

# La matriz de cuantización en la cuantización perceptual

[latexpage]De entre las etapas de la codificación de imagen y vídeo, la cuantización es la única en la que se produce una pérdida real de información.

Los codificadores basados en bloques utilizan una matriz de cuantización QM (Quantization Matrix) para reducir la precisión de los coeficientes resultantes de la etapa de transformación al dominio de la frecuencia, habitualmente utilizando la transformada DCT.

Suponiendo un bloque de  $8 \times 8$  de coeficientes  $\begin{aligned} F(u,v) \end{aligned}$  resultantes de la transformada DCT o similar, la matriz de cuantización define la cantidad de reducción que se va a aplicar a cada coeficiente para reducir el bit-rate resultante en la compresión.  $u, v$  son la frecuencia horizontal y vertical respectivamente. Cuando la cuantización es uniforme, a todos los coeficientes se les aplica el mismo divisor y cuando no lo es, para cada coeficiente la matriz de cuantización tiene divisores diferentes. La cuantización se regiría por la ecuación ([\ref{eq:quantization}](#)) y la decuantización por la ecuación ([\ref{eq:dequantization}](#))

$\begin{equation} \label{eq:quantization}$

$$F_Q(u,v) = \text{Round}\left(\frac{F(u,v)}{Q(u,v)}\right)$$

$\end{equation}$

$\begin{equation} \label{eq:dequantization}$

$$F'(u,v) = F_Q(u,v) \times Q(u,v)$$

$\end{equation}$

Cada uno de los valores  $Q(u,v)$  de la matriz de cuantización es un entero entre 1 y 255 que especifica el tamaño del *quantization step* ( $QStep$ ) a aplicar en la frecuencia  $(u,v)$ . El error máximo de cuantización en la reconstrucción de  $F(u,v)$  es por tanto  $Q(u,v)/2$ .

**El objetivo es reducir la precisión de los coeficientes que son perceptualmente insignificantes.**

Fundamentalmente hay dos formas de definir la matriz de cuantización:

1. Utilizando la teoría de rate-distortion.
2. Utilizando experimentos psico-visuales basados en el HVS.

1) Se obtiene una QM específica para la imagen asignando, de los bits disponibles, cierto número de bits a cada coeficiente DCT utilizando un criterio rate-distortion.

Normalmente a los coeficientes de baja frecuencia se les asigna más bits dada la compactación de energía de la DCT. El problema de esta técnica es que la QM generada depende de la imagen y tiene un coste computacional a tener en cuenta. Habitualmente se implementa un algoritmo rate-distortion adaptativo que genera la QM para cada imagen.

2) La idea es determinar el umbral  $T$  para cada coeficiente DCT de forma que idealmente, las distorsiones bajo ese umbral no se perciban. Normalmente los  $QSteps$  para coeficientes de alta frecuencia serán mayores que los de baja, lo que producirá mayores distorsiones para las altas frecuencias que para las bajas, lo que concuerda con la sensibilidad del HVS.

---

# GGD – Generalized Gaussian Distribution

*In probability theory, the **normal (or Gaussian) distribution** is a very common continuous probability distribution. Normal distributions are important in statistics and are often used in the natural and social sciences to represent real-valued random variables whose distributions are not known. [Wikipedia](#)*

[latexpage]

Es decir Normal Distribution y Gaussian Distribution es lo mismo.

Generalized viene de que via un parámetro que se añade a la distribución normal, la distribución GGD puede adoptar distintas formas. Hay dos versiones, trataremos la versión 1 que es la que se utiliza para errores, imagenes, etc...

*GGD: Known also as the exponential power distribution, or the generalized error distribution, this is a parametric family of symmetric distributions. It includes all normal and Laplace distributions, and as limiting cases it includes all continuous uniform distributions on bounded intervals of the real line. [Wikipedia](#)*

La PDF (Probability Density Function) está definida con distintas formulas depende donde lo mires, pero depende de un parámetro para obtener su aspecto.

