

Tema 5: Procesamiento de vídeo

5.1. Introducción

5.2. Detección y estimación de movimiento

5.3. Filtrado de secuencias de vídeo

5.4. Segmentación

5.5. Análisis de secuencias

5.3.1 Introducción

Podemos tener una secuencia de imágenes de poca calidad debido a :

- **Condiciones de grabación imperfectas o no controlables**
(astronomía, imagen médica, ciencia forense,...) → la mejora y restauración en estos casos es importante por las subsiguientes tareas de análisis e interpretación.
- **Compresión con pérdidas** (JPEG, MPEG, H.26) puede provocar artefactos visibles en las secuencias de imágenes → blocking, ruido de mosquito, etc.

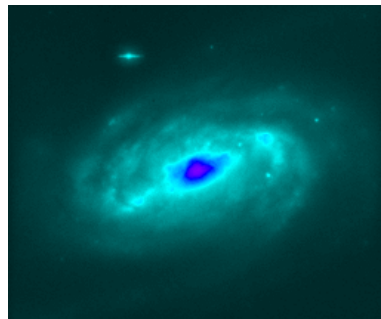


Imagen del telescopio espacial Hubble de la observación terrestre de un campo de estrellas

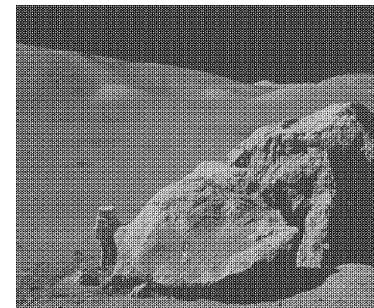


Imagen procedente de una nave tripulada

5.3.1 Introducción

- **Objetivo:** diseñar filtros de reducción de ruido → Filtros de mejora de vídeo
 - compromiso entre cantidad de reducción de ruido y pérdida de calidad perceptual de la imagen.
- Diferencia importante entre la mejora y restauración de imágenes 2D y vídeo
 - cantidad de datos a procesar.
- La mejora y restauración de vídeo debe tener una complejidad manejable y ser semiautomática → un profesional comprueba la calidad de las imágenes restauradas y ajusta parámetros.

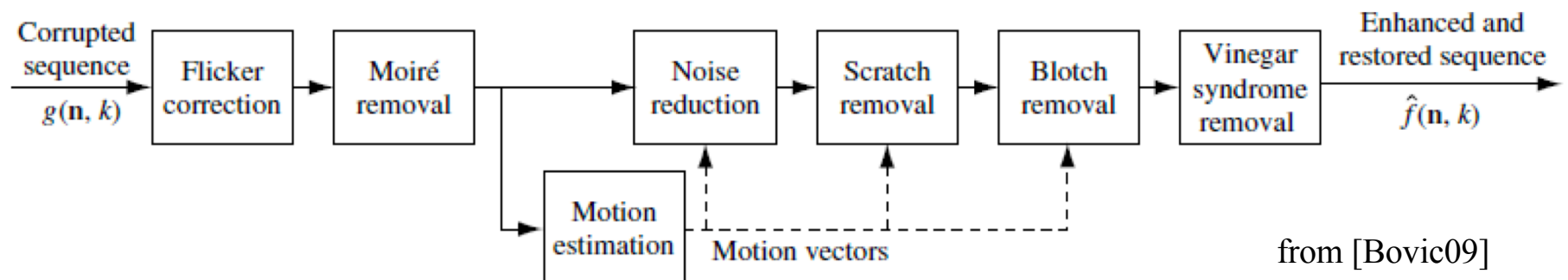
5.3.1 Introducción

Filtros intraframe:

- Tienden a introducir artefactos temporales en las secuencias de imágenes restauradas.
- Explotan las dependencias temporales de las secuencias de imágenes.

Combinación de información → **filtros espaciotemporales.**

- Las secuencias de imágenes pueden ser degradadas por múltiples tipos de artefactos → procedimiento secuencial de eliminación → es posible juzgar el desempeño de cada etapa y ajustar parámetros si es necesario.



from [Bovic09]

5.3.1 Introducción

- La mayoría de las técnicas de filtrado temporal requieren realizar Estimación de Movimiento (**EM**).
- La **EM** de secuencias de imágenes degradadas puede ser problemática.

Solución:

- Reparar los vectores de Movimiento (**VM**) claramente incorrectos.
- Utilizar algoritmos robustos frente a una cantidad limitada de **VM** incorrectos.
- En áreas con **VM** incorrectos imposibles de reparar → filtro de restauración únicamente espacial.

