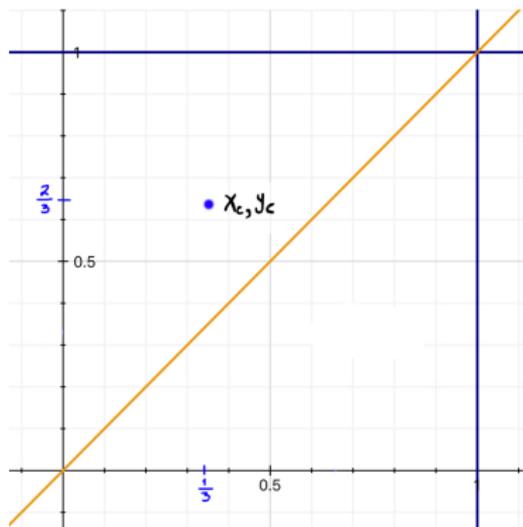
$$\frac{\delta g_3(x_c, y_c)}{\delta y_c} = -\frac{1}{3} + y_c = 0$$

$$y_c = \frac{1}{3}$$

Caso 2

$$\begin{aligned}g_3\left(\frac{2}{3}, \frac{1}{3}\right) &= \frac{1}{2} \left[\frac{4}{12} - \frac{4}{9} + \frac{\frac{4}{9}}{2} - \frac{\frac{1}{3}}{3} + \frac{\frac{1}{9}}{2} \right] \\&= \frac{1}{2} \left[\frac{4}{12} - \frac{4}{9} + \frac{4}{18} - \frac{1}{9} + \frac{1}{18} \right] \\&= \frac{1}{2} \left[\frac{4}{12} - \frac{5}{9} + \frac{5}{18} \right] \\&= \frac{1}{2} \left[\frac{4}{12} - \frac{5}{18} \right] = \frac{1}{2} \left[\frac{4}{2 \cdot 6} - \frac{5}{3 \cdot 6} \right] = \frac{1}{2} \left[\frac{4 \cdot 3 \cdot 6 - 5 \cdot 2 \cdot 6}{2 \cdot 3 \cdot 6^2} \right] \\&= \frac{1}{2} \left[\frac{3 \cdot 4 - 2 \cdot 5}{6^2} \right] = \frac{1}{2} \left[\frac{2}{36} \right] = \frac{1}{36} = 0.0277\end{aligned}$$

Caso 2



$$d_4 = \int_0^1 \int_0^y \left((x - x_c)^2 + (y - y_c)^2 \right) f_4(x, y) dx dy$$

Caso 2

$$\begin{aligned} &= \frac{1}{2} \int_0^1 \left[\frac{1}{3} (x - x_c)^3 + (y - y_c)^2 x \right] \Big|_0^y dy \\ &= \frac{1}{2} \int_0^1 \left[\frac{1}{3} (y - x_c)^3 + (y - y_c)^2 y - \frac{1}{3} x_c^3 \right] dy \\ &= \frac{1}{2} \int_0^1 \left[\frac{1}{3} (y^3 - 3y^2 x_c + 3y x_c^2 - x_c^3) + \frac{1}{3} x_c^3 + (y^3 - 2y^2 y_c + y_c^2 y) \right] dy \\ &= \frac{1}{2} \left[\frac{1}{12} y^4 - \frac{y^3}{3} x_c + \frac{y^2}{2} x_c^2 - \frac{1}{3} x_c^3 y + \frac{1}{3} x_c^3 y + \frac{y^4}{4} - \frac{2}{3} y^3 y_c + \frac{y_c^2 y^2}{2} \right] \Big|_0^1 \\ &= \frac{1}{2} \left[\frac{1}{12} - \frac{x_c}{3} + \frac{x_c^2}{2} + \frac{1}{4} - \frac{2}{3} y_c + \frac{y_c^2}{2} \right] \end{aligned}$$

Caso 2

$$= \frac{1}{2} \left[\frac{4}{12} - \frac{x_c}{3} + \frac{x_c^2}{2} - \frac{2}{3} y_c + \frac{y_c^2}{2} \right] = g_4(x_c, y_c)$$

$$\frac{\delta g_4(x_c, y_c)}{\delta x_c} = -\frac{1}{3} + \frac{2x_c}{2} = 0$$

$$\Rightarrow x_c = \frac{1}{3}$$

$$\frac{\delta g_4(x_c, y_c)}{\delta y_c} = -\frac{2}{3} + \frac{2y_c}{2} = 0$$

$$\Rightarrow y_c = \frac{2}{3}$$

$$g\left(\frac{1}{3}, \frac{2}{3}\right) = \frac{1}{2} \left[\frac{4}{12} - \frac{1}{9} + \frac{1}{18} - \frac{4}{9} + \frac{4}{18} \right] = \frac{1}{36} = 0.02777$$

Conclusión

Como : $(g_1 = g_2 < g_3 = g_4)$

Podemos decir que la partición de g_1 y g_2 es mejor que la partición de g_3 y g_4 .

