

NUEVO ALGORITMO DE COMPRESIÓN MULTIMEDIA: CODIFICACIÓN POR SALTOS LOGARÍTMICOS

Autor: Jose Javier García Aranda

Director: Francisco González Vidal



Departamento de Ingeniería de Sistemas telemáticos

1

Agenda

NUEVO ALGORITMO DE COMPRESIÓN MULTIMEDIA:
CODIFICACIÓN POR SALTOS LOGARÍTMICOS



POLITÉCNICA

0 | Introducción

- 1| Estado del arte
- 2| LHE básico
- 3| LHE avanzado
- 4| Decodificador LHE
- 5| Video LHE
- 6| Resultados
- 7| Conclusiones y trabajo futuro

2

Primeramente haremos una introducción para explicar la motivación para esta tesis y cuál es el objetivo último que persigo con esta nueva tecnología.

A continuación cubriré el Estado del arte en el que contemplaremos los algoritmos y formatos con una perspectiva diferente a la habitual para poner el contexto en nuevo algoritmo.

LHE básico sus fundamentos fisiológicos y el diseño de un compresor básico, aunque con limitaciones de bitrate.

LHE avanzado donde presentaremos el algoritmo completo y sin limitaciones.

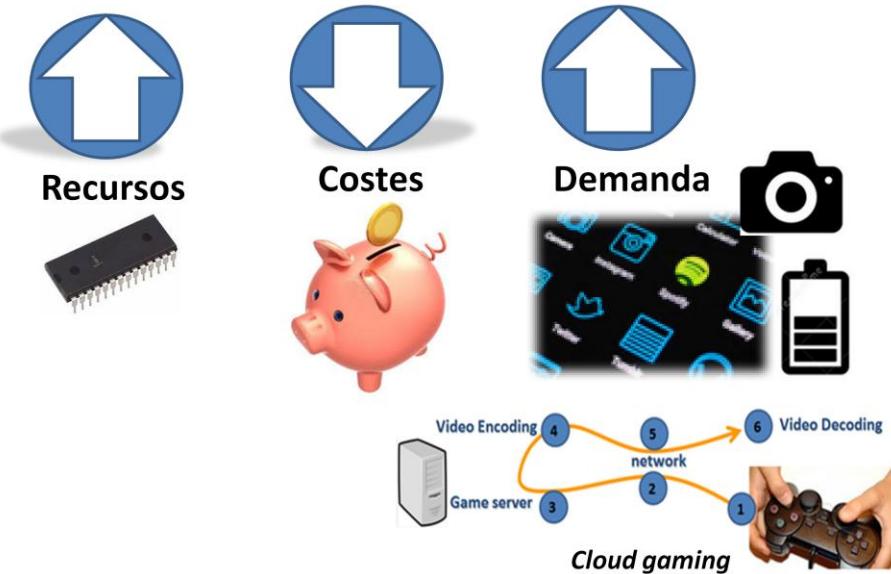
Decodificador relacionado con el LHE avanzado.

Video, vamos a ver resueltos los primeros pasos para construir un codificador de video basado en LHE.

Resultados de imagen y video.

Conclusiones y trabajo futuro donde además de publicaciones, nuevas ideas y proyectos acabaremos con un resumen de las contribuciones tecnológicas realizadas a lo largo de esta tesis.

0 | Introducción



3

Cabe preguntarse si aun hoy en dia con lo baratos que son los recursos computacionales (de ram y cpu) merece la pena el progreso en los algoritmos de compresion

A mayor disponibilidad de recursos, menor coste, pero igualmente el ser humano tiene una voracidad insaciable consumiendo informacion y este menor coste genera una mayor demanda de aplicaciones que consumen mas y mas informacion, creando una necesidad nunca satisfecha del todo. Tenemos un ejemplo muy ilustrativo con las camaras digitales de fotos. No parece que nunca tengan suficientes megapixels.

Hoy en dia tenemos servicios experimentales como el cloud gaming o el video HD online en el que la necesidad de consumo de recursos, ancho de banda y capacidad de cómputo son cada vez mayores, de modo que una reducción de los recursos computacionales siempre va a representar un progreso en la experiencia de usuario al mismo coste. De hecho un ahorro de milisegundos puede ser vital para lograr una experiencia de juego efectiva

Pero ademas, un menor coste computacional puede lograr ahorros significativos en el consumo energetico de los terminales moviles y alargar el uso de las baterias.

Por todo ello no solo merece la pena progresar en algoritmos de compresion sino que es una necesidad real, tanto comprimir mas como hacerlo de un modo mas eficiente reduciendo la complejidad computacional

0 | Introducción: motivación

Problema no resuelto : latencia de codificación en “cloud gaming” limita el despegue comercial de estos servicios.

Retos:

- El “time-memory” trade-off.
- La mejora de la calidad.
- El problema de las patentes que afecta a la difusión de los algoritmos.



LHE: nuevo algoritmo de baja complejidad computacional (lineal) y sin patentes , con calidad superior a JPEG.

4

Actualmente latencia de codificación en cloud gaming es demasiado elevada y esto representa un problema para el despegue comercial de estos servicios

Los Retos que representa este problema son 3:

- El “time-memory” trade-off en servicios de estricta latencia de codificación como es el caso del “cloud-gaming”. El time memory trade off es el equilibrio entre el tiempo que emplea un algoritmo en comprimir y el consumo de memoria que tiene. Un algoritmo puede consumir menos memoria a costa de ejecutarse mas lentamente y puede consumir menos tiempo a costa de consumir mas memoria
 - La mejora de la calidad (porque hay que acercarla lo más posible a la compresión sin pérdidas).
 - El problema de las patentes que afecta a la difusión de los algoritmos (como ocurre con JPEG2000 o con la compresión fractal).
- Por todo ello he optado por un algoritmo libre sin patentes y que rompe con los algoritmos estandares de compresion en el dominio de la frecuencia para ahorrar recursos computacionales, ganar asi velocidad y tratar al mismo tiempo de lograr una calidad superior.

0 | Introducción: objetivos técnicos concretos

Creación de un algoritmo básico de compresión (sin creación ni uso de patentes) de complejidad lineal y fundamentado en el comportamiento del ojo humano

Creación de una métrica de relevancia perceptual que permita medir la relevancia que tiene para el observador cualquier área de la imagen

Creación de los algoritmos de **downsampling elástico e interpolación elástica**

Creación de un **algoritmo completo** que usando el sub-muestreo elástico y el modelado del ojo humano permita comprimir linealmente y con mayor calidad que JPEG.

Validación del algoritmo con bases de datos de imágenes de referencia usando diferentes métricas estándares de calidad (PSNR y SSIM).

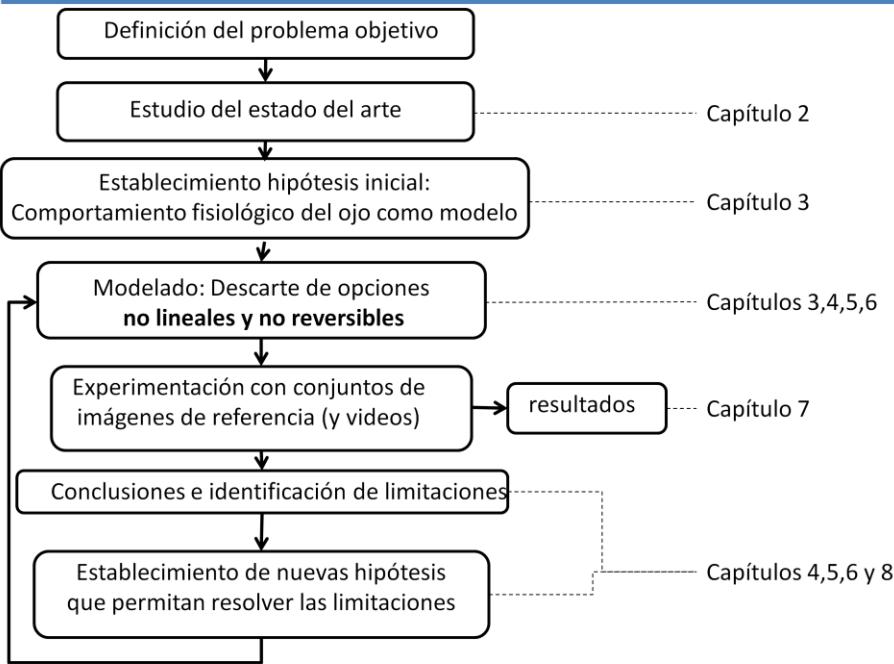
Validación del modelo de color YUV (YUV 444, YUV 422 y YUV 420) con el nuevo algoritmo.

Creación del mecanismo de utilización de LHE en compresión de video, es decir, codificación de información diferencial temporal y primera comparativa con el codificador estándar de mayor difusión (H264).