5.3.2 Métodos de reducción de ruido

5.3.2.1. Filtros lineales

5.3.2.2. Filtros de orden estadístico

5.3.2.3. Reducción de artefactos de codificación

Filtrado adaptativos.

5.3.2. Métodos de reducción de ruido

- **Fuentes de ruido** prevalentes: ruido de cámara, ruido originado por el hardware electrónico o por el almacenamiento en cinta magnética, ruido térmico y ruido granular.
- La mayoría de las secuencias de imágenes grabadas contienen una **mezcla de contribuciones de ruido**.
- La secuencia de imágenes grabada $\rightarrow g(n,k) = f(n,k) + w(n,k)$
 - $f(n,k)$: secuencia de imágenes sin corromper.
 - $w(n,k)$: combinación de ruido que se modela como un proceso aditivo blanco (a veces Gaussiano) de media cero y varianza σ_w^2 , independiente de $f(n,k)$.
 - $n=(n1,n2)$, coordenadas espaciales y k , número de **frame**.
- **Objetivo**: estimar $\hat{f}(n,k)$ dado $g(n,k)$.

5.3.2.1 Filtros lineales

A. Filtros de promediado temporal

- Realizan un promediado ponderado de *frames* sucesivos.

$$\hat{f}(\mathbf{n}, k) = \sum_{l=-K}^K h(l)g(\mathbf{n}, k-l) \quad (1)$$

$h(l)$: coeficientes del filtro utilizados para ponderar $2K+1$ *frames*.

→ si todos los frames son igual de importantes $h(l) = 1/(2K+1)$.

→ los coeficientes del filtro pueden obtenerse de forma que se minimice el error cuadrático medio entre la imagen ideal y la restaurada:

$$h(l) \leftarrow \min_{h(l)} E \left[\left(f(\mathbf{n}, k) - \hat{f}(\mathbf{n}, k) \right)^2 \right]$$

→ filtrado temporal de Wiener.

5.3.2.1 Filtros lineales

A. Filtros de promediado temporal. Filtrado temporal de Wiener

$$\begin{pmatrix} h(-K) \\ \vdots \\ h(0) \\ h(1) \\ \vdots \\ h(K) \end{pmatrix} = \begin{pmatrix} R_{gg}(0) & \dots & R_{gg}(-K) & \dots & \dots & R_{gg}(-2K) \\ \vdots & \ddots & & & & \vdots \\ R_{gg}(K) & & R_{gg}(0) & & & \vdots \\ \vdots & & & R_{gg}(0) & & \vdots \\ \vdots & & & & \ddots & \vdots \\ R_{gg}(2K) & \dots & \dots & \dots & \dots & R_{gg}(0) \end{pmatrix}^{-1} \begin{pmatrix} R_{fg}(-K) \\ \vdots \\ R_{fg}(0) \\ R_{fg}(1) \\ \vdots \\ R_{fg}(K) \end{pmatrix}$$

$R_{gg}(m) = E[g(n,k)g(n,k-m)] \rightarrow$ función de autocorrelación temporal.

$R_{fg}(m) = E[f(n,k)g(n,k-m)] \rightarrow$ función de correlación cruzada temporal.