Agrupamiento (Clustering) Usando Cuantización Vectorial

Dado un conjunto de N_v vectores de N dimensiones, $\underline{X}_j = (x_{j,1}, x_{j,2}, \dots, x_{j,N})$, con $j = 1, \dots, N_v$, que representan características de una señal, un conjunto de centroides es encontrado que mejor representan los vectores.

Una colección de centroides es llamado *libro de codificación* o *CodeBook*.

Para encontrar los centroides óptimos que particionen un espacio se utiliza el algoritmo modificado de Lloyd o mejor conocido como Linde-Buzo-Gray (LBG).

Algoritmo de Lloyd generalizado (LBG)

Algoritmo de Lloyd generalizado o LBG

- a) Dado un alfabeto, $\gamma_m = \{\underline{\gamma}_j\}$, encuentre la partición óptima de las regiones de cuantización, use la condición del vecino más cercano para formar éstas. $R = \{ \underline{x} : d(\underline{x}, \underline{y}_i) \leq d(\underline{x}, \underline{y}_j) \} \forall j \neq i$.
- b) Usando la condición de Centroide, encuentre el γ_{m+1} el alfabeto de reproducción óptimo para el conjunto de agrupamientos encontrado.
- c) Calcular la distorsión promedio para γ_{m+1} , si ha cambiado por una cantidad muy pequeña desde la última iteración, alto. En caso contrario poner $m + 1$ y regresar al paso a)

Algoritmo de Linde-Buzo-Gray

1. Calcular el primer centroide:

Dados un conjunto de vectores \underline{X}_i de N dimensiones:

$$\underline{X}_1 = [x_{1,1}, x_{1,2}, \dots, x_{1,N}]$$

$$\underline{X}_2 = [x_{2,1}, x_{2,2}, \dots, x_{2,N}]$$

\vdots

$$\underline{X}_i = [x_{i,1}, x_{i,2}, \dots, x_{i,N}]$$

\vdots

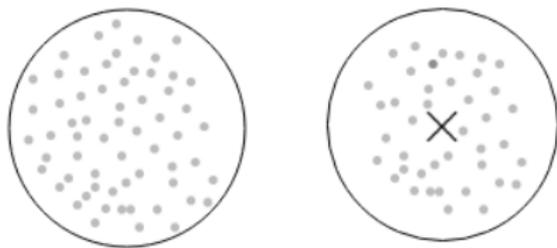
$$\underline{X}_{N_v} = [x_{N_v,1}, x_{N_v,2}, \dots, x_{N_v,N}]$$

Algoritmo de Linde-Buzo-Gray

El primer centroide se calcula, con $i = 1$:

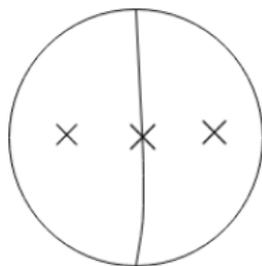
$$\underline{C}_i = \frac{1}{N_v} \sum_{j=1}^{N_v} \underline{X}_j$$

$$\underline{C}_i = [C_{i,1}, C_{i,2}, \dots, C_{i,N}] = \left[\frac{1}{N_v} \sum_{j=1}^{N_v} x_{1,j}, \dots, \frac{1}{N_v} \sum_{j=1}^{N_v} x_{N,j} \right]$$



