REDUCCIÓN DEL TIEMPO DE CÓMPUTO DE LA ETAPA DE ESTIMACIÓN DE MOVIMIENTO EN ALGORITMOS DE CODIFICACIÓN DE VÍDEO

La estimación de movimiento entre frames es una de las principales técnicas en los codificadores de vídeo que sirve para descubrir la gran cantidad de redundancia temporal en fotogramas consecutivos. Desvelando esa redundancia, la codificación puede ser más eficiente ya que sólo debe representarse la información nueva.

La idea es muy sencilla. Dados dos frames, uno al que llamaremos "actual" (current) y otro que llamaremos "referencia", el objetivo es buscar la mejor correspondencia del bloque seleccionado en el frame "actual" con uno de los bloques del frame "referencia", dentro de un área de búsqueda. Se puede ver la idea general representada en la siguiente figura:

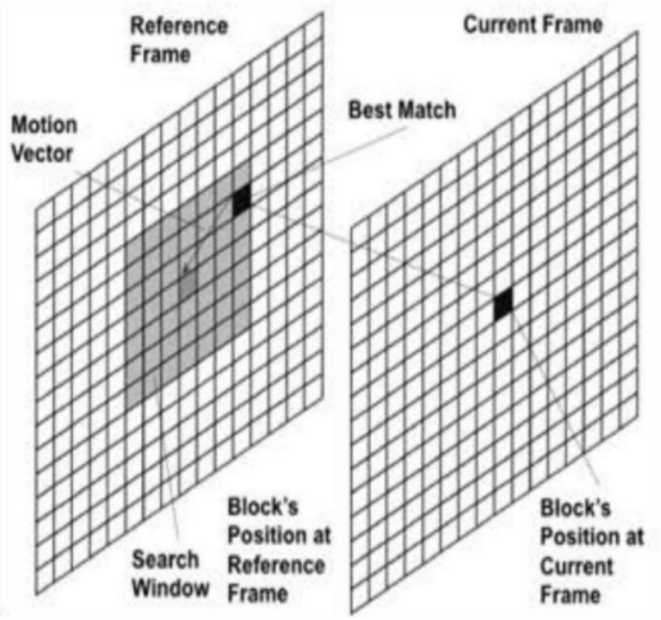

Ilustración 1. Estimación del vector de movimiento para un macrobloque. Figura extraída de Literature survey on motion estimation techniques - Scientific Figure on ResearchGate.

Disponible en: https://www.researchgate.net/figure/Motion-Estimation-Technique-and-Motion-Vector-Determination-Paper-6-PMuralidhar-and_fig1_332665573 [último acceso, 26 marzo, 2021]

Como podréis intuir, se trata de una operación computacionalmente exigente ya que cada bloque del frame "actual" debe compararse con todos los posibles candidatos en el frame "referencia" dentro de su área de búsqueda.

¿Y cuál es el mejor candidato? Es aquel que tenga el mayor nivel de similitud con el bloque "actual". Existen muchas funciones de coste que calculan este grado de similitud. Como podéis leer en la Wikipedia (sección *Evaluation Metrics*), las métricas más habituales son MAD y MSE, aunque también existen otras como SAD (*Sum of Absolute Differences*).

Al ser ésta una etapa muy exigente computacionalmente, se han propuesto diferentes métodos que alivien de alguna manera la carga computacional. La estrategia es explorar el espacio de búsqueda de una manera inteligente, solamente comparando el bloque "actual" con algunos de los bloques de "referencia" dentro del área de búsqueda (diamante, búsquedas en 3 o 4 pasos, árbol, etc.). En nuestro caso, ya que queremos llevar al límite nuestras capacidades y ver la mejora que podemos obtener, vamos a centrarnos en el método de ***búsqueda exhaustiva***. Es decir, comparamos el bloque "actual" con **TODOS LOS POSIBLES BLOQUES CANDIDATOS EN EL FRAME DE REFERENCIA**.

El resultado de todo este proceso es, por cada bloque del frame, un vector (es decir, una pareja de enteros X e Y) que indica el desplazamiento (o posición relativa) del bloque con la mejor función de coste dentro del área de búsqueda.

Evidentemente, este proceso está altamente parametrizado y, de hecho, durante el proceso de compresión de vídeo se ejecuta varias veces con variaciones en los mismos (diferentes niveles). Pero, nosotros vamos a simplificar y elegir un caso concreto:

- **Resolución de vídeo**. Trabajaremos con calidad HD.
- **Tamaño de bloque.** El frame se divide en grupos de pixeles que se procesan como un grupo o bloque. En nuestro caso de estudio lo establecemos en **N=16**. Es decir, la imagen es procesada en bloques de 16x16. Así, si la resolución de la secuencia de vídeo es HD, el número de bloques en un frame es: (1280/16) x (720/16) = 80 x 45 = 3600 bloques.
- **Área de búsqueda**. Se define como el número de pixeles en todas las direcciones en la que extendemos la búsqueda de nuestro mejor candidato. En este caso estableceremos **p = 24**. Así, en el frame "actual" una matriz de (16+24+24) x (16+24+24) = 64x64 pixeles es el área de búsqueda.
- **Función de coste**. Utilizaremos **MSE** (*Mean Squared Error*).