5.3.2.1 Filtros lineales

A. Filtros de promediado temporal

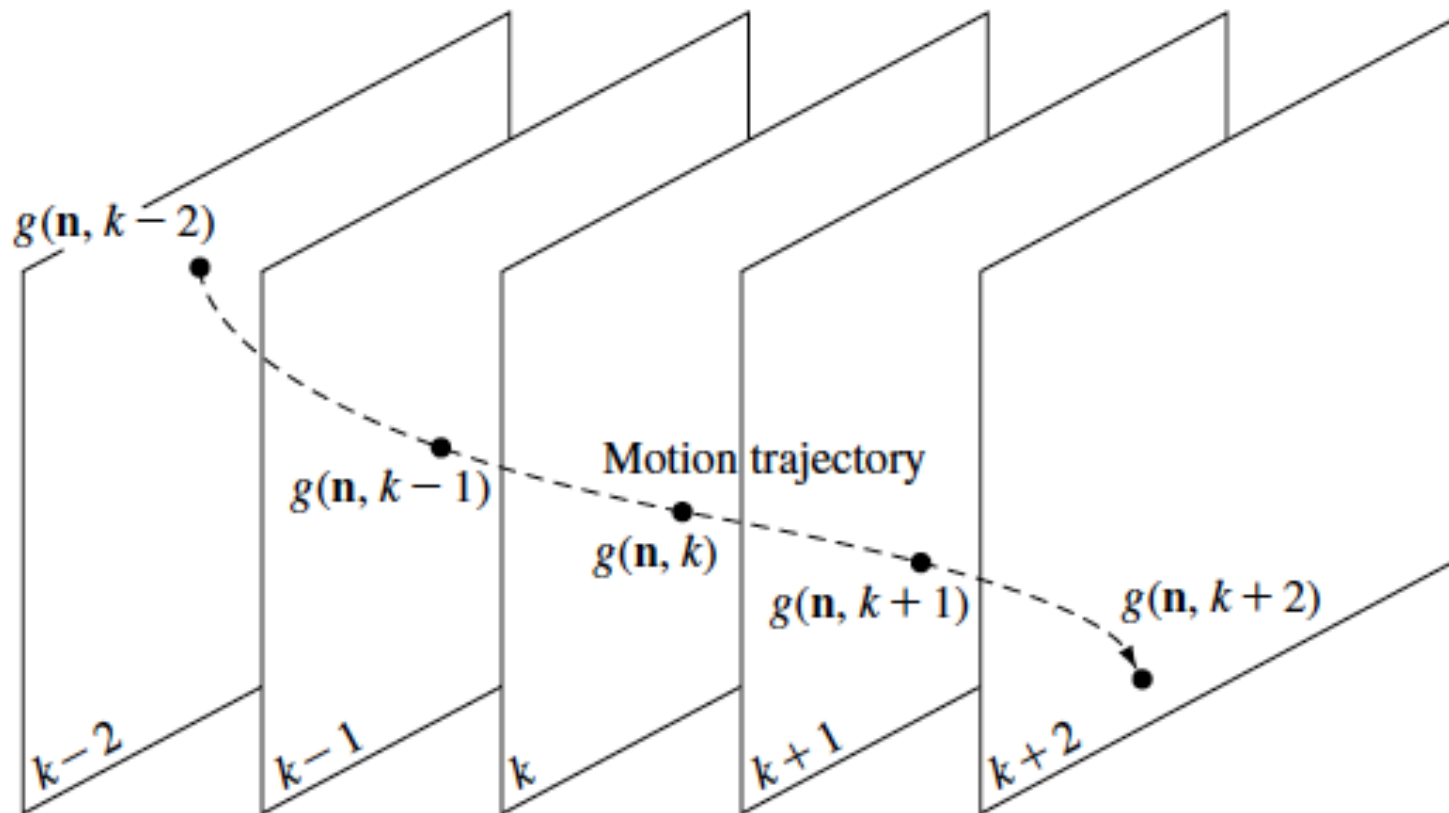
- La longitud de la ventana temporal (K) determina la cantidad de ruido que se puede eliminar.
 - Cuanto mayor sea K → mayor reducción de ruido pero más se notarán los artefactos debidos al movimiento entre frames
→ emborronamiento de los objetos en movimiento debido al promediado entre objeto y fondo.
- Los artefactos de movimiento se pueden reducir aplicando el filtro a lo largo de la trayectoria de movimiento de los objetos → filtro temporal con compensación de movimiento.

$$\hat{f}(n,k) = \sum_{l=-K}^K h(l) g(n_1 - d_x(n_1, n_2; k, l), n_2 - d_y(n_1, n_2; k, l), k - l)$$

$$\mathbf{d}(n; k, l) = (d_x(n_1, n_2; k, l), d_y(n_1, n_2; k, l))$$

5.3.2.1 Filtros lineales

A. Filtro temporal de compensación de movimiento.



from [Bovic09]

5.3.2.1 Filtros lineales

A. Filtros de promediado temporal

- El filtro (1) se puede extender a un filtro espaciotemporal → **Filtro de promediado ponderado 3D.** (2)

$$\hat{f}(\mathbf{n}, k) = \sum_{(\mathbf{m}, l) \in S} h(\mathbf{m}, l) g(\mathbf{n} - \mathbf{m}, k - l)$$

S: ventana espaciotemporal del filtro.