Algoritmo de Linde-Buzo-Gray

2. Generar dos centroides nuevos a partir del centroide \underline{C}_i o centroides anteriores:



$$\underline{C}_{i+1} = \underline{C}_i * \epsilon_1$$

$$\underline{C}_{i+2} = \underline{C}_i * \epsilon_2$$

$$\epsilon_1 = 0.9999$$

$$\epsilon_2 = 1.0001$$

Algoritmo de Linde-Buzo-Gray

3. Agrupe los vectores de entrenamiento en dos grupos de acuerdo al centroide más cercano.

$$d_1 = \text{distancia} (\underline{X}_j, \underline{C}_{i+1})$$

$$d_2 = \text{distancia} (\underline{X}_j, \underline{C}_{i+2})$$

Algoritmo de Linde-Buzo-Gray

Si $d_1 < d_2$

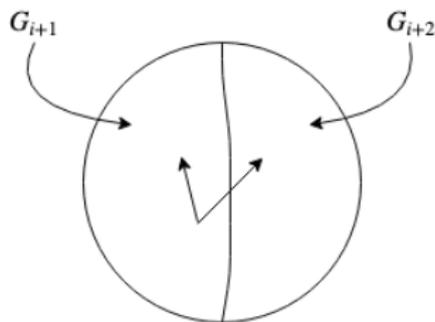
Se asigna el vector \underline{X}_j a la region G_{i+1}

$$\underline{X}_j \rightarrow G_{i+1}$$

en caso contrario asignarlo a la región G_{i+2} :

$$\underline{X}_j \rightarrow G_{i+2}$$

Esta operación se hace para todos los vectores, $1 \leq j \leq N_v$



Algoritmo de Linde-Buzo-Gray

4. Recalcular los centroides

El proceso de separar los vectores en regiones y encontrar nuevos centroides se debe repetir en varias iteraciones. Para cada iteración t se calcula la distancia global, empezando con:

$$D_g^t = 0$$

Sumar la distancia global con la menor distancia calculada para todos los vectores \underline{X}_j , $1 \leq j \leq N_v$

$$d_1 = \text{distancia}(\underline{X}_j, \underline{C}_{i+1})$$

$$d_2 = \text{distancia}(\underline{X}_j, \underline{C}_{i+2})$$

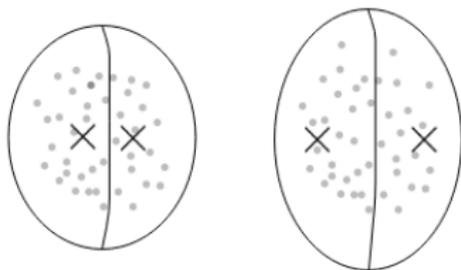
$$D_g^t = D_g^t + \min(d_1, d_2)$$

Algoritmo de Linde-Buzo-Gray

Con los vectores separados se vuelven a calcular los centroides:

$$\underline{C}_{i+1} = \frac{1}{\text{Num. Vect. } G_{i+1}} \sum_{j=1}^{\text{Num. Vect. } G_{i+1}} \underline{X}_j$$

$$\underline{C}_{i+2} = \frac{1}{\text{Num. Vect. } G_{i+2}} \sum_{j=1}^{\text{Num. Vect. } G_{i+2}} \underline{X}_j$$



Algoritmo de Linde-Buzo-Gray

Si se cumple que la diferencia de distancia entre una iteración actual y la anterior:

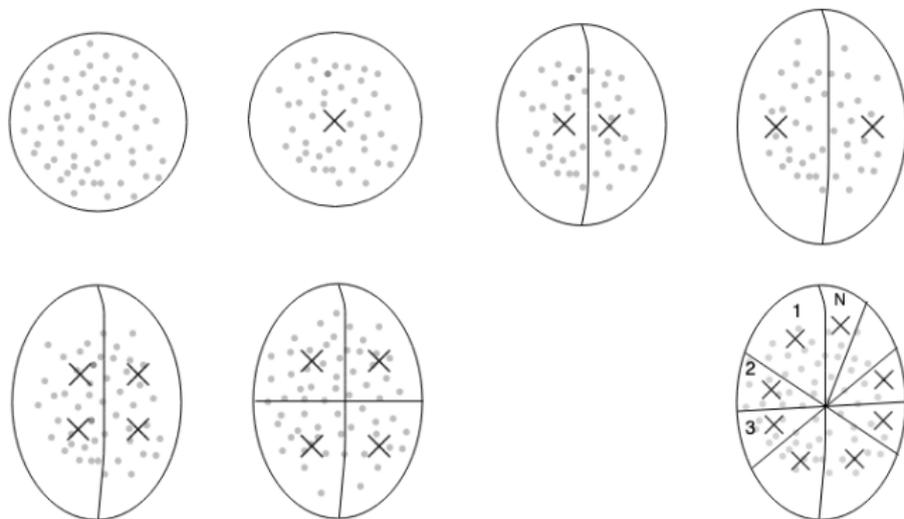
$$|D_g^t - D_g^{t-1}| \geq \epsilon$$

ir al punto 3.