Con estos datos, debéis tener en cuenta lo siguiente:

- La comparación de dos bloques cualesquiera comprende la realización de 16x16 restas, 16x16 multiplicaciones, 16x16 sumas y una división (aunque el compilador la cambiará por un desplazamiento aritmético a la derecha).
- La búsqueda de un bloque en su área de búsqueda comprende 48x48 comparaciones o cálculos de la función de coste. Por lo tanto, tenemos un total de $48x48x16x16 \cong 590.000$ x 3 (restas, multiplicaciones y sumas), 48x48 divisiones (desplazamientos) y 48x48 comparaciones (para determinar el mínimo).
- Finalmente se elige aquel bloque dentro del área de búsqueda con el menor coste.
- El resultado será una matriz de 80 x 45 vectores o estimaciones de movimiento (una por cada bloque de 16x16 píxeles en el frame "actual" (HD).

Además, para completar el análisis, hay que darse cuenta de que las áreas de búsqueda en el frame "referencia", para dos bloques en el frame "actual" pueden solapar. Eso puede tener su impacto en la estrategia elegida para distribuir los datos, etc.

Por ejemplo, en la siguiente imagen se puede observar cómo los píxeles de la franja central en el frame "referencia" son tenidos en cuenta para la búsqueda del mejor candidato de los bloques #1 y #2. Es decir, un bloque del frame "referencia" que caiga en esa zona deberá compararse tanto con el bloque amarillo como con el bloque azul. Este escenario debe generalizarse cuando aumenta el número de bloques y las dimensiones a tener en cuenta (ver Ilustración 3).

