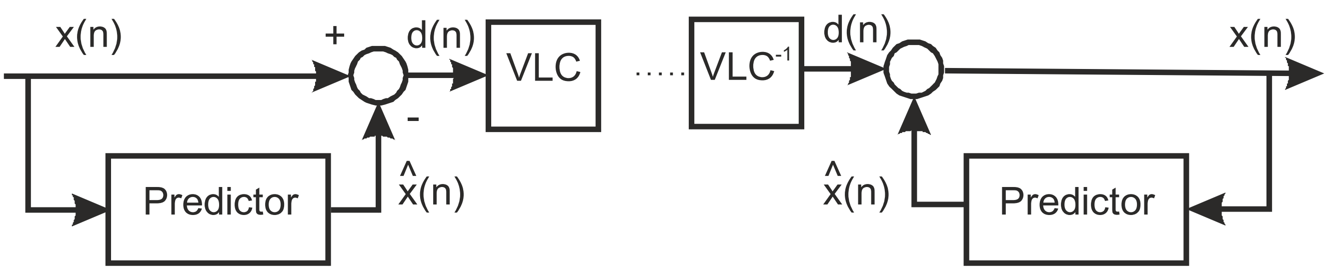
TV: TeleVisión – Plan 2010

Codificación diferencial

Contenido

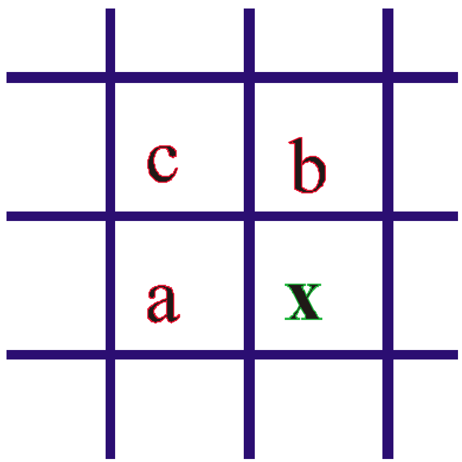
Codificación diferencial

- Las señales reales incluyen un alto grado de redundancia.
- Las transiciones bruscas son extrañas, y la correlación entre elementos adyacentes se aproxima a la unidad.
- Los sistemas diferenciales intentan aprovechar dicha continuidad para comprimirlas, manteniendo una calidad aceptable.



Codificación diferencial sin pérdidas

Codificacion diferencial sin pérdidas (JPEG)



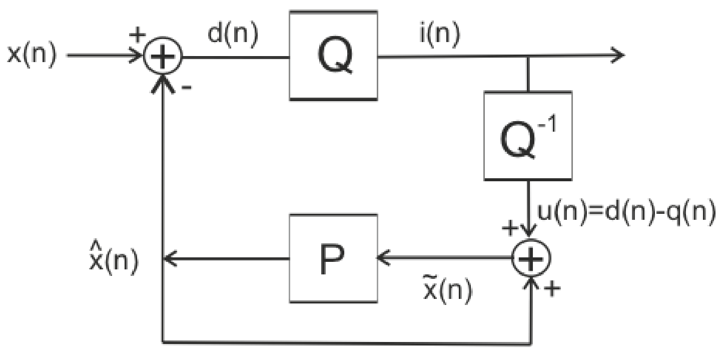
Valor Selección	Predicción
0	ninguna
1	a
2	b
3	c
4	a+b-c
5	a+(b-c)/2
6	b+(a-c)/2
7	(a+b)/2

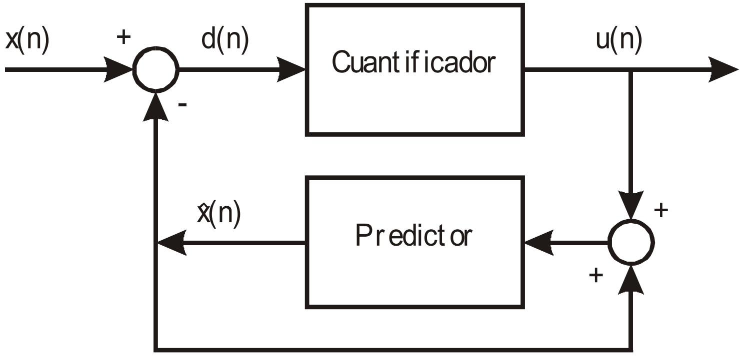
Codificación diferencial

- x(n) representa muestras de una señal
- d(n) representa la señal diferencia, obtenida al substraer a x(n) una predicción, $\hat{x}(n)$, de la señal de entrada.
- La señal diferencia es cuantificada, obteniendose u(n)=d(n)-q(n), donde q(n) representa el error de cuantificación.

Se denomina ganancia de predicción a la relación entre la potencia (AC) de la señal de entrada y la potencia (AC) de la señal de diferencias:

$$G_p = \frac{\sigma_x^2}{\sigma_d^2}$$





Codificación diferencial

La reducción de la tasa binaria se consigue mediante:

- ★ La codificación de longitud variable (VLC, p.e. Huffman) de las señales diferencia
- ★ La reducción de la calidad de la señal (pérdidas):
  - ✸ Cuantificación de la señal diferencia
  - ✸ Reducción de la resolución espacial o temporal (evitando transmitir la señal diferencia y reconstruyendo a partir de la predicción exclusivamente)

Codificación diferencial

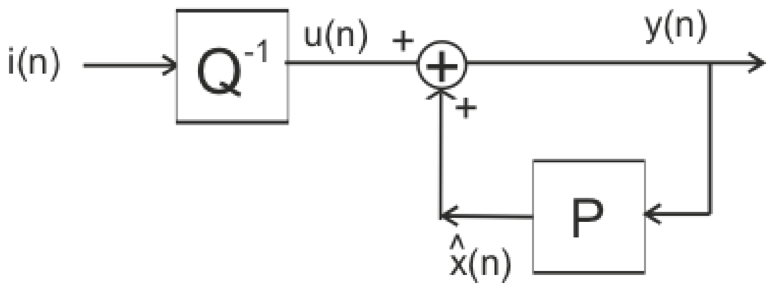
La mejor predicción $\hat{x}(n)$ que puede elaborarse a partir de las muestras anteriores x(n-1), x(n-2), etc. es aquella que minimiza la potencia de la señal diferencia, esto es,