- Los coeficientes $h(\mathbf{m}, l)$ pueden ser seleccionados iguales pero el filtro mejora si se adaptan a la secuencia de imágenes a ser filtrada, p.e. optimizándolos en términos del error cuadrático medio → **Filtro de Wiener 3D.**

5.3.2.1 Filtros lineales

A. Filtros de promediado temporal. Filtro de promediado ponderado 3D

- Desventajas filtro de Wiener 3D:
 - Se precisa conocimiento a priori de la función de autocorrelación 3D de la secuencia de ideal de imágenes.
 - Asume estacionariedad en 3D en sentido amplio, lo cual no se cumple debido al movimiento de objetos y cambios de escena.
- Se suelen utilizar formas más simples para seleccionar los coeficientes del filtro 3D de forma adaptativa.

Tema 5.3.2.1 Filtros lineales

A. Filtros de promediado temporal. Filtro de promediado ponderado 3D

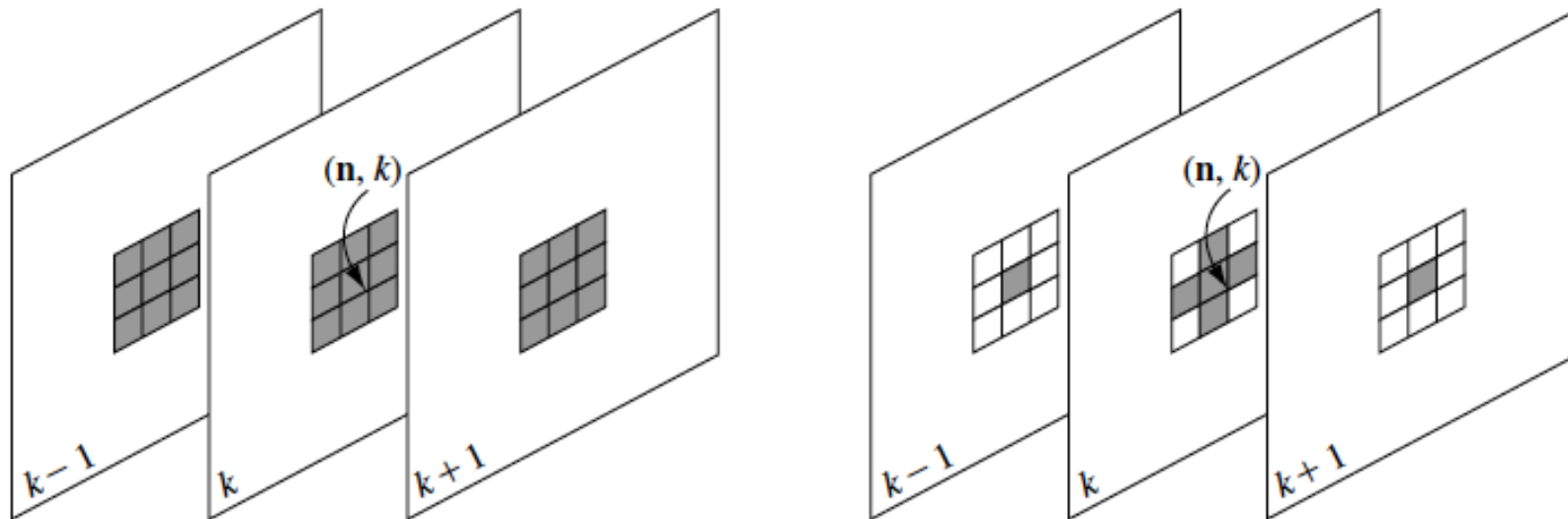
$$h(\mathbf{m}, l; \mathbf{n}, k) = \frac{c}{1 + \max \left(\alpha, (g(\mathbf{n}, k) - g(\mathbf{n} - \mathbf{m}, k - l))^2 \right)}$$

$h(\mathbf{m}, l; \mathbf{n}, k)$: pondera la intensidad en la localización espacial $\mathbf{n} - \mathbf{m}$ en el frame $k - l$ para la estimación de la intensidad $\hat{f}(\mathbf{n}, k)$

- Los píxeles cuya intensidad dista mucho de $g(\mathbf{n}, k)$ (debido al movimiento de dentro de la ventana 3D S) serán ponderados con un peso muy bajo.
- Igual que con el filtro temporal (1), el filtro espaciotemporal (2) puede implementarse con compensación de movimiento.

Tema 5.3.2.1 Filtros lineales

A. Filtros de promediado temporal. Filtro de promediado ponderado 3D



from [Bovic09]

5.3.2.1 Filtros lineales

B. Filtros temporales recursivos

- **Desventaja filtros temporales:** necesitan almacenar en memoria temporal varios frames.
- Los filtros recursivos utilizan menos memoria temporal, generalmente sólo almacenan un frame.

$$\hat{f}(\mathbf{n}, k) = \hat{f}_b(\mathbf{n}, k) + \alpha(\mathbf{n}, k) \left[g(\mathbf{n}, k) - \hat{f}_b(\mathbf{n}, k) \right]$$

$\hat{f}_b(\mathbf{n}, k)$: predicción del frame k en base a los frames previamente filtrados.