Los objetivos técnicos concretos para lograr resolver el cloud gaming y el consumo energético que afecta a los terminales móviles son los siguientes:

El objetivo fundamental es el cloud gaming, y para ello quiero desarrollar una nueva tecnología de compresión, basada en unas ideas básicas iniciales que quiero desarrollar como es la métrica de PR y el downsampling elástico y que finalmente son llevadas a cabo con éxito

0 | Introducción: metodología

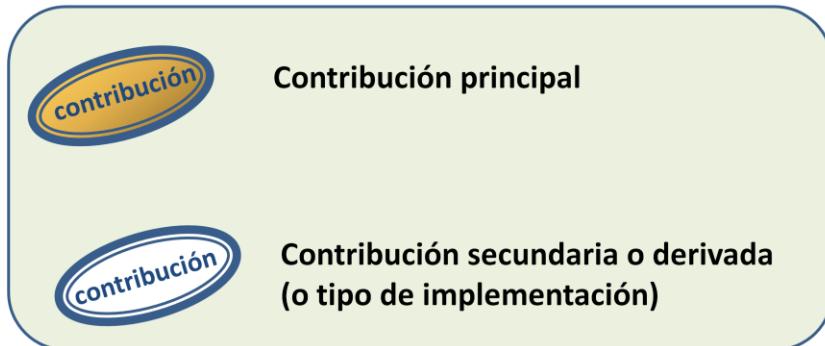


La metodología empleada en el desarrollo de esta tesis es inductiva, validando iterativamente hipótesis falsables.

De color verde las partes que han exigido mas creatividad

Ademas del descarte de opciones no lineales se descartan las opciones no reversibles, ya que necesitamos que el decodificador pueda reconstruir la imagen original... esto es una obviedad y puede liar al oyente, dandole la impresión de que me refiero a algo mas de lo que quiero simplemente decir

0 | Introducción: etiquetas



7

Diferenciar entre contribución principal y lo que sería una sub parte de dicha contribución principal. De modo que puede haber una contribución principal que se pueda subdividir en varias subcontribuciones.



0| Introducción

1 | Estado del arte

2| LHE básico

3| LHE avanzado

4| Decodificador LHE

5| Video LHE

6| Resultados

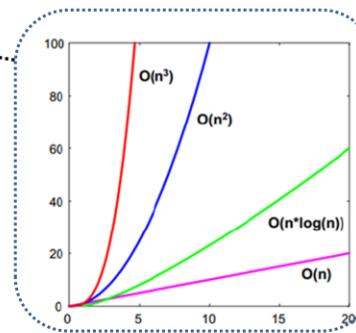
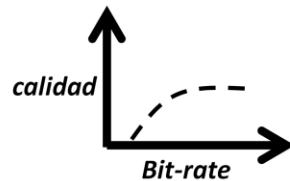
7| Conclusiones y trabajo futuro

1 | Estado del arte: complejidad

Selección y diseño de algoritmos

criterios

- Espacio-tiempo*
- Paralelización*
- Calidad del resultado*



Típico diagrama R-D
No muestra la complejidad, sólo el resultado

9

Cuando comparamos algoritmos para escoger el mejor, realmente no hay una sola variable en la que basarse sino que tenemos :

el consumo de recursos computacionales de espacio (memoria) y tiempo. Si nos fijamos en la figura de la izquierda veremos como diferentes ordenes de complejidad computacional pueden elevar drásticamente los tiempos de cálculo y eso es algo en lo que nos hemos de centrar para el diseño de un algoritmo que queremos que sea muy eficiente.

La capacidad de paralelización que aunque no disminuye coste si puede reducir el tiempo de ejecución

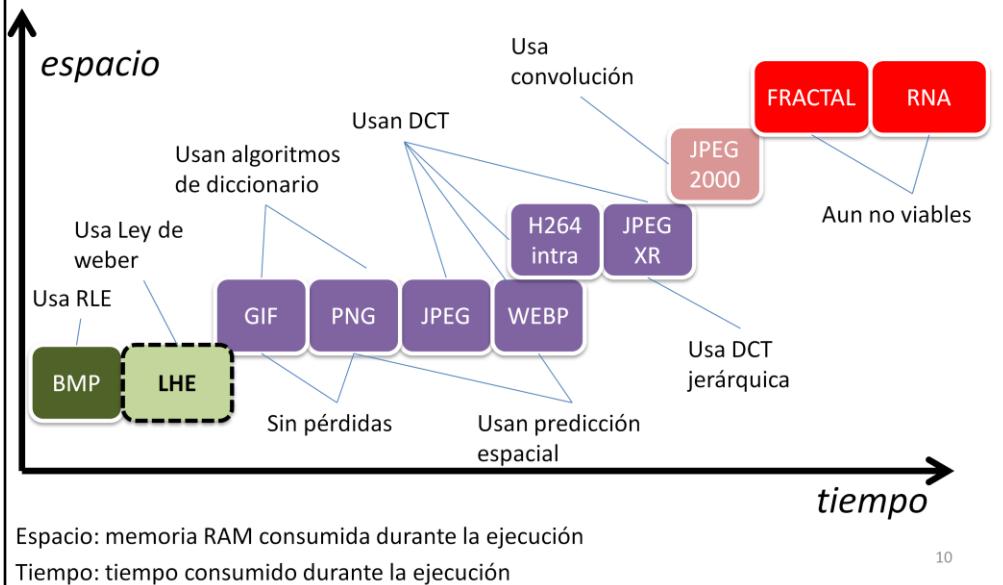
Y por último la Calidad del resultado, sobre todos si nos centramos en compresión multimedia

Normalmente los algoritmos se presentan con un diagrama rate-distortion como este, el cual nos permite comparar algoritmos sin tener en cuenta otra cosa que el resultado y no el coste.

Si comparamos el coste y no el resultado tendremos algo como esto

1 | Estado del arte: algoritmos/formatos

Esta gráfica muestra la complejidad computacional y no cuanto comprime cada formato ni cuanta calidad proporciona.



Esta grafica muestra la complejidad computacional en recursos de memoria eje vertical y tiempo u operaciones matematicas en el eje horizontal sin dar cuenta de la calidad del resultado de cada algoritmo sino solo de su coste

Pues bien, El nuevo algoritmo se ubica entre los menos cososos computacionalmente por ser lineal y no involucrar transformaciones de frecuencia. De hecho los avances que normalmente encontramos en publicaciones suelen ser trabajos sobre mejoras en DCTs que aprovechan un tipo de simetria axial en imágenes médicas, mejoras de implementacion de una DCT sobre un determinado HW, mejoras en la codificacion de coeficientes de la transformada de coseño, etc

pero nadie hasta ahora ha investigado un mecanismo de codificacion nuevo como es la ley de weber, que es precisamente la ley que GOBIERNA nuestros sentidos (vista, oido , tacto) asi como otros comportamientos fisiologicos del ojo humano. Y esto es la clave de su baja complejidad y de la calidad de sus resultados

1 | Estado del arte: transform compressors



J. Cooley y J. W. Tukey, «An Algorithm for the Machine Calculation of Complex Fourier Series,»
Mathematics of Computation, vol. 19, nº 90, p. 297–301, 1965.

$$S \neq DCT^{-1} (DCT (S))$$

Transform compressors (dominan la compresión con pérdidas)

- JPEG 1986 -1994
- MPEG1 Intraframe (JPEG) 1992
- MPEG2 Intraframe (JPEG) 1994
- Mp3 (audio) 1993
- JPEG 2000
- H264 intra 2003
- JPEGXR 2009
- WebP (VP8 intraframe) 2010
- H265 intraframe "HEVC" 2013
- ...

VS

Non- Transform compressors (dominan la compresión sin pérdidas)

- G721 (ADPCM speech codec) 1984
- GIF 1987
- Lossless JPEG (1993)
- PNG 1996
- FLAC (audio) 2000
- JPEG- LS (2003)
- ...

11

En 1965 se publicó el primer artículo sobre la transformada rápida de Fourier. Aunque inicialmente fue creada con usos militares, para detectar mediante sismógrafos, los posibles ensayos nucleares soviéticos, enseguida fue utilizada ampliamente para el procesamiento de señal y en concreto para la compresión con pérdidas. Con la llegada de internet en los 80 se vio claramente la necesidad de desarrollar un formato con pérdidas para almacenar y transmitir imágenes como fue JPEG.

El motivo por el que no se utiliza el paso al dominio de la frecuencia para la compresión sin pérdidas es que es muy difícil mantener bajo control el error porque es una transformación no reversible completamente. Sin embargo la cuantización y eliminación de componentes frecuenciales de una señal ha resultado ser una técnica muy adecuada para eliminar información de una imagen o del audio sin dañar demasiado la calidad.

Esto ha hecho que los transform compressors dominen la transformación con pérdida mientras que los non transform compressors dominan la transformación sin pérdidas, como GIF o PNG. De hecho los pocos intentos que se han hecho en el espacio con técnicas similares al ADPCM para transformar con pérdidas no han dado los resultados con la calidad que ofrece, por ejemplo JPEG. Y por ello nunca se contempló el dominio del espacio seriamente para la compresión con pérdidas.