\[

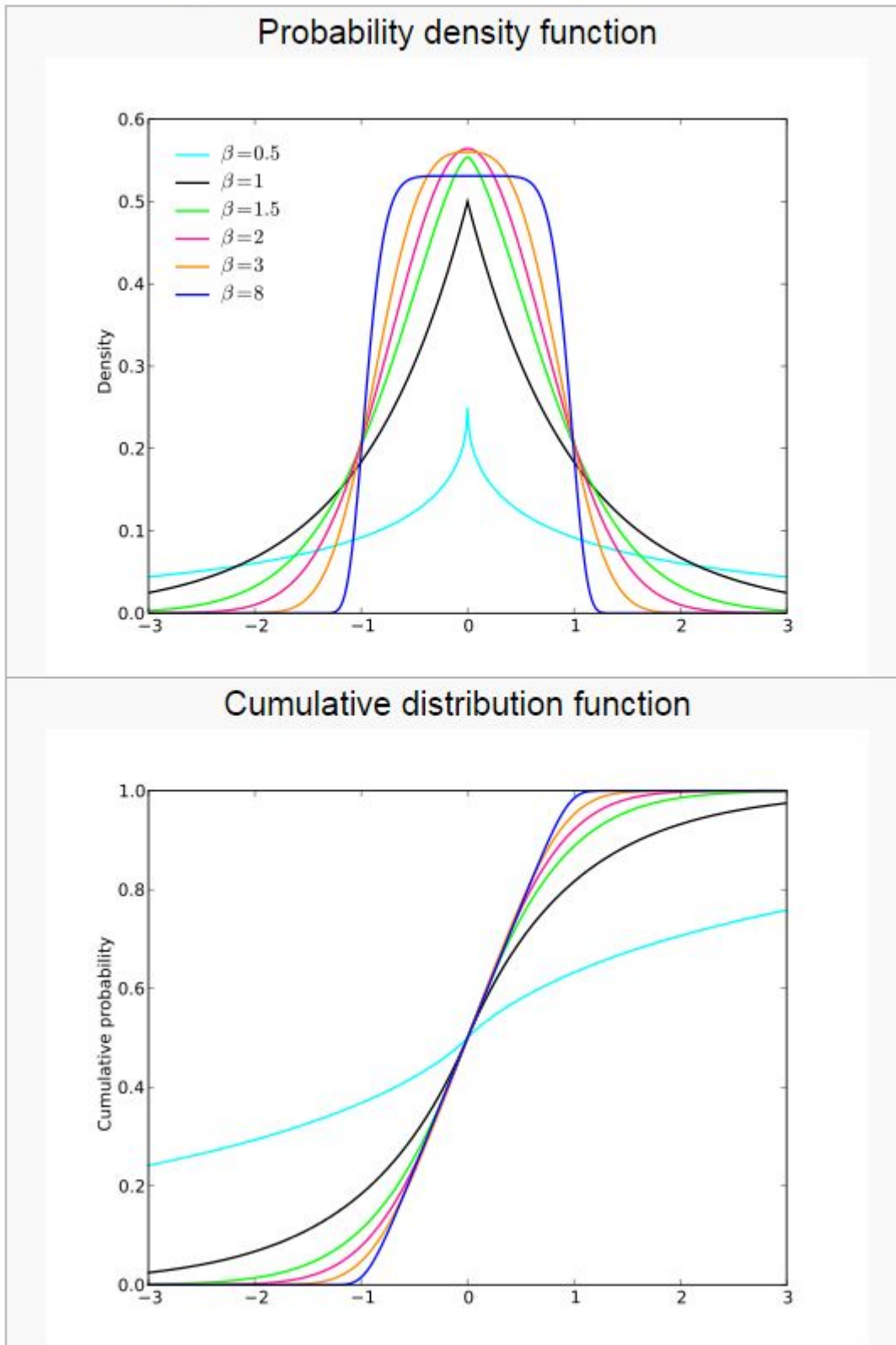
$$GG(x, \alpha, \beta) = \frac{\beta}{2\alpha\Gamma}$$

$(\frac{1}{\beta}) e^{-\frac{|x-\mu|}{\alpha} \beta}$

\]

donde  $\Gamma$  es la [función Gamma](#)

## Generalized Normal (version 1)



Siguiendo la formula el parámetro  $\beta=2$  hace que la distribución sea la Normal cuya media es  $\mu$  y su varianza es  $\frac{\alpha^2}{2}$ .

Si  $\beta=1$  la distribución será una Laplaciana.

Y si  $\beta \rightarrow \infty$  tenemos una distribución uniforme centrada en  $\mu$  y con amplitud entre  $[-\alpha, \alpha]$

La varianza de una GGD es  $\sigma^2 = \frac{\alpha^2 \Gamma(\frac{3}{\beta})}{\Gamma(\frac{1}{\beta})}$

En el caso de  $\beta=2$  (distribución normal) la varianza es  $\frac{\alpha^2}{2}$  como hemos dicho porque las funciones  $\Gamma(\frac{3}{\beta})$  y  $\Gamma(\frac{1}{\beta})$  toman valores  $\frac{1}{2}\sqrt{\pi}$  y  $\sqrt{\pi}$  respectivamente. Estos valores particulares de Gamma se pueden ver en [Valores Particulares de Gamma](#)

Por tanto, si conozco la varianza de una distribución de coeficientes por ejemplo, y sabiendo que la distribución de los coeficientes es una Laplaciana ( $\beta=1$ ), puedo despejar  $\alpha$  y con esto tendría la PDF de la distribución en concreto.

---

## Calculo de los CSF Weights

De cara a la escritura de la Tesis, en esta entrada voy a explicar cómo se calculan correctamente los CSF Weights basándose en lo expuesto por Beegan.

Daré los Peak CSF Weights por niveles y por subbandas.

En la versión que se ha utilizado en todos los papers hasta ahora se ha utilizado el código en `GetCSFWeights.m` y en `GetCSFSbWeights.m` para niveles y subbandas. En la versión de subbandas se hacía un escalado proporcional para mantener los valores picos de cada nivel y para ello se utilizaba un

(inventado) SF ScalingFactor.

Esto se ha entendido y se ha corregido, de forma que ahora se hace bien en la versión de código en `GetCSFWeightsNew.m` que da tanto los valores nuevos por niveles y los valores por subbandas. Este es el método que voy a explicar aquí.

Esta es la línea de llamada

```
[lev_csfw,          sb_csfw,          sbmed_csfw,
fmax]=GetCSFWeightsNew(Fmax_pxdg,nlevels,flat)
```

**Fmax\_pxdg** es la Frecuencia máxima en pixels por degree que tiene la curva CSF en su extremo derecho.

**nlevels** es el número de niveles

**flat** indica si llegado a la frecuencia máxima se colocan todos los pesos al máximo, como propone Nadenau para lo que él denomina el peor caso.

Se calcula la curva csf con el siguiente código, (es la función csf)

```
% Calculo de la CSF en un array sin necesidad de llamar a
funcion csf() o H()
a=-(0.114.*f).^1.1;
csf=2.6*(0.0192+0.114*f).*exp(a);
```

Se define un rango de `0:0.1:fmax` que genera 6400 bins, es decir 6400 puntos para el cálculo de la CSF.

Para cada subbanda wavelet, se definen los bins mayor y menor que la delimitan y se define también el valor de la csf para esos bins.