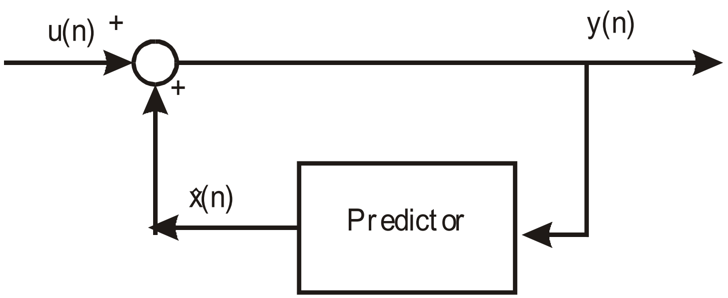
$$\hat{x}(n) = \mathrm{E}\left[X(n) / X(n-1), X(n-2)...\right]$$

Para evitar transmitir información lateral, el estado del decodificador debe depender sólo de las muestras de las diferencias cuantificadas. En consecuencia el estado del codificador también debe depender exclusivamente de las diferencias cuantificadas:

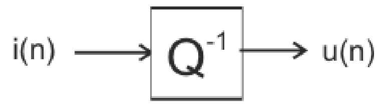
$$\hat{x}(n) = \sum_{j=1}^{N} h_j \tilde{x}(n-j)$$

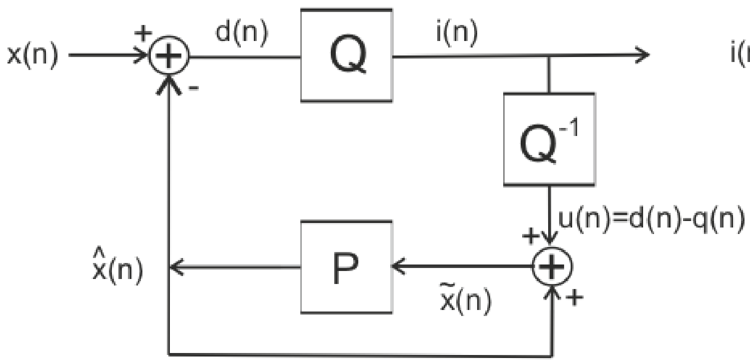
Codificación diferencial: receptor

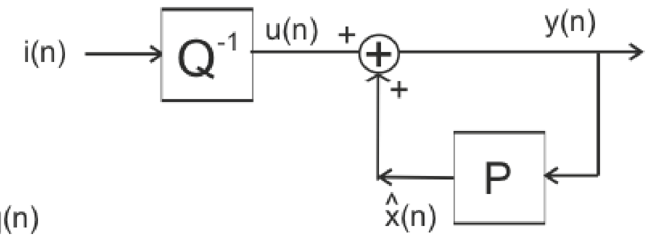


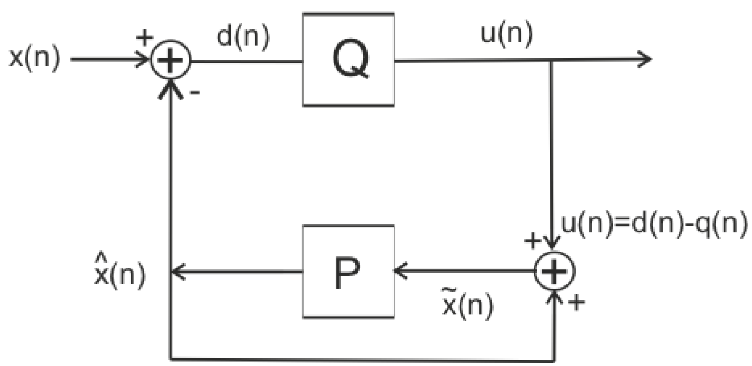


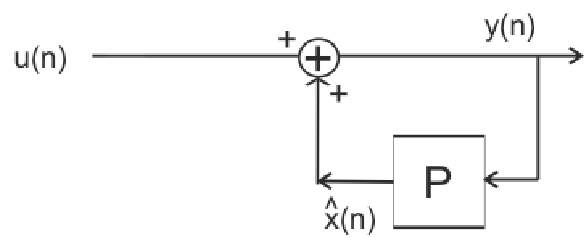
Codificación diferencial. Simplificación para análisis.











Codificación diferencial

Las ecuaciones básicas de un sistema DPCM son:

$$d(n) = x(n) - \hat{x}(n)$$

$$u(n) = d(n) - q(n)$$

$$y(n) = \hat{x}(n) + v(n)$$

donde u(n)=v(n) en ausencia de errores de transmisión.

La señal reconstruida por el receptor es:

$$y(n) = \hat{x}(n) + u(n) = \hat{x}(n) + d(n) - q(n) =$$

$$= \hat{x}(n) + x(n) - \hat{x}(n) - q(n) = x(n) - q(n)$$

de donde el error de reconstrucción es igual al error de cuantificación:

$$r(n) = y(n) - x(n) = q(n)$$

$$\sigma_r^2 = \sigma_q^2 = \epsilon_q^2 \sigma_d^2 2^{-2R}$$

donde ε_q^2 es el factor de rendimiento del cuantificador, que depende de las características estadísticas de la señal a cuantificar (d), del número de niveles (2^R) y del tipo de cuantificador.

Codificación diferencial

Supongamos que cuantificamos la señal original empleando R_{pcm} bits/elemento. El error de reconstrucción coincidirá con el de cuantificación:

$$\sigma_{r,pcm}^2 = \sigma_{q,pcm}^2 = \varepsilon_{pcm}^2 2^{-2R_{pcm}} \sigma_x^2$$

Si utilizamos DPCM, el error de reconstrucción es el de cuantificación de la señal de diferencias:

$$\sigma_{r,dpcm}^2 = \sigma_{q,d}^2 = \varepsilon_d^2 2^{-2R_{dpcm}} \sigma_d^2$$

Si $R_{pcm}=R_{dpcm}$, el cociente entre ambas expresiones representa la ganancia en calidad que se obtiene por el empleo de DPCM:

$$\frac{\sigma_{r,pcm}^2}{\sigma_{r,dpcm}^2} = \frac{\varepsilon_{pcm}^2 \sigma_x^2}{\varepsilon_{dpcm}^2 \sigma_d^2} = \frac{\varepsilon_{pcm}^2}{\varepsilon_{dpcm}^2} G_p$$

donde ε_{pcm}^2 puede aproximarse por la unidad (distribución uniforme) mientras ε_{dpcm}^2 puede aproximarse a 9/2 (dist. Laplaciana).