Y Desde que nació JPEG muchas mejoras han surgido siempre trabajando en el dominio de la frecuencia, compactando cada vez más la energía al tiempo que se paga el precio de aumentar la complejidad computacional

Y Puesto que yo lo que buscaba era un nuevo algoritmo cuya mayor ventaja sea la eficiencia computacional, decidí evitar la costosa transformación a frecuencia y trabajar el espacio, rompiendo de este modo con la corriente de los compresores con transformación y explorando otro rumbo diferente

1 | Estado del arte: fisiología del ojo

Comportamientos fisiológicos actualmente aprovechados en compresores:

Comportamiento del ojo	Aplicación
El ojo es más sensible al cambio de luminancia que al de crominancia	Modelo de color YUV (usado en todos los compresores con pérdidas, incluido LHE)
El ojo es bueno detectando pequeños cambios de brillo en áreas relativamente grandes ("estrellas") pero no cuando el brillo cambia rápidamente en pequeñas áreas (variación de alta frecuencia - ruido).	Cuantización más bruscamente de los coeficientes correspondientes a las altas frecuencias (usado en todos los "transform compressors")
El ojo responde con diferente precisión a diferentes frecuencias	Multichannel (usada en DWT)
La respuesta del ojo es logarítmica	LHE
El ojo no detecta cambios de brillo por debajo de un umbral mínimo	LHE
El ojo se adapta al brillo medio local	LHE

12

Para trabajar en el dominio del espacio me he basado en el comportamiento fisiológico del ojo y en esta tabla se resumen los comportamientos del ojo que son aprovechados en los compresores actuales

Hay una serie de comportamientos del ojo que no están siendo explotados por los algoritmos actuales como son 1,2,3

NO ADELANTARSE A EXPLICAR EN QUE CONSISTEN ESTAS 3 CARACTERISTICAS



0| Introducción

1| Estado del arte

2| LHE básico

3| LHE avanzado

4| Decodificador LHE

5| Video LHE

6| Resultados

7| Conclusiones y trabajo futuro

2| LHE básico

Basado en modelado del ojo humano

contribución

- La respuesta logarítmica al estímulo luminoso (**ley de Weber**)
- El umbral mínimo de contraste lumínico para detectar cambios
- El proceso de acomodación del ojo al brillo medio local



Figura 64. Diagrama de bloques del compresor LHE básico

14

LHE esta basado en el modelado del comportamiento fisiológico del ojo humano

Concretamente LHE modela a 3 características fundamentales de su comportamiento

UNA VEZ MAS: NO EXPLICAR, SOLO ENUMERAR!!! Así ahorro tiempo y se explica después

Respuesta logarítmica o ley de weber que gobierna nuestros sentidos

Umbral mínimo: en penumbra nos cuesta siempre detectar diferencias de tonos

El proceso de acomodación del ojo al brillo medio local: el ojo tiene mucha capacidad para diferenciar tonos de gris pero cuando miramos un punto de una imagen cualquiera, el ojo pierde capacidad de diferenciar tonos que se alejan del brillo medio de ese punto, es una propiedad local

Dicho esto, El codificador básico de LHE se va componer un cuantizador logarítmico que transforma la señal en saltos logarítmicos o hops y codificador que transfore esos hops en códigos binarios

A continuación voy a detallar como se modelan cada una de estas tres características del ojo en el algoritmo LHE

2| LHE básico: respuesta logarítmica al estímulo luminoso

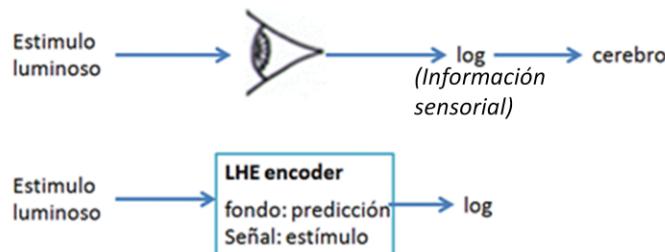
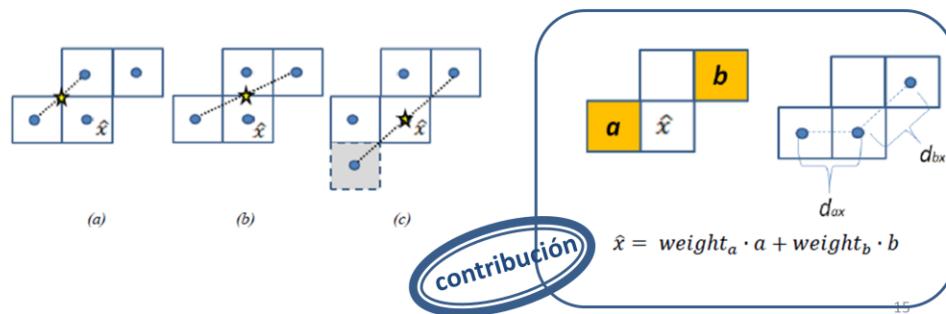


Figura 49. Modelado de la respuesta logarítmica del ojo humano



Respuesta logarítmica al estímulo luminoso

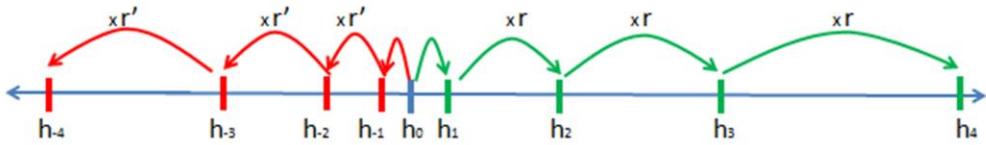
El ojo lo que detecta son variaciones lumínicas respecto del brillo de fondo y transforma estímulos luminosos en información sensorial que envía al cerebro. La respuesta del ojo no es proporcional al estímulo sino que la información sensorial es logarítmica. Esto significa que Para percibir el doble de luz hace falta en realidad 10 veces más luz.

Para modelar esto hacen falta dos cosas. en primer lugar estimar cual es el brillo de fondo que a la vez va a ser la predicción del brillo que hace para cualquier pixel y en segundo lugar cuantificar logarítmicamente la diferencia de la señal con ese brillo de fondo.

Para estimar el brillo de fondo hay varias posibles estrategias pero la que mejor resultado da es la media ponderada entre el pixel situado a la izquierda y el superior derecho. Esto proporciona un valor estimado que está centrado en el eje X. Lo ideal sería también centrar la predicción en el eje Y pero LHE funciona escaneando pixel a pixel, scan line a scanline y los pixels inferiores no están disponibles de modo que opto por utilizar la opción b

Si considerasemos en la predicción el izq y superior, la predicción sería correcta para la esquina entre ambos pixels que es donde se encuentra el punto medio, pero teniendo en cuenta el izq y el supedere el punto medio se encuentra más cerca del pixel que queremos predecir y por consiguiente nuestra predicción es mejor

2| LHE básico: respuesta logarítmica al estímulo luminoso



$$h_2 = h_1 \cdot r \quad h_3 = h_2 \cdot r \quad h_4 = h_3 \cdot r$$

$$h_{-2} = h_{-1} \cdot r' \quad h_{-3} = h_{-2} \cdot r' \quad h_{-4} = h_{-3} \cdot r'$$

$$r_{\max_pos} = \left(\frac{0.8 \cdot (255 - h_0)}{h_1} \right)^{1/3} \quad r = \min(2.5, r_{\max_pos})$$

$$r_{\max_neg} = \left(\frac{0.8 \cdot h_0}{h_1} \right)^{1/3} \quad r' = \min(2.5, r_{\max_neg})$$

contribución

16

LHE distribuye de forma logarítmica a lo largo del intervalo disponible una serie de saltos o hops centrados en lo que hemos estimado como brillo medio, a lo que llamo h_0 . La relación entre los hops es la razón geométrica r . Como vemos en las ecuaciones que relacionan todos los hops entre si.

Codificaremos la señal con el hop más cercano al valor original de cada pixel. Es decir que si un pixel está más cerca de h_2 que del resto dicho pixel se transforma en un 2, simplemente, no en el valor de la luminancia, sino en el índice del hop, en este caso el 2.

En cuanto al cálculo de la razón, debe de ser suficiente para que los hops cubran los intervalos positivo y negativo disponible. Una razón muy grande expandiría los hops y se podrían salir del intervalo y una muy pequeña los compactaría, ganando precisión en zonas suaves con tonos similares pero perdiéndola en saltos bruscos de la luminancia. Esto nos lleva a controlar el valor de la razón para mantenerla dentro de unos márgenes a través de las ecuaciones de razón positiva y negativa que se muestran en pantalla

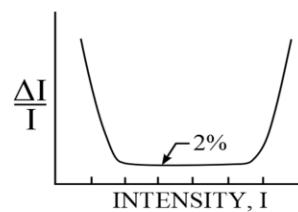
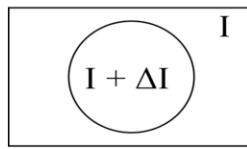
El hop1 es peculiar pues no hay un hop menor con el que guarda la relación r , y pequeñas fluctuaciones de h_1 van a impactar en todos los hops. Su valor debe ser no menos que lo que el ojo humano pueda distinguir, eso nos lleva al segundo modelado: umbral mínimo de detección que veremos en la siguiente diapositiva.