$\alpha(\mathbf{n}, k)$: ganancia del filtro para actualizar la predicción

5.3.2.1 Filtros lineales

Filtros temporales recursivos

- Una elección común es: $\hat{f}_b(\mathbf{n}, k) = \hat{f}(\mathbf{n}, k - 1)$
- O su forma con compensación de movimiento:

$$\hat{f}_b(\mathbf{n}, k) = \hat{f}(\mathbf{n} - \mathbf{d}(\mathbf{n}; k, k - 1), k - 1)$$

- $\alpha(\mathbf{n}, k)$ puede ser fija o adaptativa para evitar artefactos debidos al filtrado.

$$\alpha(\mathbf{n}, k) = \begin{cases} 1 & \text{if } |g(\mathbf{n}, k) - \hat{f}_b(\mathbf{n}, k)| > \varepsilon \\ \alpha & \text{if } |g(\mathbf{n}, k) - \hat{f}_b(\mathbf{n}, k)| \leq \varepsilon \end{cases}$$

5.3.2.2 Filtros OS

- **Filtros de orden estadístico (OS)** → variante no lineal de los filtros de promediado ponderado
 - En los filtros OS los datos observados dentro de una ventana espaciotemporal son ordenados antes de ser utilizados.
 - Utilizan información de magnitud en lugar de correlación temporal.
 - Se utilizan habitualmente en filtrado direccional
- se consideran diferentes direcciones de filtrado correspondiendo con diferentes orientaciones de los ejes espaciotemporales.

5.3.2.2 Filtros OS

- **Estructura** general de un **filtro OS**:

$$\hat{f}(\mathbf{n}, k) = \sum_{r=1}^{|S|} h_{(r)}(\mathbf{n}, k) g_{(r)}(\mathbf{n}, k)$$

$g_{(r)}(\mathbf{n}, k)$: intensidades ordenadas de la secuencia de imágenes ruidosas.

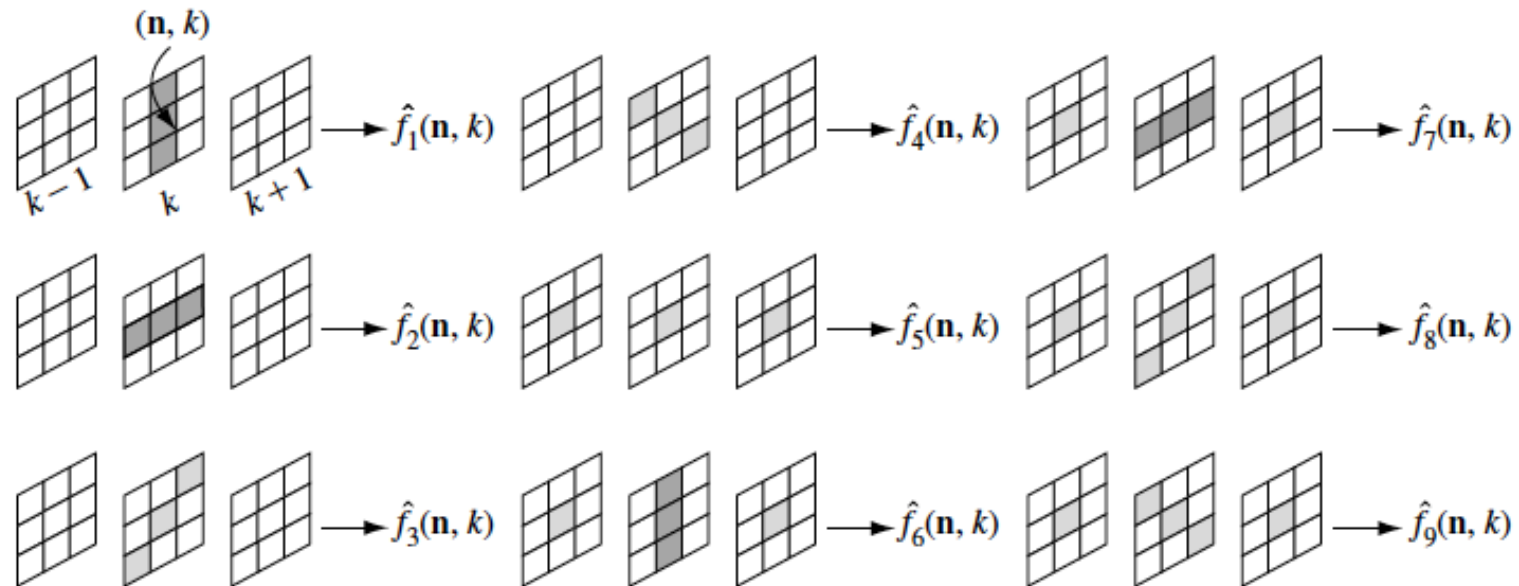
$|S|$: número de intensidades en la ventana espaciotemporal S.