Este proceso se repite varias veces hasta que se establezcan los centroides. En caso contrario ir al punto 2 para modificar los centroides por una pequeña cantidad.

Algoritmo de Linde-Buzo-Gray

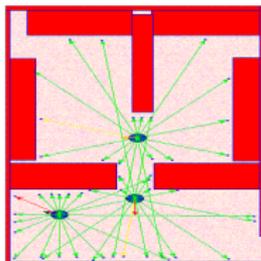
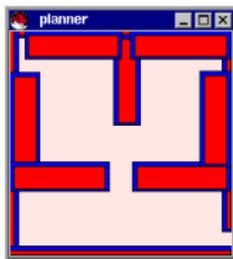
Se termina cuando se han obtenido las particiones necesarias.



Algoritmo de Linde-Buzo-Gray

Ejemplo:

Se coloca a un robot en un medio ambiente y éste empieza a navegar tomando muestras con su sensores. Se asume que se conoce la posición del robot en todo momento.

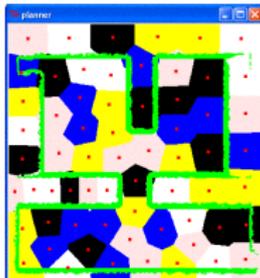


Algoritmo de Linde-Buzo-Gray

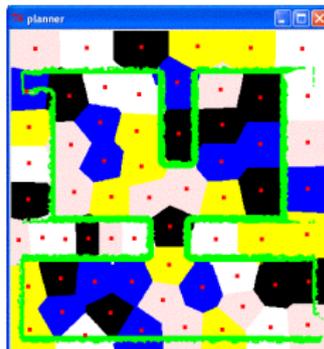
Se genera una nube de puntos del medio ambiente, separando el espacio libre del espacio ocupado.



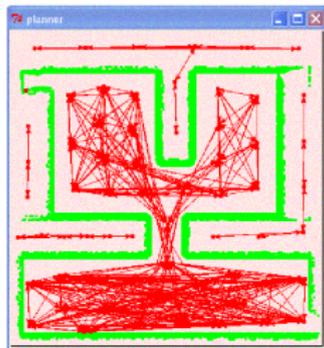
Se cuantiza el espacio libre, es decir puntos en los cuales no se detectó un obstáculo.



Algoritmo de Linde-Buzo-Gray



Uniendo los centroides se encuentra el mapa topológico del lugar.



Reconocimiento de voz

En el reconocimiento de voz utilizando cuantizadores vectoriales cada palabra se representa por un cuantizador vectorial. Después la palabra de entrada es cuantizada usando cada uno de los cuantizadores y el cuantizador que de el valor de distancia más pequeño es el patrón de reconocimiento.



r_l



Algoritmo de Lloyd generalizado (LBG)

En un principio cada cuantizador estaría formado por los centroides que representan el modelo de la señal de voz que se desea reconocer.

Recordando que el modelo del tracto vocal se definió anteriormente como:

$$H(z) = \frac{\sigma}{\sum_{i=0}^M a_i \cdot z^{-i}}$$

La relación que existe entre los pesos \underline{w} encontrados por el filtro de Wiener, filtros adaptables LMS, u otros métodos como el algoritmo de Levinson y Durbin, algoritmos genéticos, etc, y los coeficientes del modelo \underline{a} es la siguiente:

Sea $\underline{w}^T = [w_1 \quad w_2 \quad \dots \quad w_p]$, entonces $\underline{a}^T = [1 - \underline{w}^T]$

$$\underline{a}^T = [a_0 \quad a_1 \quad \dots \quad a_M] = [1 \quad -w_1 \quad -w_2 \quad \dots \quad -w_M]$$

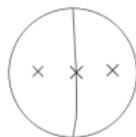
Algoritmo de Lloyd generalizado (LBG)

Se desearía que cada cuantizador que representa una señal de voz, contara con N regiones de vectores de LPC. El problema de tener directamente los centroides $\underline{C}_i = (c_{i,1}, c_{i,2}, \dots, c_{i,M})$ como una relación directa con los vectores \underline{A}_{j_s} que representan el modelo de la señal de voz, es que, una vez que se calcule el promedio entre ellos para obtener el primer centroide y después al ser modificado para obtener dos centroides, esta modificación puede hacer que los centroides resultantes puedan ser inestables. Es decir: el primer centroide se calcula, utilizando todos los vectores \underline{A}_{j_s} de cada uno de los bloques de las señales de voz:

$$\underline{C}_1 = \frac{1}{N_v} \sum_{j=1}^{N_v} \underline{A}_j$$

Algoritmo de Linde-Buzo-Gray

Dos centroides nuevos a partir del primer centroide \underline{C}_1 :



$$\underline{C}_2 = \underline{C}_1 * \epsilon_1$$

$$\underline{C}_3 = \underline{C}_1 * \epsilon_2$$

$$\epsilon_1 = 0.9999$$

$$\epsilon_2 = 1.0001$$

Cada uno de estos centroides representa el modelo de la señal de voz, por ejemplo para el centroide \underline{C}_2 , el modelo es el siguiente:

$$H(z) = \frac{\sigma}{\sum_{i=0}^M c_{2,i} \cdot z^{-i}}$$

Algoritmo de Lloyd generalizado (LBG)

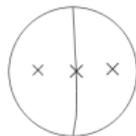
Al desarrollar la sumatoria del denominador se genera un polinomio de orden M , encontrando las raíces del polinomio puede suceder que algunas de éstas quedan fuera del círculo unitario, teniéndose un sistema que es inestable. Por lo tanto en lugar de usar los coeficientes \underline{A}_{is} lo que se usan son los vectores de correlaciones \underline{r}_{is} de las señales de voz. Para sacar los promedios y las modificaciones de los centroides se usan los vectores de correlaciones \underline{r}_{is} ,

$$\underline{c}_1 = \frac{1}{N_v} \sum_{j=1}^{N_v} \underline{r}_j$$

Algoritmo de Lloyd generalizado (LBG)

$$\underline{C}_2 = \underline{C}_1 * \epsilon_1$$

$$\underline{C}_3 = \underline{C}_1 * \epsilon_2$$



Pero para separar los vectores en regiones, al tener que utilizar la distancia de Itakura Saito, es necesario encontrar los coeficientes \underline{A}_{iS} a partir de los centroides de correlaciones de la señal de voz, ya que esta distancia requiere los vectores de correlaciones de los coeficientes \underline{A}_{iS} , es decir la correlación corta $r_a(i)$ de los coeficientes:

$$r_a(i) = \sum_{j=0}^{M-i} a_j a_{j+i}, \quad 0 \leq i \leq M$$

Algoritmo de Lloyd generalizado (LBG)

Con la distancia de Itakura Saito se encuentra, para cada vector de correlaciones \underline{r}_{is} , a que centroide se parece más:

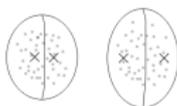
$$d_{is}(S_i(\omega), \hat{S}_j(\omega)) = \frac{1}{\sigma^2} [r_i(0)r_{aj}(0) + 2 \sum_{n=1}^M r_i(n)r_{aj}(n)]$$

De esta forma se separan los vectores de correlaciones en dos regiones y una vez separados, para encontrar los nuevos centroides se procede de la siguiente forma:

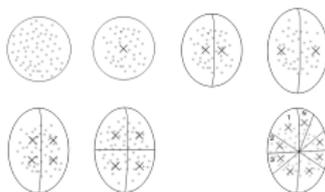
$$\underline{C}_{i+1} = \frac{1}{\text{Num. Vect. } G_{i+1}} \sum_{j=1}^{\text{Num. Vect. } G_{i+1}} \underline{r}_j$$
$$\underline{C}_{i+2} = \frac{1}{\text{Num. Vect. } G_{i+2}} \sum_{j=1}^{\text{Num. Vect. } G_{i+2}} \underline{r}_j$$

Algoritmo de Linde-Buzo-Gray

Se continúa con este proceso hasta que los centroides ya estén estabilizados.



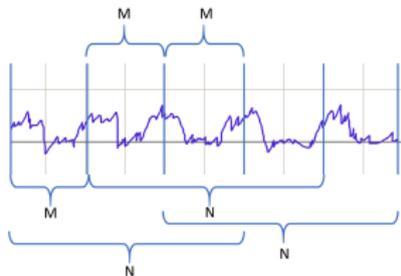
Se siguen modificando los centroides hasta que se tengan el número de regiones deseado.



Una vez terminado el proceso de generar el cuantizador, entonces lo que se guarda como patrón, es decir los centroides, que son los vectores de correlaciones de los coeficientes \underline{A}_{iS} , es decir la correlación corta $r_a(i)$ de los coeficientes.

Reconocimiento de comandos de voz

Para hacer el reconocimiento de un comando de voz, lo primero que se hace es aplicar técnicas de procesamiento de señales a la señal de entrada, separando la señal en bloques y encontrando la función de correlaciones r_l de cada bloque.