Para cubrir con hops el 80% del intervalo disponible usaremos una razón por defecto de 2.5 pero debemos tener en cuenta que para evitar que los hops más grandes se salen del intervalo que queremos recubrir a veces habrá que topar la razón, y cuanto más nos aproximemos a un extremo más necesario será hacerlo. Y ese tope lo he llamado r_{\max}

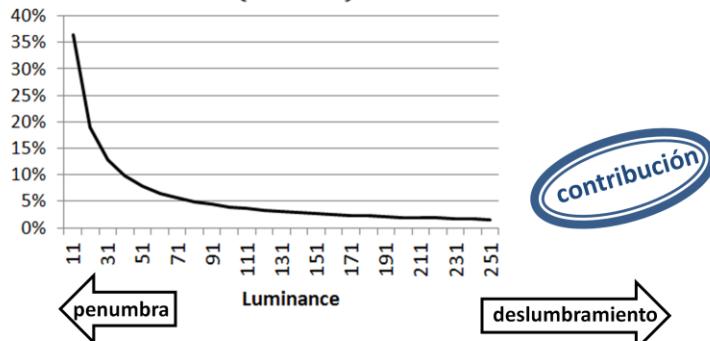
Dire también que al estar los hops separados logarítmicamente, el intervalo cubierto por cada hop es asimétrico y el valor central del intervalo será la luminancia que le demos a un pixel codificado por un hop. Esto no aplica al h_1 ni al h_4 . No aplica al h_1 porque entonces nos alejamos del 2% de la fracción de Weber y no aplica al h_4 porque si se compactan los hops esto aleja muchísimo el último hop pues su cota superior del rango cubierto por el hop es 255

2| LHE básico: umbral de detección h1

Fracción de Weber:



Weber fraction using $\Delta I = 4$
($h_{1\min} = 4$)



17

La segunda de las propiedades fisiológicas del ojo que modela LHE es el umbral de detección

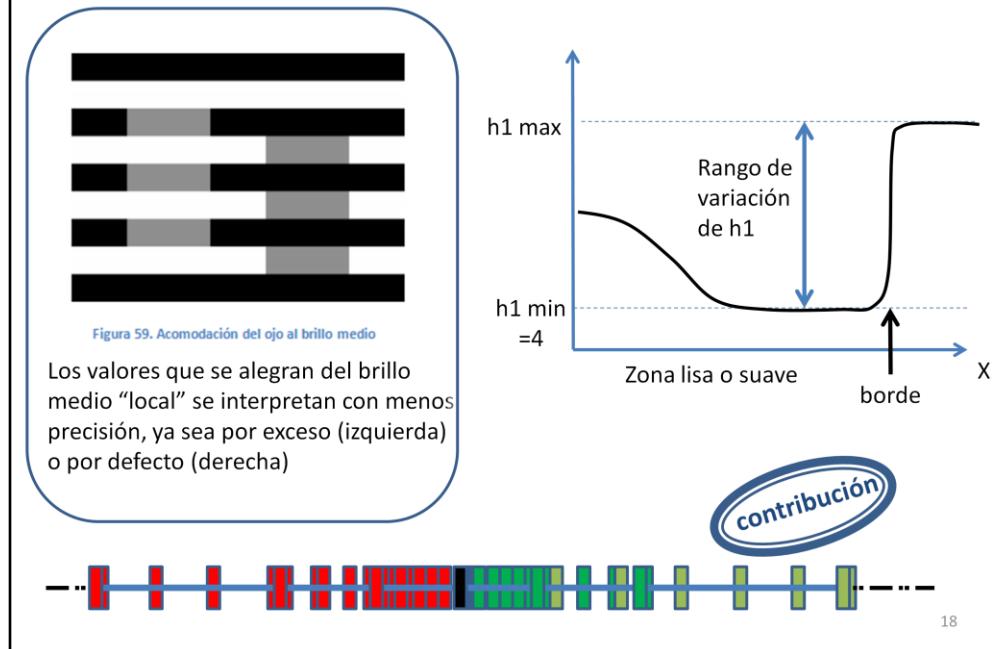
Para modelarlo consideramos La fracción de weber que es una medida de la sensibilidad del ojo. a mayor valor, menor sensibilidad.

En la oscuridad la sensibilidad del ojo disminuye. A medida que aumenta la luz llegamos a un 2% de valor

En situaciones de deslumbramiento la sensibilidad vuelve a disminuir pero los dispositivos de visualización se ajustan para no brillar hasta el deslumbramiento, por eso la parte de la curva correspondiente al deslumbramiento no la voy a modelar

Este 2% se traduce para una señal de luminancia considerada entre 0 y 255 en un valor de aproximadamente 4 y ese será el valor mínimo de h_1 , el menor de los hops, pero no su valor máximo que como veremos puede variar en función de cómo el ojo se adapte al brillo local de la imagen como veremos a continuación

2| LHE básico: acomodación al brillo medio local



Esta imagen es muy interesante. Aparece un tono gris claro y un tono gris mas oscuro. Sin embargo son el mismo tono

Aunque los dos tonos de gris son idénticos percibimos como mas claro el izquierdo. Es debido a la acomodación al brillo medio del ojo humano. Debido a las franjas blancas y negras el brillo medio de la izquierda es diferente al de la derecha y el ojo percibe bien variaciones cercanas al brillo medio pero percibe con error (ya sea exceso o defecto) variaciones que se alejan del brillo medio. El ojo es en realidad sensible a solo 50 tonos de gris alrededor del brillo medio (las famosas 50 sombras de grey--- que muchos conocereis como el titulo de una novela y pelicula)

El brillo medio de la izquierda es mayor por encontrarse delimitado por franjas blancas. Es un gris medio pero el ojo le asigna un valor erróneo mas brillante del que le corresponde. En el caso derecho ocurre lo contrario, solo que el error que comete el ojo es hacia un brillo mas oscuro. Siempre que un tono se aleja del brillo medio se pierde precisión en su interpretación por el ojo, ya sea por exceso o por defecto.

Tambien ocurre que un tono muy oscuro es percibido como un negro en las cercanías del blanco, en ese caso se comete un error por defecto,

Lo que hace LHE es ganar precisión cuando el brillo medio se estabiliza, compactando los hops entorno al brillo medio. Para ello ira reduciendo el valor de H1 desde un valor maximo (10) hasta el umbral minimo de detección que hemos dicho anteriormente que es 4. Si de repente el tono a codificar se aleja mucho (un borde) asignaremos automaticamente el valor maximo de h1 que puede ser 10 , para lograr codificar con mejor precision valores lejanos y los hops se EXPANDIRAN de nuevo

2| LHE básico: número de hops



Figura 53. Ejemplo de asignación de hops y reconstrucción realizada por LHE



19

En la imagen del barco vemos como se traducen a hops los valores de la señal y como son posteriormente reconstruidos con cierto error pero con calidad aceptable. Hemos reducido de 256 grados de libertad a solo 9 y podemos comprimir iconos, cosa que con jpeg es imposible a menos que se conserven todas las componentes de frecuencia

El motivo para escoger 9 hops es experimental. Con 9 hops la curva de calidad se estabiliza y no nos proporciona mucha mas calidad seguir añadiendo hops. El caso limite serian 256 hops en cuyo caso cada luminancia se corresponderia con un hop. Pero con +-4 es suficiente

2 | LHE básico: codificador entrópico

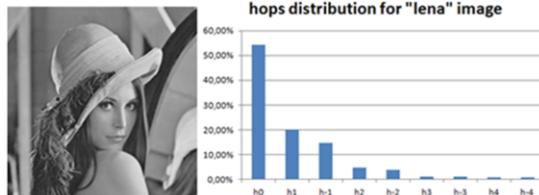
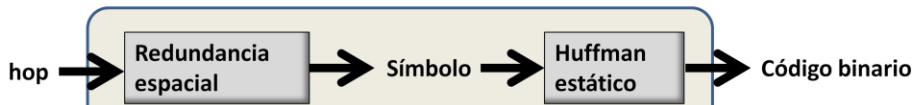


Figura 61. Distribución estadística de hops obtenidos por LHE para la imagen "Lena"

Símbolo: número de comprobaciones [1..9]

Hop	Orden de comprobación
h0	1
Up	2
h1	3
h-1	4
h2	5
h-2	6
h3	7
h-3	8
h4	9
h-4	10

Por último se calculan códigos binarios de longitud variable mediante Huffman estático

20

Una vez que LHE ha transformado toda la imagen en hops, es decir cada pixel se ha transformado en un hop, llega el momento de codificar en binario estos hops. Para ello utilizo una estrategia en dos etapas. Primero aprovechamos las redundancias espaciales para transformar los hops en simbolos. Y luego transformamos los simbolos en códigos de longitud variable usando el algoritmo de huffman estatico.