Codificación diferencial

El número de bits necesarios en DPCM para mantener la misma calidad que en PCM:

$$1 = \frac{\sigma^2_{r,pcm}}{\sigma^2_{r,dpcm}} = \frac{\varepsilon^2_{pcm} 2^{-2R_{pcm}} \sigma^2_x}{\varepsilon^2_{dpcm} 2^{-2R_{dpcm}} \sigma^2_d} = \frac{\varepsilon^2_{pcm}}{\varepsilon^2_{dpcm}} 2^{-2\Delta R} G_p$$

donde $\Delta R = R_{pcm} - R_{dpcm}$. En consecuencia, $\Delta R = \frac{1}{2}\log_2 \frac{\varepsilon^2_{pcm}}{\varepsilon^2_{dpcm}} G_p$, o lo que es lo mismo, por cada incremento de 6dB (multiplicar por 4) en la ganancia de predicción puede ahorrarse un bit por elemento manteniendo la misma calidad.

Diseño de predictores lineales

Para mantener la calidad minimizando el número de bits por elemento es necesario maximizar la ganancia de predicción, o lo que es lo mismo, minimizar la potencia de la señal diferencia.

Teóricamente deberían encontrarse los coeficientes h_i que minimizan el error de predicción:

$$d(n) = x(n) - \hat{x}(n) = x(n) - \left[\sum_{j=1}^{N} h_j y(n-j)\right] = x(n) - \left[\sum_{j=1}^{N} h_j (x(n-j) - q(n-j))\right]$$

$$d(n) = x(n) - \sum_{j=1}^{N} h_j x(n-j) + \sum_{j=1}^{N} h_j q(n-j)$$

esto es, $d(n) = d^*(n) + n(n)$ donde:

$$\boxed{d^*(n) = x(n) - \sum_{j=1}^{N} h_j x(n-j)} \qquad \boxed{n(n) = \sum_{j=1}^{N} h_j q(n-j)}$$

Diseño de predictores lineales

La señal diferencia, d(n) se obtiene a partir del error de predicción $d^*(n)$ elaborado a partir de la señal de entrada x(n) más el error de cuantificación de muestras anteriores filtrado.

El error de cuantificación es filtrado y realimentado al sistema. Si el espectro del ruido de cuantificación es blanco esta operación lo coloreará. El decodificador realizará el filtrado inverso y como resultado se obtendrá de nuevo ruido blanco.

Si R>3 (número de bits por elemento) es mayor que 3 (o si R>1 en sistemas adaptativos), el segundo término puede despreciarse sin introducir grandes perturbaciones, de forma que el objetivo de minimizar la potencia de d(n) puede reemplazarse por el de minimizar la de $d^*(n)$

Diseño de predictores lineales: primer orden

El predictor es $\hat{x}(n) = h_1 x(n-1)$.

La señal diferencia es:

$$d(n) = x(n) - \hat{x}(n) = x(n) - h_1 \cdot x(n-1)$$

$$\sigma_d^2 = E[D^2(n)] =$$

$$= E[X(n)X(n)] - 2h_1 E[X(n)X(n-1)] + h_1^2 E[X(n-1)X(n-1)]$$

$$\sigma_d^2 = \sigma_x^2(1 + h_1^2 - 2\rho_1 h_1)$$

$$\frac{\partial \sigma_d^2}{\partial h_1} = 0 \rightarrow h_1 = \rho_1$$

$$G_p = \frac{\sigma_x^2}{\sigma_d^2} = \frac{1}{(1 + h_1^2 - 2\rho_1 h_1)} = \frac{1}{(1 - \rho_1^2)}$$

Diseño de predictores lineales: primer orden

Conforme aumenta el coeficiente de correlación, aumenta la ganancia de predicción. Para imágenes, donde el valor de este coeficiente se situa facilmente alrededor de 0.96 el valor de la ganancia de predicción puede ser del orden de 11 dB.

Las ecuaciones anteriores pueden escribirse, de forma equivalente:

$$\sigma_d^2 = E\left[(X(n) - h_1 X(n-1))^2\right]$$

$$\frac{\partial \sigma_d^2}{\partial h_1} = -2E\left[(X(n) - h_1 X(n-1))X(n)\right] = E\left[D(n)X(n-1)\right] = 0$$

El error de predicción, d(n), es ortogonal a los datos considerados para realizar la prediccion (Principio de ortogonalidad). En codificación de imágenes, para reducir el volumen de operaciones, se utiliza en ocasiones $h_1=1$.

Diseño de predictores lineales: orden N

$$\hat{x}(n) = \sum_{j=1}^{N} h_j x(n-j)$$

$$\sigma_d^2 = E[D^2(n)] = E\left[\left(X(n) - \sum_{j=1}^{N} h_j X(n-j)\right)^2\right]$$

$$\frac{\partial \sigma_d^2}{\partial h_i} = -2E\left[\left(X(n) - \hat{X}_{opt}(n)\right)\frac{\partial \hat{X}_{opt}(n)}{\partial h_i}\right] = 0$$

y como $\frac{\partial \hat{X}_{opt}(n)}{\partial h_i} = X(n-i)$ queda el sistema de ecuaciones

$$E\left[\left(X(n) - \hat{X}_{opt}(n)\right)X(n-i)\right] = 0 \quad i = 1,2 \ldots N$$

El error de predicción es ortogonal a los datos usados en la predicción (principio de ortogonalidad).

Diseño de predictores lineales: orden N

El sistema anterior puede escribirse:

$$R_{xx}(k) = \sum_{j=1}^{N} h_{j,opt} R_{XX}(k-j) \qquad k = 1,2,\cdots,N$$

$$\mathbf{h}_{opt} = \mathbf{R}_{\mathbf{xx}}^{-1}\mathbf{r}_{\mathbf{xx}}$$

donde $\mathbf{r}_{\mathbf{xx}}^{\mathbf{T}} = \{R_{xx}(i)\}$ y $\mathbf{R}_{\mathbf{xx}} = [R_{xx}|i-j|] \; i,j = 1,2,\cdots,N$

y, por la ortogonalidad antes expuesta

$$E\left[\left(X(n)-\hat{X}_{opt}(n)\right)\hat{X}_{opt}(n)\right]=0 \quad \Rightarrow \quad E\left[X(n)\hat{X}_{opt}(n)\right]=E\left[\hat{X}_{opt}^{2}(n)\right]$$

$$\sigma_d^{*2} = E\left[X^2(n)\right]-2E\left[X(n)\hat{X}_{opt}(n)\right]+E\left[\hat{X}_{opt}^{2}(n)\right]=\sigma_x^2-E\left[X(n)\hat{X}_{opt}(n)\right]$$