- El filtro OS más simple:

$$\hat{f}(\mathbf{n}, k) = \text{median}(g(\mathbf{n}, k - 1), g(\mathbf{n}, k), g(\mathbf{n}, k + 1))$$

5.3.2.2 Filtros OS

- **Filtro OS de mediana multitapa (MMF):** se combinan las salidas de filtros de mediana simples sobre diferentes regiones espaciotemporales.



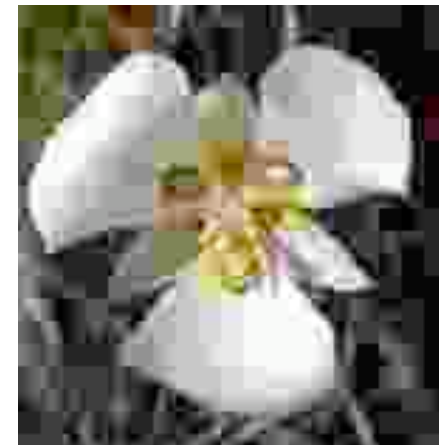
$$\hat{f}(n, k) = \text{median} \left(g(n, k), \max \left(\hat{f}_1(n, k), \dots, \hat{f}_9(n, k) \right), \min \left(\hat{f}_1(n, k), \dots, \hat{f}_9(n, k) \right) \right)$$

5.3.2.2 Filtros OS

- **Ventaja** (de los MMF): aunque no se realiza estimación de movimiento, se reducen considerablemente los artefactos en los ejes de los objetos en movimiento.
- Una ventaja de ordenar las observaciones ruidosas antes de utilizarlas es que se pueden detectar fácilmente valores significativamente diferentes de $g(\mathbf{n},k)$, estos valores corresponden generalmente a objetos en movimiento, se pueden evitar esos valores a la hora de filtrar.

5.3.3 Reducción de artefactos de codificación

- La compresión con pérdidas puede provocar artefactos visibles en las secuencias de imágenes.
- La transformada discreta del coseno (DCT) es una de las técnicas más utilizadas en compresión de imagen y vídeo (JPEG, H.26, MPEG) → cada bloque es codificado de forma separada → provoca *blocking* cuando se utiliza con una tasa de bit baja para conseguir un buen ratio de compresión.
- También puede introducir emborronamiento o ruido de mosquito como consecuencia de eliminar las altas frecuencias cuando se truncan los coeficientes de la DCT.



5.3.3 Reducción de artefactos de codificación

Reducción de artefactos en el dominio espacial

- Debido a los bordes horizontales y verticales de los bloques, aparecen altas frecuencias adicionales en el espectro de la señal de vídeo decodificada.
- La solución más simple sería filtrar paso bajo → podemos perder altas frecuencias propias de las imágenes.
- **Filtrado adaptativo:**
 - Etapa de clasificación (determinar si un bloque contiene bordes) + etapa de filtrado.
 - Si el bloque no contiene bordes → filtrado 2D.
 - Si el bloque contiene bordes → filtrado direccional 1D.

5.3.3 Reducción de artefactos de codificación

Reducción de artefactos en el dominio frecuencial

- Es necesario tener acceso a los coeficientes de la DCT o recalcularlos.
- Se puede explotar la naturaleza periódica de los artefactos de bloque debida a la estructura de rejilla → los armónicos generados por los patrones regulares pueden ser medidos fácilmente en el dominio frecuencial → la amplitud de los armónicos es proporcional a la magnitud de los artefactos de bloque mientras que la fase puede ser utilizada para verificar que los armónicos no son debidos a información propia de la imagen.
- Se pueden eliminar estas componentes frecuenciales igualando a cero los correspondientes coeficientes de la DCT y obteniendo después la DCT inversa.