El hop mas frecuente es el hop nulo. De hecho la distribucion estadistica de hops es muy asimetrica. Eso nos permite asignar a los simbolos códigos de longitud variable usando huffman estatico.

La redundancia espacial basicamente consiste en que un pixel se suele parecer al anterior (o lo que es lo mismo el hop mas frecuente es el hop nulo). Seguidamente y por orden de probabilidad el siguiente hop mas frecuente será el situado en la posición superior del pixel, debido a la redundacia espacial H , en la que una linea se suele parecer a la linea inmediatamente anterior.

El hop UP puede representar a cualquier hop y si nuestro hop coincide con su valor lo transformamos en el simbolo 2.

el simbolo será siempre el numero de comprobaciones que hagamos hasta que coincida el hop con un elemento de la tabla

2 | LHE básico: paralelización

Dependencias entre pixels:

1	2	3	4
3	4	5	6
3	6	7	8
4	8	9	10

1	2	3	4	5	6	7	8
2	4	5	6	7	8	9	10
3	6	7	8	9	10	11	12
4	8	9	10	11	12	13	14
5	10	11	12	13	14	15	16
6	12	13	14	15	16	17	18
7	14	15	16	17	18	19	20
8	16	17	18	19	20	21	22

Figura 65. Paralelización de operaciones

N: lado

$$3N - 2 \quad vs \quad N^2$$

Incremento de velocidad (lado=1024)

↑ 350 veces

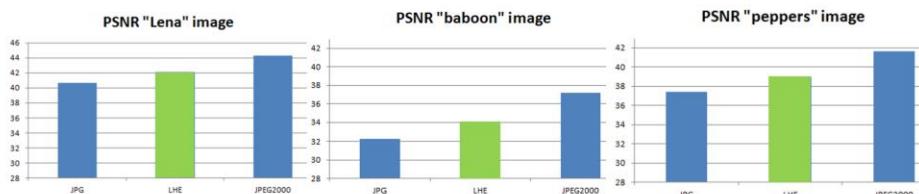
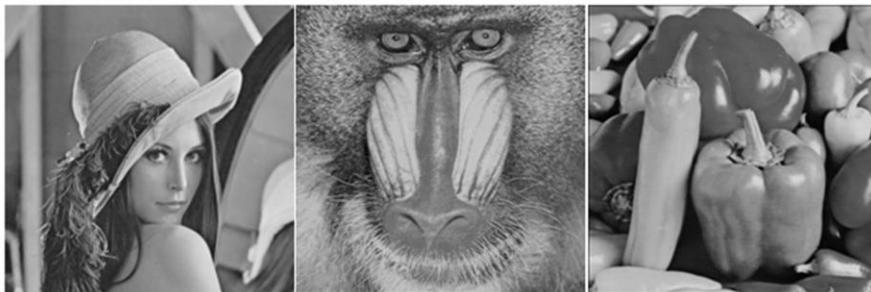
21

La paralelización en LHE básico tiene la limitación del cálculo del brillo medio, para el que necesitamos tener previamente codificados el izq y supdere

Eso limita la paralelización. Un bloque de 16 pixels se puede hacer en 10 pasos y uno de 64 pixels en 22. y en general si el lado es N, son $3N-2$ pasos

Puedo decir que la velocidad aumenta entre 200 y 500 veces. Por ejemplo una imagen de N=512 es 200 veces más rápida y si N=1280 es 500 veces más rápida

2 | LHE básico: resultados



- Estrictamente lineal $O(N)$
- Bit-rate no elegible

22

La calidad es intermedia pero la complejidad es la mas baja

Ademas es estrictamente lineal tal como se demuestra en ejecuciones secuenciales con un prototipo java

La limitacion es que no hemos elegido cuanto queriamos comprimir, solo disponemos de un ratio que ademas no escogemos



0| Introducción

1| Estado del arte

2| LHE básico

3 | LHE avanzado

4| Decodificador LHE

5| Video LHE

6| Resultados

7| Conclusiones y trabajo futuro

Puede haber muchos LHE avanzados. Es construir un compresor basado en LHE basico pero que permita elegir el bit rate

Hay muchas posibilidades y opciones abiertas a la imaginación

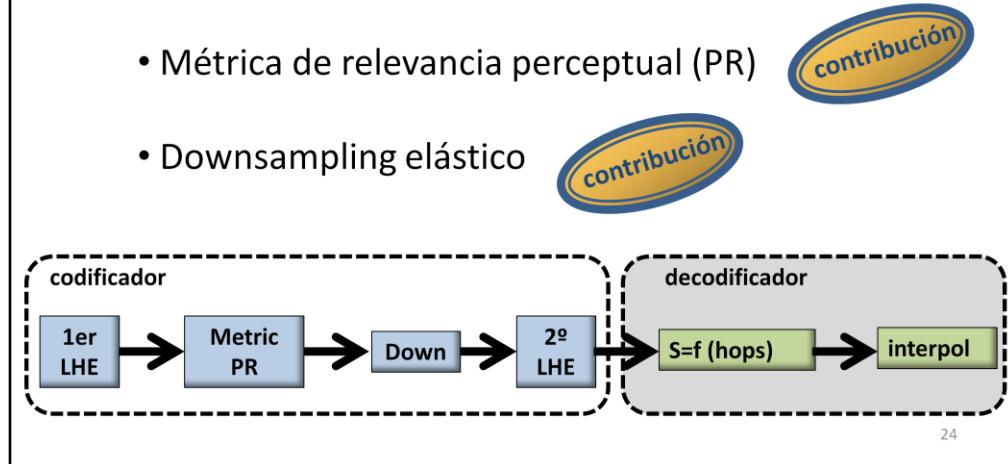
3 | LHE avanzado

Necesidad:

El LHE básico ofrece un solo ratio de compresión, que además no podemos elegir

LHE avanzado está basado en:

- LHE básico
- Métrica de relevancia perceptual (PR)
- Downsampling elástico



Lhe avanzado esta basado en lhe basico y en dos contribuciones que nos van a permitir lograr un bitrate variable que es precisamente lo que no tenemos con LHE basico

Estad dos contribuciones son :

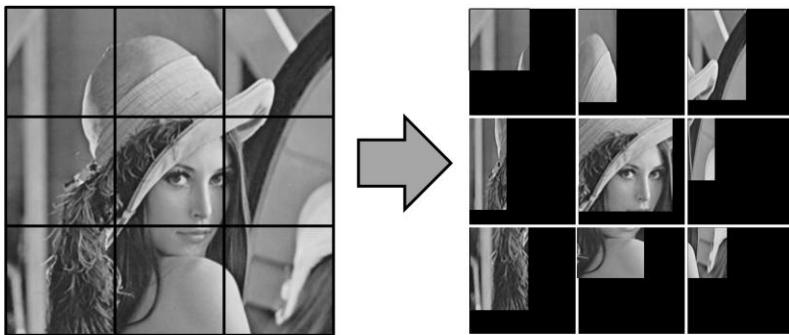
-una metrica de RP que nos permita medir la importancia de la informacion sensorial para la percepcion de la mente humana y asi **tener un criterio** para degradar mas las áreas menos importantes de la escena.

-y en segundo lugar, un mecanismo de downsampling elastico que nos permita dividir la imagen en bloques y submuestrearlos a diferente cadencia en funcion de su metrica de relevancia perceptual

El mecanismo completo aparece representado en el diagrama: primero calcularemos el LHE basico de la imagen, calculamos las metricas y realizamos un downsampling elastico, ejecutando un segundo LHE basico sobre los bloques submuestreados
En decodificacion haremos el proceso inverso , en lugar de downsampling elastico haremos interpolacion elastica para regenerar la imagen original

3 | LHE avanzado

Sub-muestreado elástico



cada bloque de la imagen es sub-muestreado con diferente intensidad

Dentro de un bloque la cadencia de muestras es variable entre sus 4 esquinas

25

Lo que vemos en esta imagen ilustra lo que he comentado de submuestrear de diferente modo cada bloque en función de la información que contiene

Pero además, dentro de cada bloque vamos a tomar diferentes muestras en función de una métrica de relevancia perceptual que calculemos en cada una de sus 4 esquinas, de modo que no solo estamos ante un submuestreo heterogéneo sino que además dentro de un mismo bloque es elástico. Ahora ilustrare mejor esta cuestión

3 | LHE avanzado: PPP (Pixels Per Pixel)