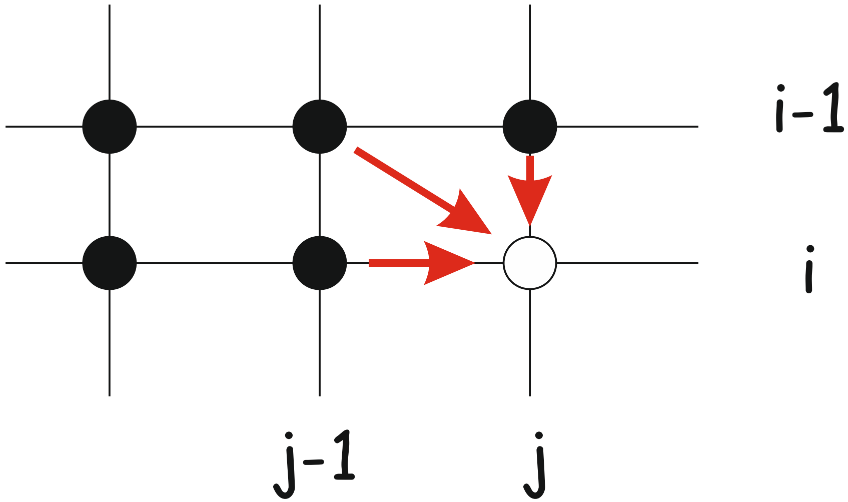
$$\sigma_{d,\min}^2 = \sigma_x^2 - \sum_{i=1}^{N} h_{i,opt} R_{xx}(i) = \sigma_x^2 - \mathbf{h}_{opt}{}^{\mathbf{T}}\mathbf{r}_{\mathbf{xx}} = \sigma_x^2 - \mathbf{r}_{\mathbf{xx}}^{\mathbf{T}}\mathbf{R}_{\mathbf{xx}}^{-1}\mathbf{r}_{\mathbf{xx}}$$

Codificación diferencial para imágenes y vídeo

Diseño de predictores lineales para imágenes

La predicción del elemento x(i,j) puede realizarse usando cualquiera de los anteriores. Candidatos:

x(i,j-1) y x(i-1,j) para orden 2 | x(i,j-1), x(i-1,j) y x(i-1,j-1) para orden 3



Llamando ρ_d al coeficiente de correlación según la diagonal (esto es ρ(1,1)), ρ_v a la correlación según la vertical (ρ(0,1)) y ρ_h según la horizontal (ρ(1,0)) se encuentran fácilmente los siguientes resultados:

Diseño de predictores lineales para imágenes

$$\hat{x}(i,j) = h_h x(i,j-1) + h_v x(i-1,j)$$

$$h_h = \frac{\rho_h - \rho_v\rho_d}{1-\rho_d^2} \qquad h_v = \frac{\rho_v - \rho_h\rho_d}{1-\rho_d^2}$$

Modelo separable:

$$\rho_d = \rho_v\rho_h \quad \Rightarrow h_h = \frac{\rho_h(1-\rho_v^2)}{1-\rho_h^2\rho_v^2} \qquad h_v = \frac{\rho_v(1-\rho_h^2)}{1-\rho_h^2\rho_v^2}$$

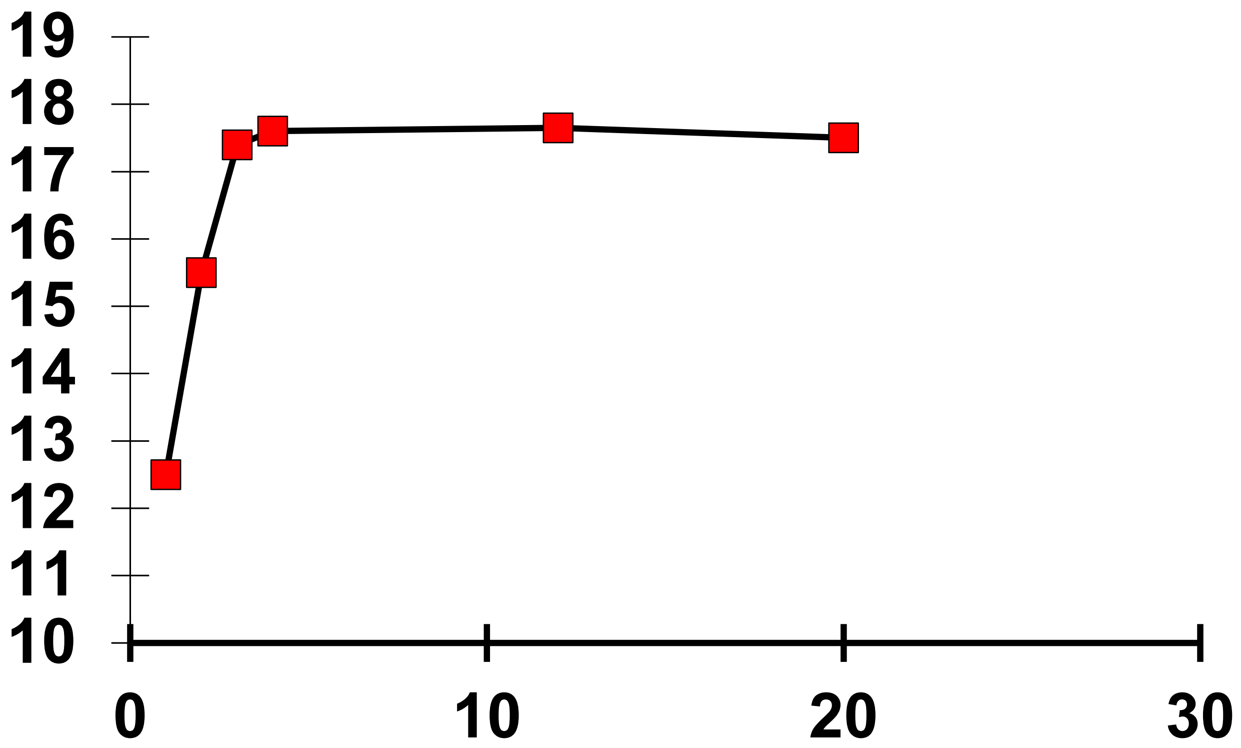
$$\text{Si } \rho = \rho_h = \rho_v \text{ resulta } \; h_h = h_v = \frac{\rho}{1+\rho^2}$$

Modelo isótropico:

$$\rho_v = \rho_h = \rho \qquad \text{y } \rho_d = \rho^{\sqrt{2}}\text{. y por tanto } h_h = h_v = \frac{\rho}{1+\rho^{\sqrt{2}}}$$

Codificación diferencial en imágenes

La ganancia que se obtiene mediante el empleo de sistemas predictivos en la codificación de imágenes se satura rápidamente, de manera que no resultan rentables ordenes de predicción elevados.
Figura: ganancia de predicción (en dB) para distintos ordenes del predictor.