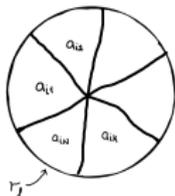
Reconocimiento de comandos de voz

Entonces se comparan los vectores de los centroides \underline{C}_{ij} de cada cuantizador con el vector de correlación r_l , donde l es el número de bloque, y se escoge para cada cuantizador el centroide con la menor distorsión. Guardando la distancia mínima d_{ij}^{min} de cada cuantizador i para todos los bloques de la señal de voz.

$$d_{ij}^h = r_a^{ih}(0)r_l(0) + 2 \sum_{k=1}^M r_a^{ih}(k)r_l(k)$$

$$d_{ij}^{min} = \{d_{ij}^h < d_{ij}^b\} \quad b \neq h$$

donde el vector r_a^{ih} corresponde al centroide C_h del cuantizador i .



Reconocimiento de comandos de voz



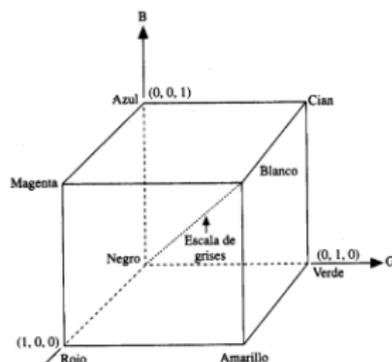
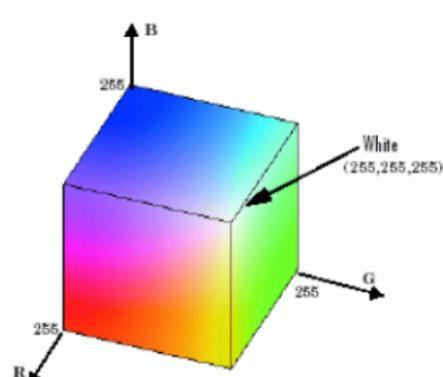
La palabra reconocida es j si:

$$\sum_{k=0}^L d_j k^{min} < \sum_{k=0}^L d_i k^{min}$$
$$\forall j \neq i$$

donde L es el número de bloques total de la señal de voz de entrada.

Reconocimiento de Imágenes Utilizando Cuantizadores Vectoriales

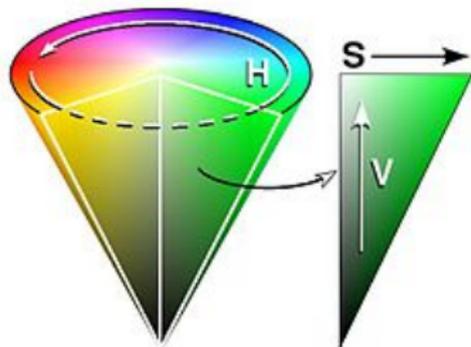
Para el reconocimiento de imágenes usando cuantizadores vectoriales es necesario encontrar vectores que mejor representen éstas. Para encontrar los vectores que representan las imágenes, cada color aparece con sus componentes espectrales primarias de rojo, verde y azul. Está basado en un sistema de coordenadas cartesianas y su subespacio de color está definido como un cubo unitario, de manera que, los valores de cada punto de este espacio están normalizados de $[0, 1]$.



Espacio de color HSV

Para distinguir un color de otro deben considerarse el tono (hue), la saturación (saturation) y el brillo o valor (value), estos valores juntos forman el espacio HSV por sus siglas en ingles. A partir de los valores RGB de la imagen se pueden encontrar los valores HSV

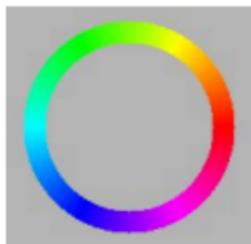
1. El Brillo o intensidad o valor V .



$$V = \max(R, G, B)$$

Espacio de color HSV

2. El tono H (hue) está asociado con la longitud de onda dominante en una mezcla de ondas luminosas.



El valor del tono H se encuentra de la siguiente forma:

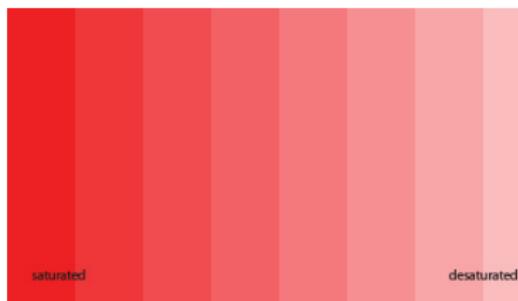
$$H = 60(G - B)/(V - \min(R, G, B)), \quad \text{si } V = R$$

$$H = 120 + 60(B - R)/(V - \min(R, G, B)) \quad \text{si } V = G$$

$$H = 240 + 60(R - G)/(V - \min(R, G, B)) \quad \text{si } V = B$$

Espacio de color HSV

3. La saturación S es la pureza relativa o cantidad de luz blanca mezclada con un tono. Los colores puros del espectro están completamente saturados. Los colores como el rosa (rojo mas blanco) y el lavanda (azul y blanco) están menos saturados, siendo su grado de saturación inversamente proporcional a la cantidad de luz blanca añadida.



$$S = (V - \min(R, G, B)) / V, \quad \text{si } V \neq 0$$

$$S = 0, \quad \text{si } V = 0$$

Filtros para encontrar contornos

Los filtros se utilizan para encontrar o resaltar algunas características de las imágenes. Por ejemplo, si se tiene una imagen como la que se muestra a continuación y se quisiera encontrar sus contornos horizontales y verticales se usarían los filtros de Sobel para lograr ésto.



Filtros para encontrar contornos

El siguiente nucleo es el del filtro Sobel que sirve para encontrar contornos horizontales a través de convolucionar este nucleo con la imagen:

$$\textit{Sobel Horizontal} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$



Filtros para encontrar contornos

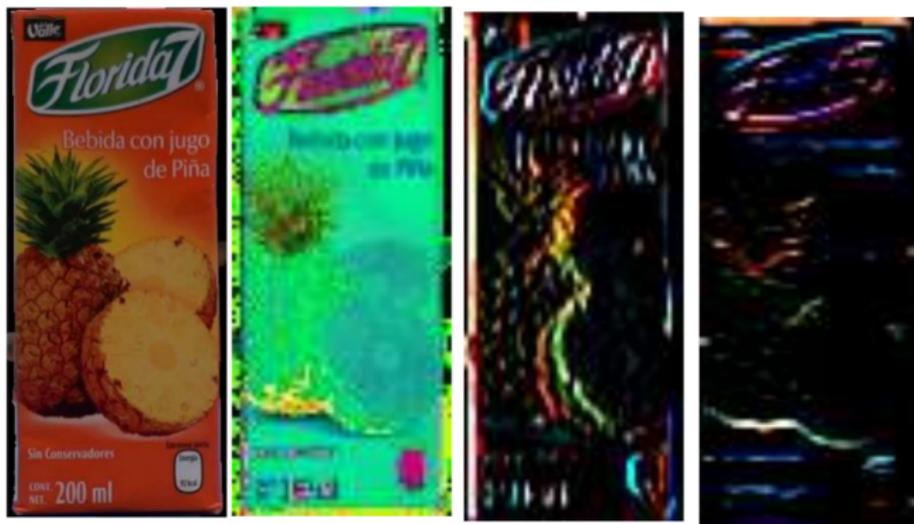
El siguiente nucleo del filtro Sobel el cual sirve para encontrar contornos verticales a través de convolucionar este nucleo con la imagen:

$$\textit{Sobel Vertical} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$



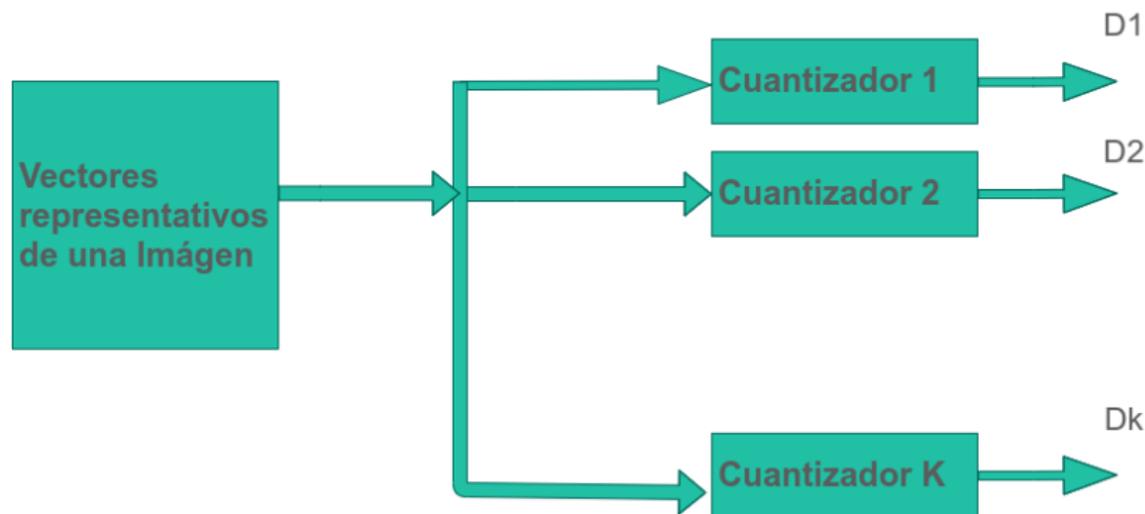
Reconocimiento de Imágenes Utilizando Cuantizadores Vectoriales