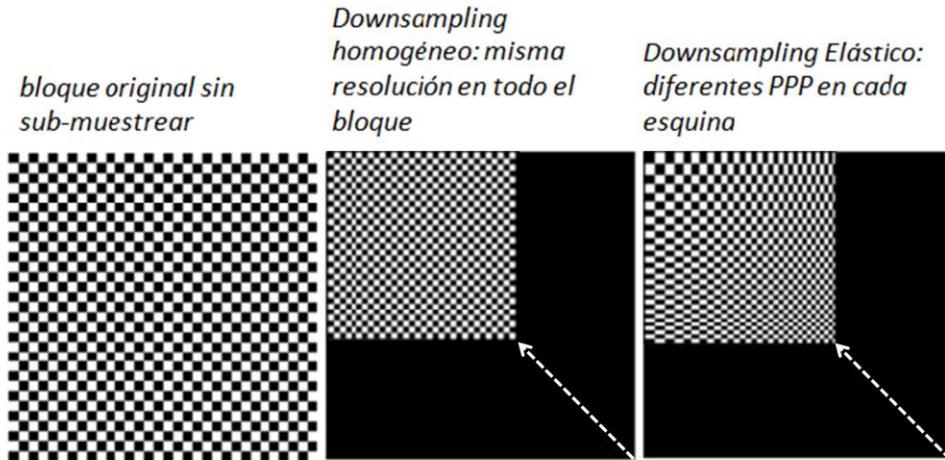


Figura 73. Downsampling homogéneo vs Downsampling Elástico

26

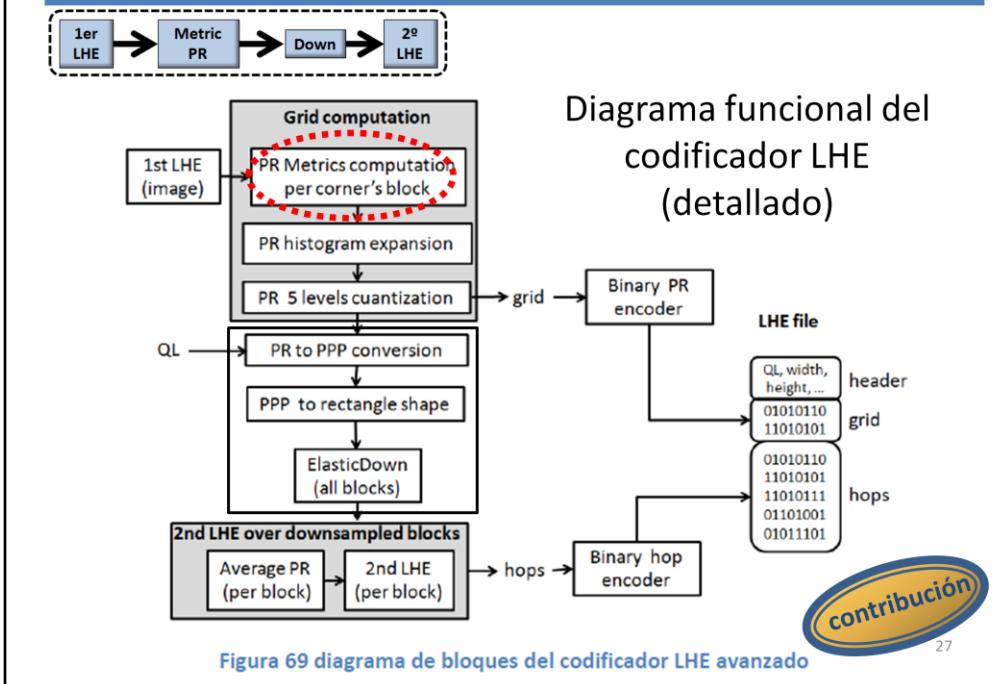
En esta imagen aparece una comparativa de lo que seria un submuestreado homogeneo convencional y el downsampling al que he llamado elástico.

-dentro de cada bloque la cadencia con que se toman las muestras tanto en el eje X como en el eje Y puede ser variable. Esto significa que cada pixel del bloque muestreado representará más o menos pixels de la imagen original. Es lo que he llamado PPP.

-En el ejemplo, un pixel submuestreado de la esquina inferior derecha se corresponde con varios pixeles de la imagen original, mientras que un pixel de la esquina superior izquierda se corresponde con un solo pixel original, conservando mayor detalle en dicha esquina.

La elección de los PPP en las esquinas de un bloque laharemos en función de una métrica a la que llamaremos relevancia perceptual y que explicaré a continuación. El muestreado elástico nos permite emplear mas muestras allí donde se necesitan, donde hay detalles, y menos donde no se necesitan, en zonas lisas ogradadas.

3 | LHE avanzado: diagrama bloques funcionales



Esta es la arquitectura del codificador que como he comentado anteriormente se compone de 4 bloques que son el primer LHE, el calculo de la metrica de relevancia perceptual, el downsampling elastico y el segundo LHE.

El diagrama mas detallado lo vemos aquí y Lo mostrare a lo largo de la presentacion para ubicar el contenido que explicaré en cada momento

Tras el primer LHE básico tenemos el bloque de calculo de la metrica "grid computation" en el que se van a ejecutar las operaciones de cálculo de la métrica de PR para cada esquina de los bloques y se van a ajustar y cuantizar para ser despues almacenadas o transmitidas, en lo que llamaré la "información de malla"

Despues hay una traducción de las métricas de PR a PPP, un downsampling elástico y por ultimo un segundo LHE sobre los bloques sub-muestreados.

A la derecha tenemos los codificadores binarios tanto de la informacion de la relevancia perceptual de las esquinas de la malla como de los hops en los que se han transformado los bloques submuestreados de la imagen

3 | LHE avanzado: Relevancia Perceptual

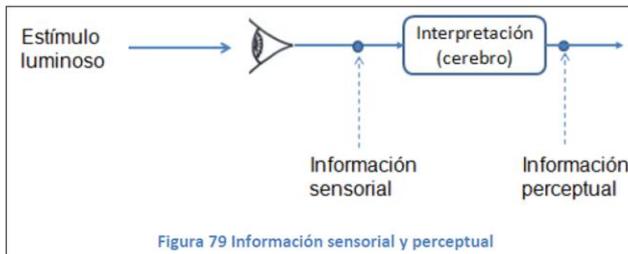


Figura 79 Información sensorial y perceptual

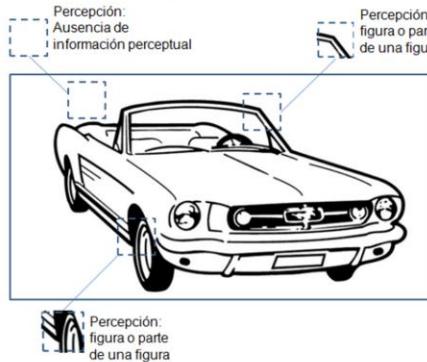


Figura 80. Distintas percepciones en fragmentos de una imagen

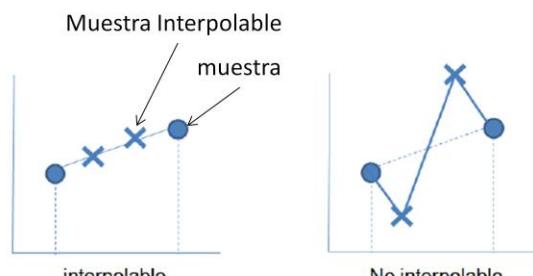


Figura 81. Interpolabilidad de la señal

el ojo transmite información sensorial al cerebro para su interpretación. Esta información sensorial está expresada logarítmicamente y en nuestro modelado su equivalente son los hops de LHE

Tras la interpretación que el cerebro hace de estos hops se obtiene la información perceptual, que es la interpretación de lo que estamos viendo (por ejemplo, un coche).

Nuestra métrica va a medir **cuanto de relevante es una información sensorial para su percepción**. Diferenciaremos lo que es liso o gradado de lo que contiene bordes abruptos o suaves. Y lo haremos tanto en la dirección X como en Y, ya que en base a esta métrica vamos a asignar posteriormente los PPP de las esquinas de los bloques

Lo que tiene que lograr la métrica de PR es calcular la interpolabilidad de la señal, pues si es constante o gradada entonces es interpolable con pocas muestras y eso es porque contiene poca relevancia perceptual.

Si por el contrario la pendiente de la señal contiene cambios de signo , entonces no es posible interpolar con pocas muestras

Ademas La magnitud de los hops que cambian de signo tambien nos indican lo grave el error cometido al interpolar

3 | LHE avanzado: Relevancia Perceptual

Un bloque puede contener más información sensorial (3 bordes) pero no por ello es menos interpolable que otro que contiene un solo borde. Lo que importa de cara a la percepción es si es un borde suave o abrupto, pero no si hay tres bordes o solo uno.