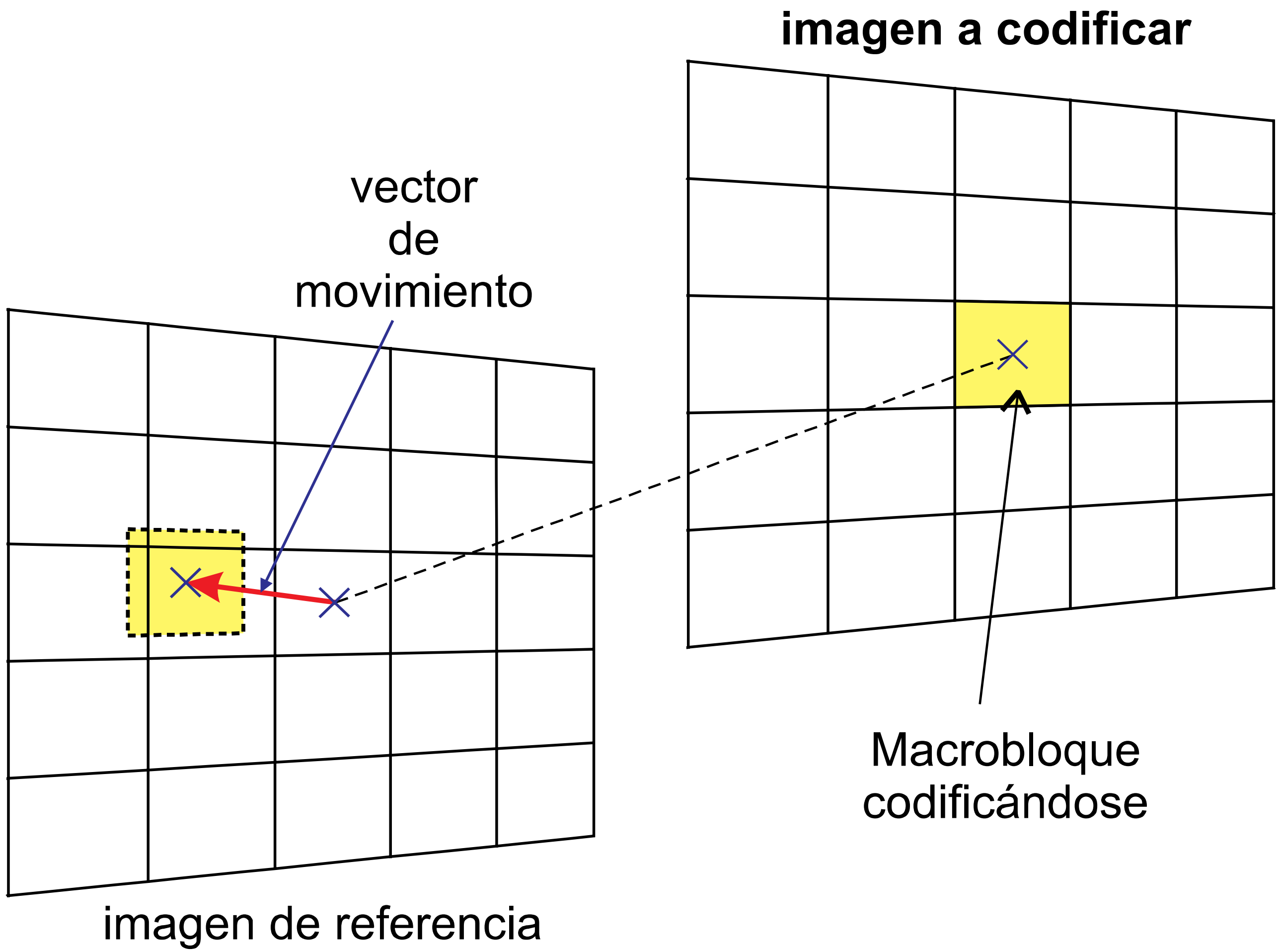


Codificación diferencial en vídeo

- ★ Codificación de la diferencia entre imágenes sucesivas (codificación entre cuadros o “interframe”).
- ★ El único requisito es que las imágenes que se utilizan para la predicción estén disponibles en el receptor.
- ★ Para zonas estáticas, la diferencia será próxima a cero.
- ★ Las zonas donde la iluminación o los objetos sufren cambios o se mueven presentarán diferencias elevadas
- ★ El error de predicción puede reducirse significativamente si el movimiento de los objetos se estima y la predicción se realiza respecto a una imagen con el movimiento compensado.
- ★ La imagen a codificar se divide en bloques cuyo movimiento se estima utilizando diferentes técnicas.
- ★ El receptor no dispone del vector de movimiento, por lo que debe transmitirse como información lateral.
- ★ Por su importancia volveremos sobre este tema más adelante.