Se crean cuantizadores vectoriales de cada uno de los objetos a reconocer con varias imágenes de entrenamiento de éstos, utilizando vectores representativos de las imágenes como pueden ser sus valores de RGB, HSV, contornos verticales y horizontales, etc.



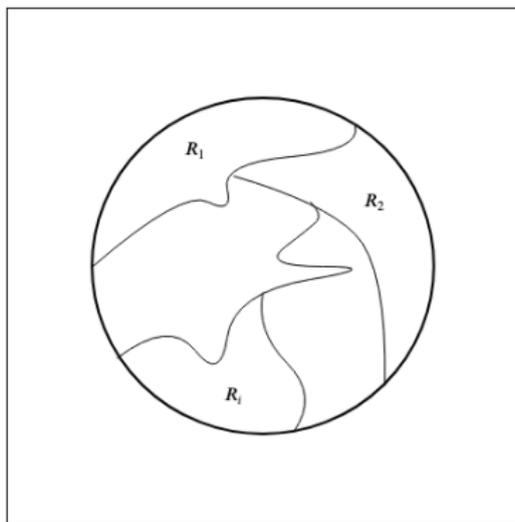
Reconocimiento de Imágenes Utilizando Cuantizadores Vectoriales

Una vez creado los cuantizadores para cada objeto, se hace el reconocimiento de un objeto al tratar de cuantizar su imagen con cada uno de ellos.



Reconocimiento de Imágenes Utilizando Cuantizadores Vectoriales

Cada cuantizador i cuenta con N regiones con centroides \underline{C}_{ij} , los cuales representan todos los vectores de los objetos, entonces se comparan estos centroides con cada uno de los vectores \underline{v}_k característicos de la imagen a reconocer y se escoge el vector centroide \underline{C}_{ij} con la menor distancia $d_{\underline{v}_k, \underline{C}_{ij}}$.



Reconocimiento de Imágenes Utilizando Cuantizadores Vectoriales

Para comparar los centroides \underline{C}_{ik} , $k = 1, \dots, N$, con los vectores de la imagen \underline{v}_l , $l = 1, \dots, L$, donde L es el número de vectores en la imagen, se puede utilizar la distancia euclidiana:

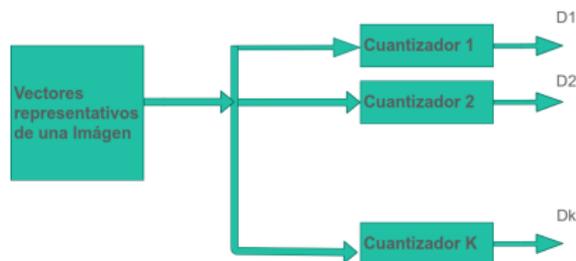
$$d_{\underline{v}_l, \underline{C}_{ik}} = |\underline{v}_l - \underline{C}_{ik}|$$

El vector \underline{v}_l es más parecido al centroide \underline{C}_{ij} si:

$$d_{\underline{v}_l, \underline{C}_{ij}}^{min} \leftarrow \{d_{\underline{v}_l, \underline{C}_{ij}} < d_{\underline{v}_l, \underline{C}_{ik}}\}, \quad l = 1, \dots, L$$

Reconocimiento de Imágenes Utilizando Cuantizadores Vectoriales

El objeto j es reconocido si: $D_j < D_i, \quad \forall j \neq i$



Donde las D_i s son las distancias totales mínimas de comparar cada uno de los vectores \underline{v}_k de la imagen con los centroides \underline{C}_{ij} del cuantizador i .

$$D_i = \sum_{k=1}^L d_{\underline{v}_k, \underline{C}_{ij}}^{\min}$$