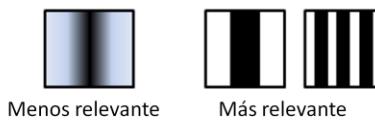
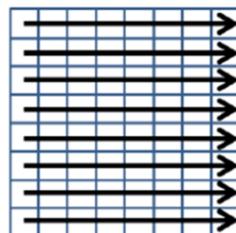


Figura 82. Relevancia es independiente del número de bordes

$$PR_x = \frac{\sum_{i=1}^{C_x} |h_i|}{C_x \cdot |H_{max}|}$$



Al ser una media, esta métrica es independiente de la resolución.

29

la métrica de PR va a tener en cuenta no cuantos bordes hay en un bloque sino cuantos abruptos son. Nos va a dar igual si hay un borde o tres bordes, lo que importa es los abruptos que son. Esto es así porque para reproducir la señal, aunque haya un solo borde necesitaremos muchas muestras ya que por tener un solo borde no vamos a necesitar menos, pues con pocas muestras el borde se difuminará y se perderá

En este caso he representado la formula correspondiente a la metrica de relevancia perceptual en la coordenada horizontal. Como vemos la metrica trabaja con hops, no con luminancias

Si Un borde implica un hop h_i , su importancia normalizada sería h_i/h_{max}
La métrica al final lo que esta calculando es la media de la importancia de los bordes que recorre en cada una de las scanlines que constituyen el bloque

El mismo razonamiento se aplica para calcular la PRy

3 | LHE avanzado: Relevancia Perceptual

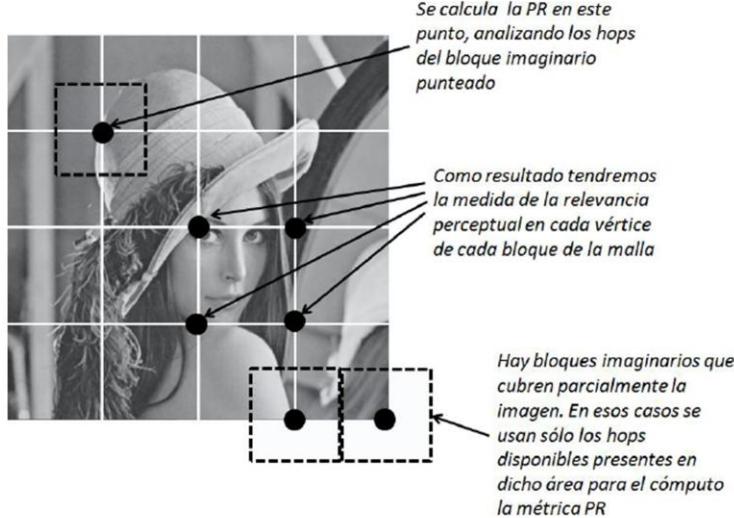


Figura 88. Métricas de PR en los vértices de la malla

30

La imagen es dividida en una malla de bloques y en cada esquina de la malla calcularemos la métrica teniendo en cuenta los hops correspondientes a un bloque imaginario centrado en cada esquina.

Traduciremos posteriormente la metrica a PPP y como en las esquinas los bloques comparten el valor de ppp , la resolucion de las fronteras de los bloques sera la misma y no habra efecto de bloques como el que se produce en JPEG debido a las diferencias de componentes de frecuencia entre bloques.

3 | LHE avanzado: malla

Malla de bloques sin tamaño fijo

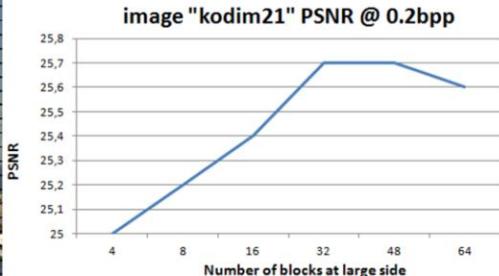


Figura 77. Diferentes resultados con Kodim21 usando diferente número de bloques en la malla

Malla de bloques sin tamaño fijo: **numero de bloques constante** (sin precedentes)
La información de malla es la misma en SD, HD o UHD

24 + 44 imágenes

Kodak

USC University of
Southern California
Signal and Image Processing Institute
Ming Hsieh Department of Electrical Engineering



31

El número de bloques que vamos a considerar va a ser siempre el mismo. Para sacarle el máximo rendimiento al downsampling elástico los bloques no pueden ser muy pequeños porque la info de malla sería grande, y además si son muy pequeños no tendría sentido asociar diferentes PPP a cada esquina.

Ni muy grandes porque como los PPP evolucionan linealmente no podríamos aumentar los PPP a la mitad de un bloque y luego volver a reducirlos. El tamaño que mejor resultados de calidad ha dado **sobre las bases de datos de imágenes de referencia de kodak y la galería miscelánea de la universidad de california (un total de unas 70 imágenes = 24+44)** ha sido 32 de ancho. Con un numero de bloques fijo, la información de malla es constante e igual en SD, HD o UHD

3 | LHE avanzado: Saturación de PR

Nuestros sentidos se llegan a saturar

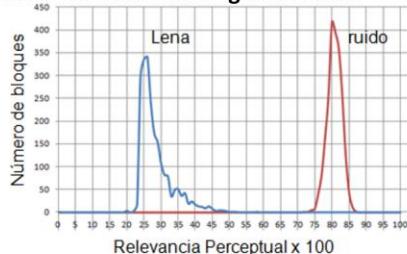


Figura 91. Histograma de valores de la métrica de PR_x y PR_y para dos imágenes diferentes

Umbral de saturación
PR=0.5

Igual número de muestras

contribución

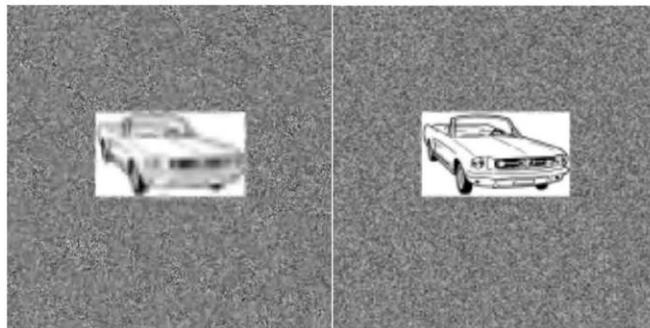


Figura 92. Resultados sin topar y topando la PR en 0.5. Ambas imágenes igualmente reducidas al 10%

En la imagen se muestra el histograma de la relevancia perceptual de dos imágenes, lena y una imagen de ruido aleatorio

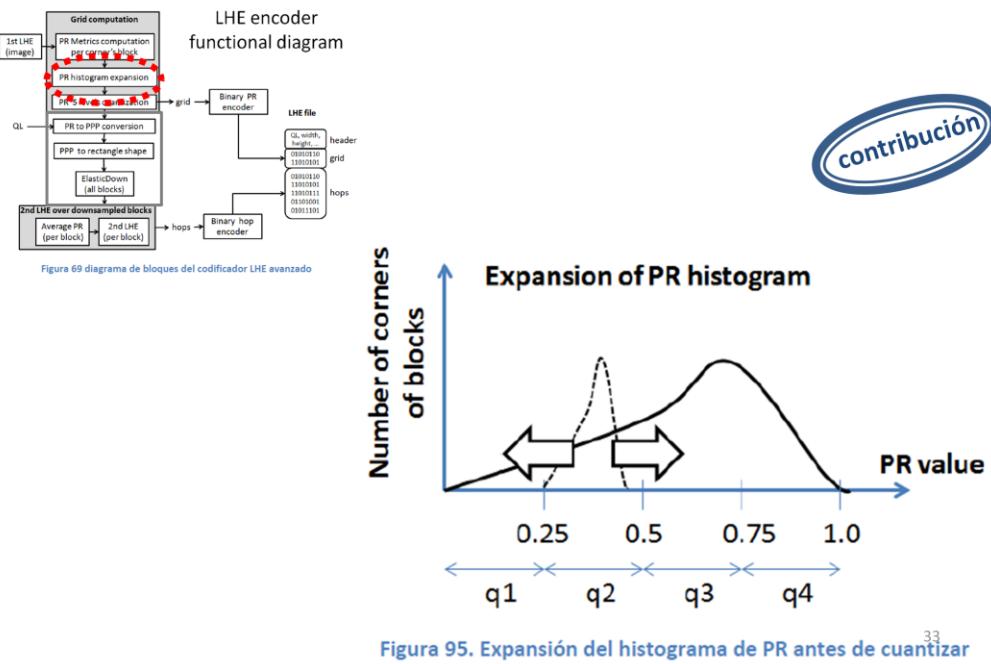
cuando un área es muy ruidosa su PR es muy elevada pero nuestros sentidos están saturados y no merece la pena proteger mejor una zona ruidosa que una zona de alta PR que tiene bordes pero no llega a ser ruidosa.