Compensación de movimiento

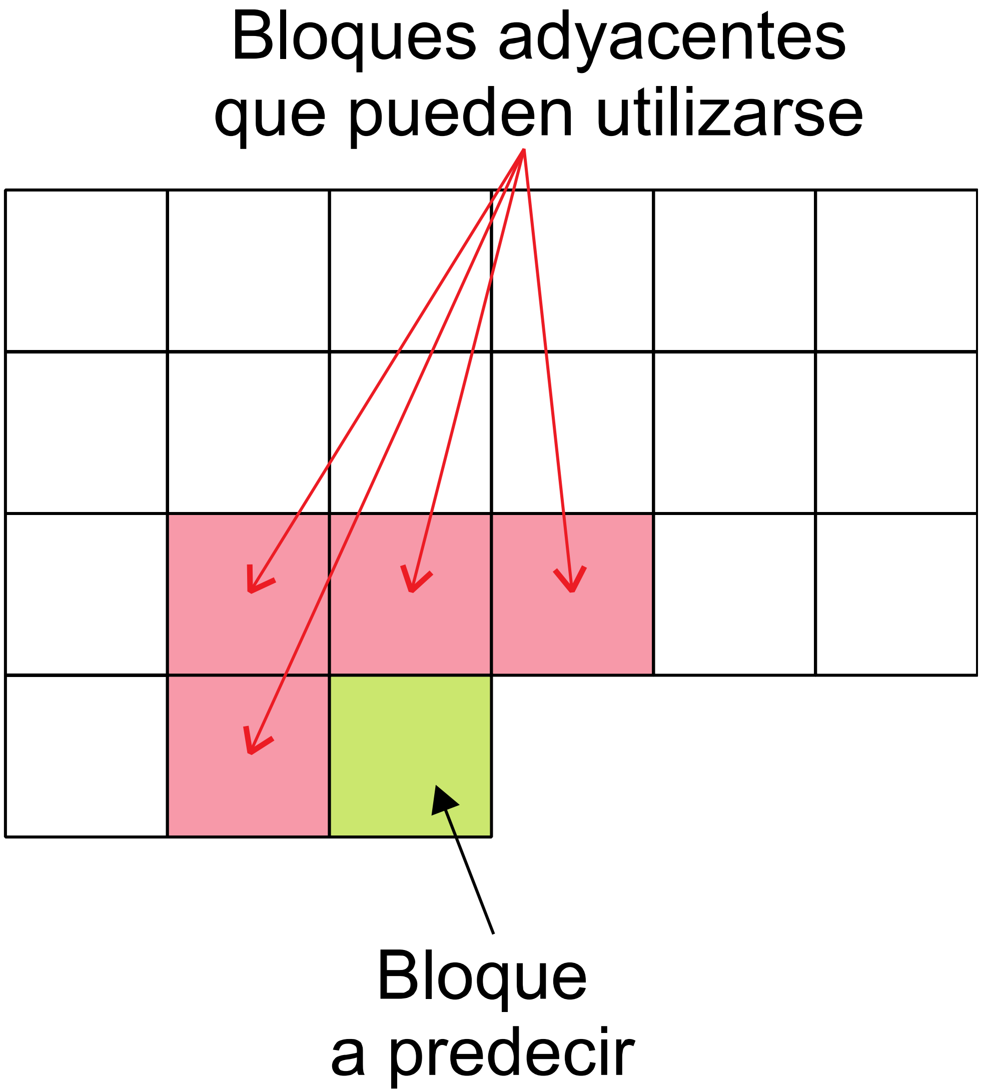
$$e(x,y,t) = i(x,y,t) - i(x - vx, y - vy, t - \Delta t)$$



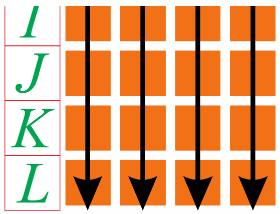
Ejemplos de codificación diferencial

Predicción intra en H.264

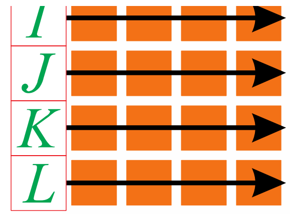
- La imagen está dividida en bloques (detalles mucho más adelante)
- Para la predicción se utilizan bloques que el receptor ha recibido previamente
- Existen multiples opciones. Aquí vamos a ver una de ellas (luminancia 4×4)