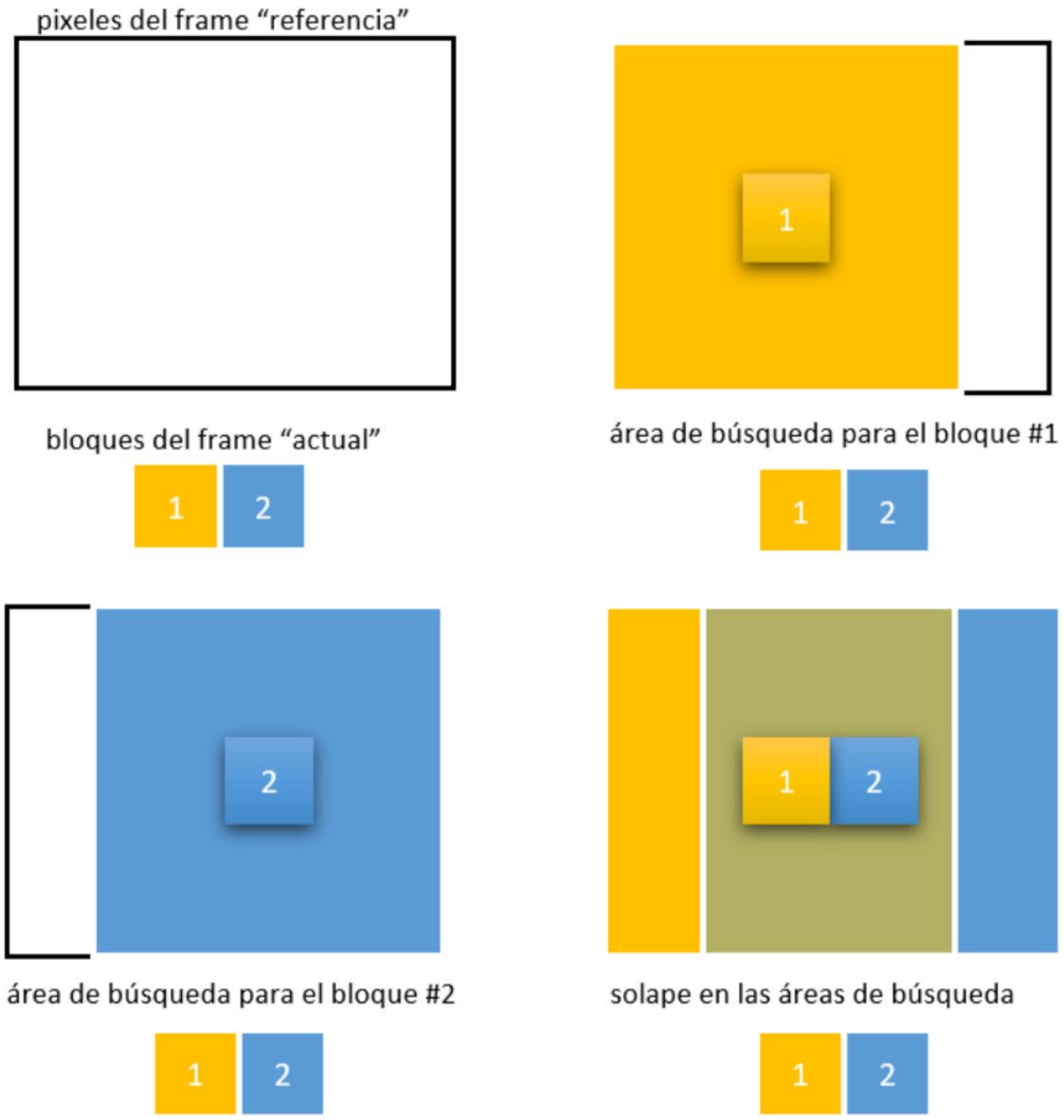

Ilustración 2. Los píxeles de la imagen de referencia pueden participar en los cálculos para más de un vector de movimiento asociado a un bloque.

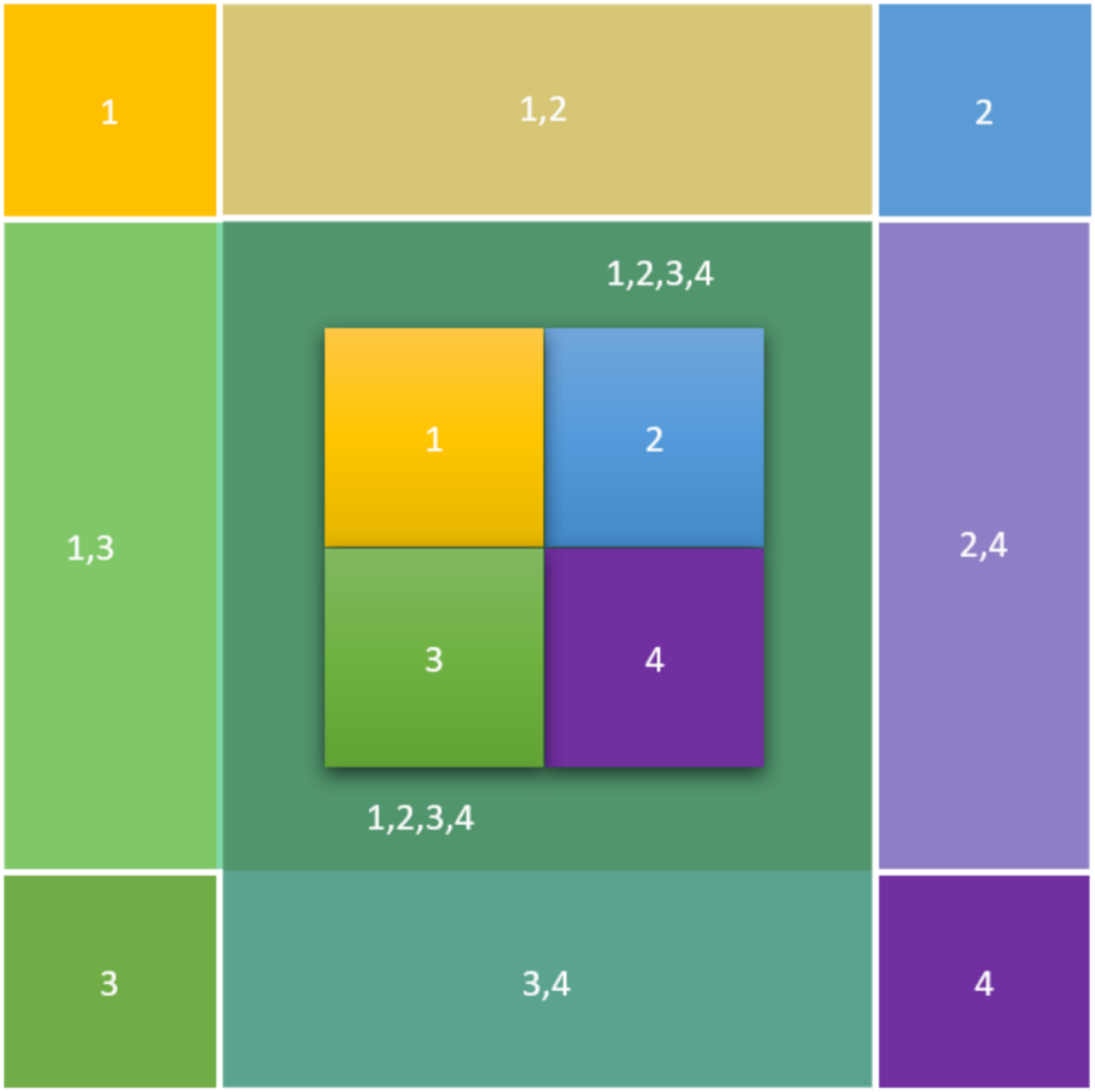

Ilustración 3. Contribuciones de los píxeles del frame "referencia" al cálculo de los vectores de movimiento para cuatro bloques del frame "actual"