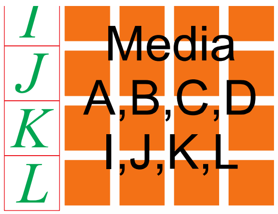
Predicción intra en H.264



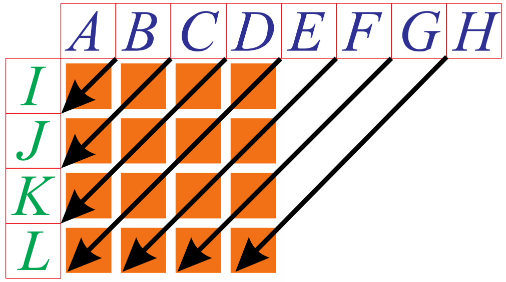
0-Vertical



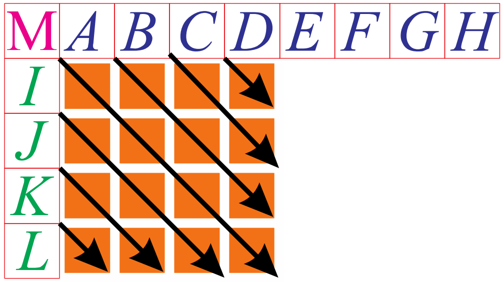
1-Horizontal



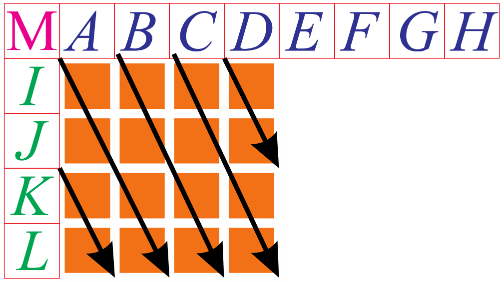
2-DC



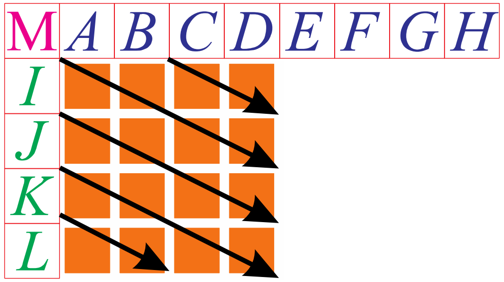
3-Diagonal down left

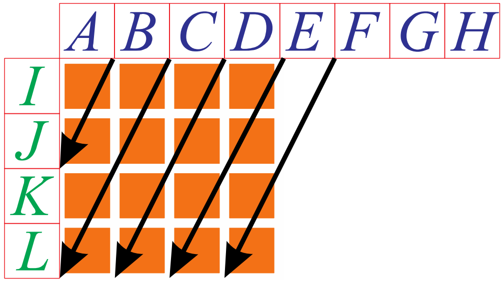


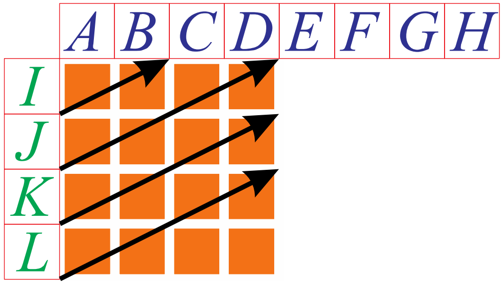
4-Diagonal down right



5-Vertical right







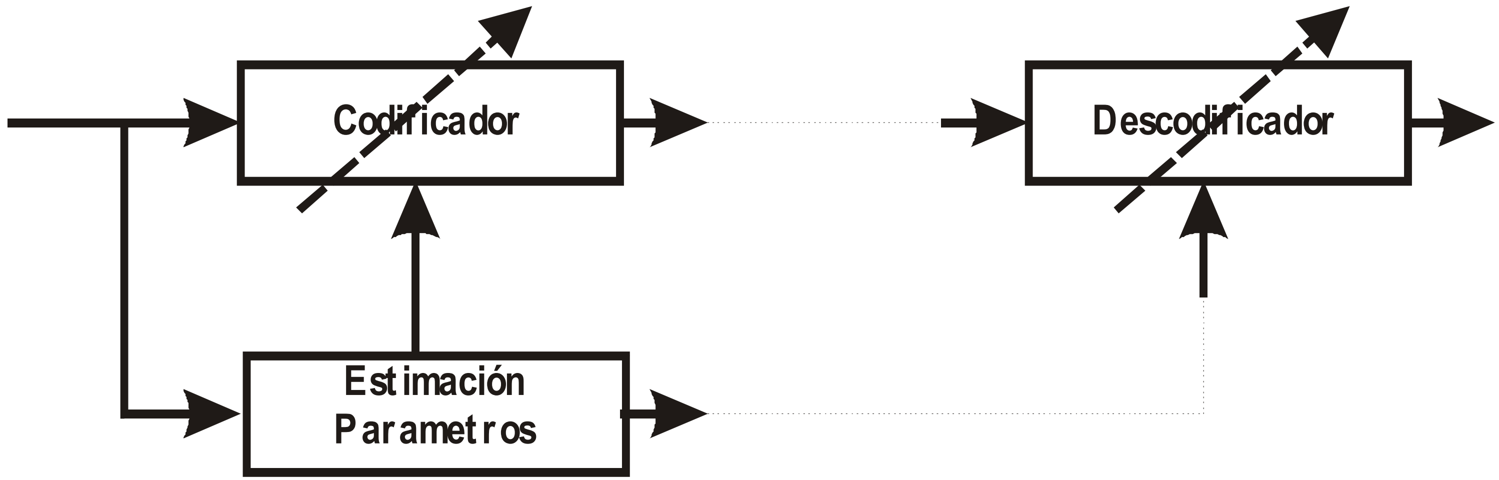
Predicción inter en H.264

- ★ La imagen de referencia se toma de una lista de imágenes ya disponibles en el receptor.
- ★ El vector de movimiento es el desplazamiento entre la región que se está codificando y la zona de la imagen de referencia que se ha seleccionado para realizar la predicción.
- ★ El vector de desplazamiento puede ser un número entero de píxels, o de medios o cuartos de píxel (lo cual implica interpolar la zona de referencia)
- ★ Como los vectores de movimiento se parecen unos a otros, se codifican diferencialmente respecto a los vectores de movimiento de los bloques vecinos ya codificados.
- ★ Aunque la idea general es esta, la predicción Inter H.264 es más compleja… los detalles mucho más adelante.

Adaptación

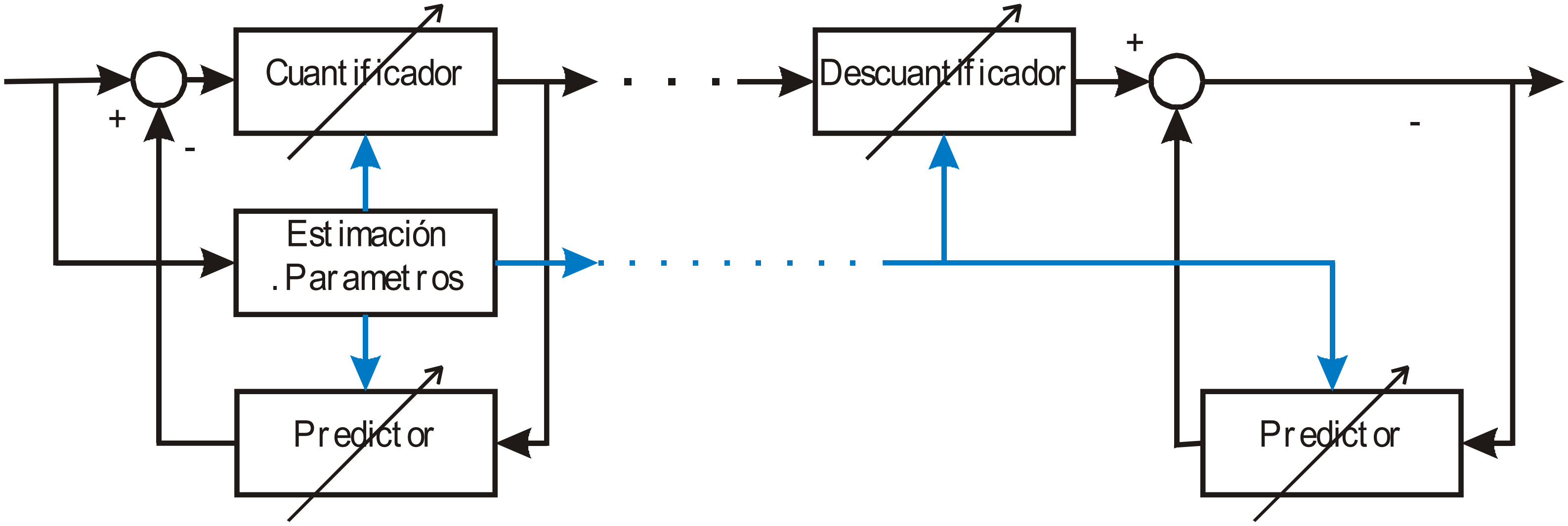
Adaptación hacia adelante

- Un codificador adaptativo extrae cierta información de la señal a codificar y aplica dicha información para adaptarse a la señal a codificar. Como dicha información no está disponible en el decodificador es necesario transmitirla hacia el decodificador.



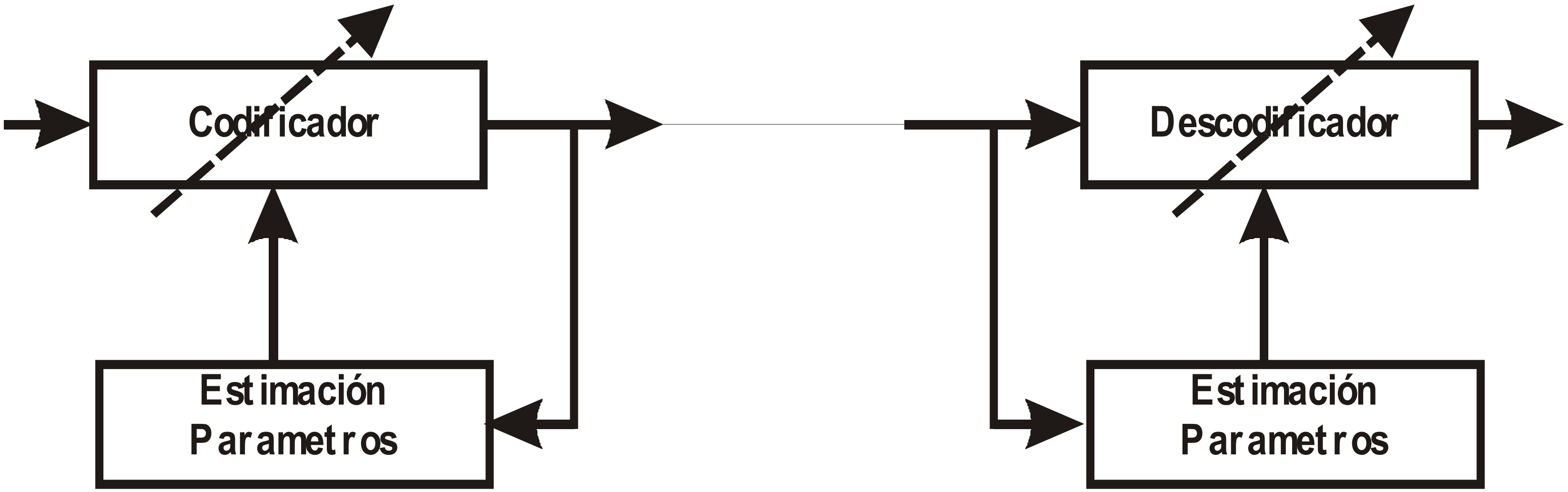
Adaptación hacia adelante

En un sistema DPCM, la adaptación puede realizarse sobre el predictor, sobre el cuantificador o sobre ambos, tal y como se indica en la figura. El modo secuencial sin pérdidas de JPEG es un ejemplo de sistema DPCM adaptativo con adaptación hacia delante. El codificador elige el predictor que va a utilizarse e incluye el valor del selector previamente a la información (diferencial) de la imâgen.



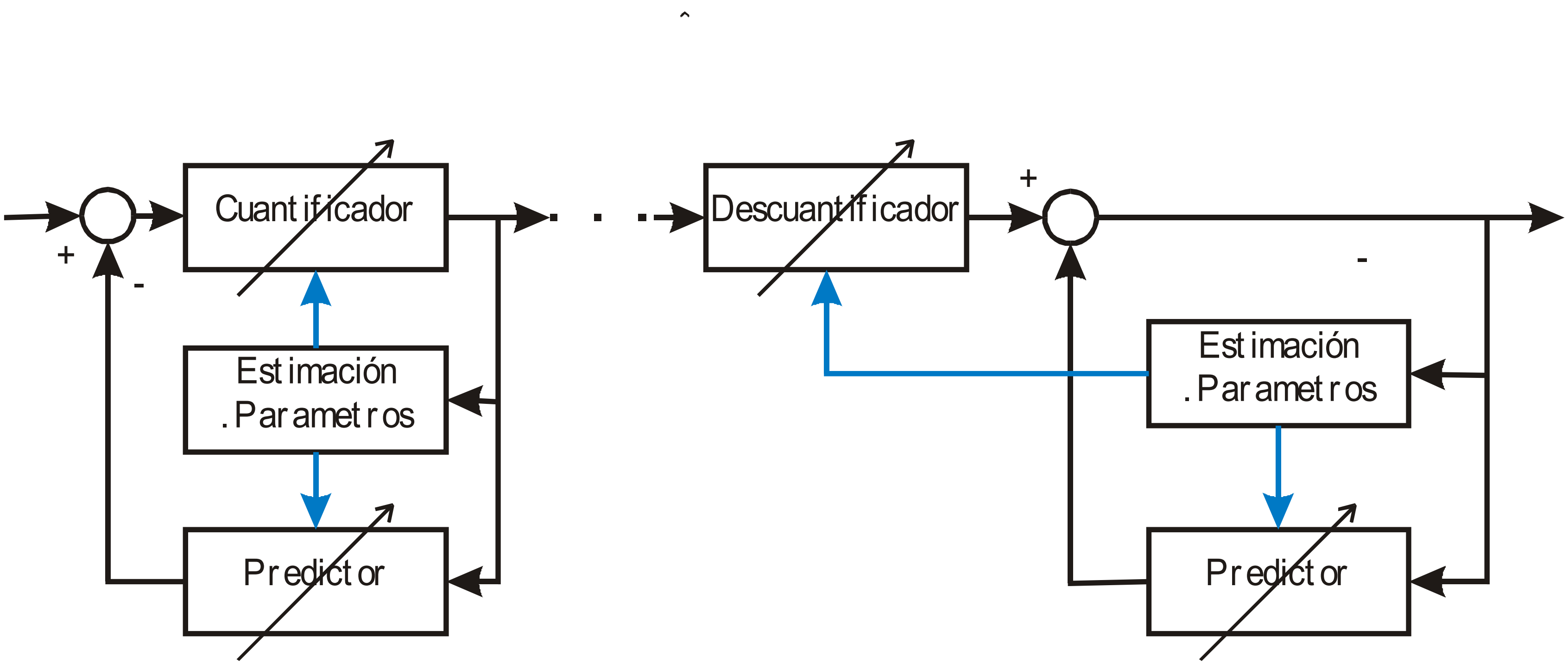
Adaptación hacia atrás

La información que se utiliza para adaptar el codificador se extrae de la propia señal codificada. Como esta información está también disponible para el decodificador, éste puede incluir los mismos algoritmos de estimación de parámetros que el codificador y obtener los mismos resultados, obviándose la transmisión de información lateral.



Adaptación hacia atrás

El ahorro que supone la ausencia de información lateral se paga con calidad de la adaptación y sensibilidad frente a errores en la línea.



Otras técnicas

- ★ Visibilidad de ruido
- ★ Intercambio de resolución espacial y temporal
- ★ Relleno condicional

Bibliografía

- N.S: Jayant y P. Noll, "***Digital Coding of Waveforms: Principles and Applications to Speech and Video***", Prentice Hall Signal Processing Series, New Jersey, 1984.
- S. Khalid, "***Introduction to Data Compression***", 3rd ed., Elsevier, San Francisco, 2006.
- I.E. Richardson, ***"The H.264 Advanced Video Compression Standard"***, 2nd ed., Wiley, Chichester, UK, 2010.