Si asignamos siempre menos PPP cuanto mayor sea la PR, estaremos protegiendo el ruido más que los bordes. Este umbral es configurable y lo he establecido en 0.5 que es cuando un pixel no va a poder representar más de dos pixeles reales para poder interpolar correctamente la señal original

Esta característica de hecho se corresponde con la capacidad del que es bueno detectando pequeños cambios de brillo en áreas relativamente grandes (“estrellas”) pero no cuando el brillo cambia rápidamente en pequeñas áreas (variación de alta frecuencia - ruido)

Estableciendo este límite, es decir, topando la métrica para que no supere 0.5, no malgastaremos mas muestras en proteger mejor zonas ruidosas que bordes, tal y como se muestra en las dos imágenes, la primera sin la métrica saturada y la segunda con la métrica saturada. En la segunda, el coche queda mejor protegido a costa de no proteger mas el ruido aunque tenga mas relevancia que 0.5

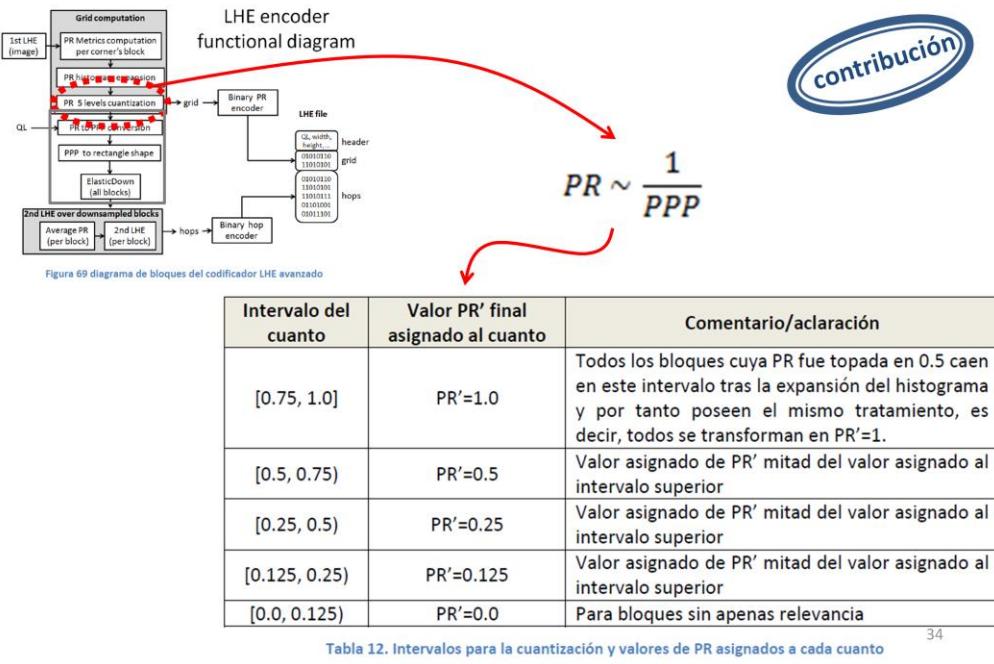
3 | LHE avanzado: expansión de PR



No podemos almacenar la relevancia perceptual como numeros flotantes porque ocuparian mucho y tampoco merece la pena. De modo que vamos a cuantizarla, para lo cual lo primero que dedo hacer es expandir el histograma para acentuar las diferencias entre las diferentes partes de la imagen. Si no lo hiciese, el proceso de cuantización posterior podria igualar lo que es diferente. En el ejemplo representado todo el histograma original (el que aparece con linea punteada) caeria en el mismo cuanto, pero con la expansión vamos a proteger mejor las zonas con mas detalle.

La expansión la haremos igual para cualquier imagen, expandimos desde 0.2 (muy suave) hasta 0.5 (muy abrupto), que es el limite topado de PR. **El mecanismo de expansión no es una contribución pero si la idea de aplicar la expansión sobre la métrica de relevancia perceptual para diferenciar un histograma estrecho en 5 cuantos y asi asignar posteriormente PPPs diferentes a cada cuanto de PR**

3 | LHE avanzado: cuantización de PR



a continuación voy a cuantizar en 5 posibles cuantos o valores la metrica ya expandida pero no asignando el valor medio de cada intervalo sino distribuyendo de forma geométrica los valores de los cuantos.

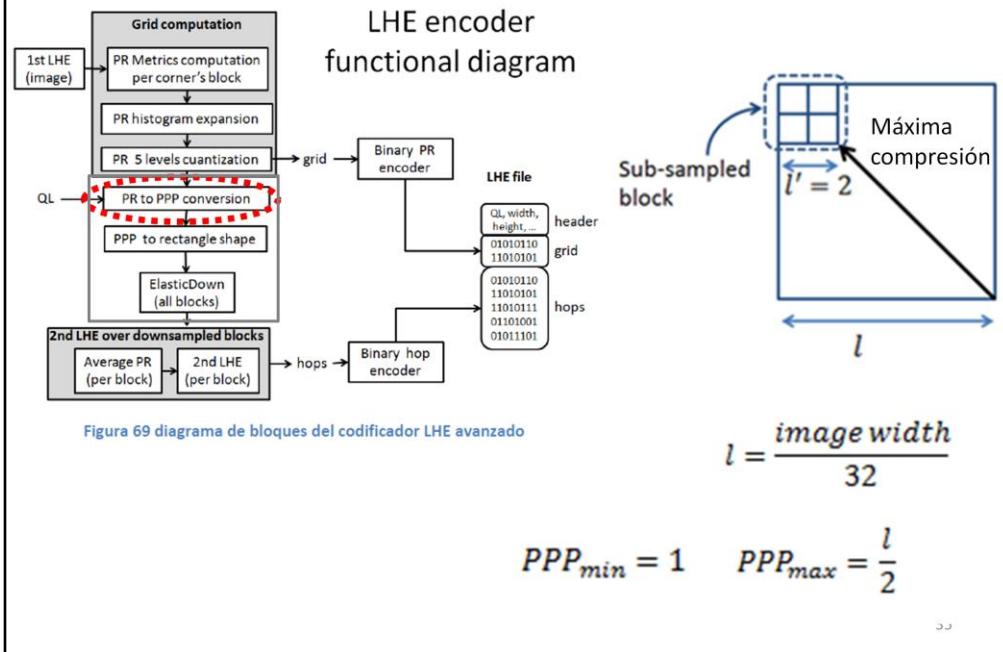
En la tabla se muestran los valores asignados a los 5 cuantos en progresion geometrica de razon 2, es decir, 0,125,025,05 y 1. un cuanto siempre es el doble del anterior

Esta asignación acentúa las diferencias de PR y genera los mejores resultados de calidad

La explicación matemática se encuentra en la relación entre PPP y PR', la cual es inversamente proporcional (como se presentará más adelante). Esto implica que incrementos lineales de PR no se traducen en incrementos lineales de PPP, sino que si una de ellas aumenta el doble, la otra disminuye a la mitad. Si usase cuantos distribuidos linealmente estaría siendo incoherente con la relación entre ambas variables

En otras palabras, para poder representar el doble de detalles (pixels intermedios) hace falta muestrear el doble de pixels (PPP mitad) y para ello hay que doblar el valor de PR'.

3 | LHE avanzado: traducción de PR' a PPP



Ya nos encontramos en el tercer bloque del codificador

El siguiente paso una vez que tenemos la metrica de PR expandida y cuantizada es traducirla a PPP para saber cuanto vamos a submuestrear en cada esquina de cada bloque

Una de las cosas que debemos tener en cuenta es que cada bloque será reducido como mucho a 4 muestras, uno por cada esquina, y como hemos definido una malla donde los bloques no tienen un tamaño fijo, el valor máximo de PPP se calcula como aparece en la formula y el valor minimo es logicamente 1, pues un pixel muestreado no puede representar menos de un pixel original. Esto equivale a la máxima precision pues en ese caso no hay sub-muestreo

3 | LHE avanzado: traducción de PR' a PPP

$$CF = CF_{min} \cdot r^{99-QL} \rightarrow PPP = \frac{PPP_{max} \cdot CF_{min} \cdot r^{99-QL}}{1 + (PPP_{max} - 1) \cdot PR'} \text{ donde } QL \in [0,99]$$

Ecuación 23. Nivel de calidad ("QL" o "Quality Level") y reformulación de PPP

contribución

Un valor de QL define una “calidad equivalente”

Una misma Relevancia perceptual (PR) se transforma en los mismos PPP

23,8dB



16,4dB



Figura 103. Otras dos imágenes con igual nivel de calidad (QL) y distinto PSNR y tasa de bit

Dos imágenes con el mismo valor de QL tendrán la misma “calidad equivalente” 36

Esta es la formula, cuyo desarrollo aparece en la tesis , que relaciona la PR expandida y cuantizada con el PPP. Hay un parametro que aparece en la formula que es el QL que seria el quality level o nivel de calidad de la compresión

QL toma valores entre 0 y 100 y permite dividir la calidad (el parametro CF de la ecuacion de la izqda) en 100 posibles niveles separados de forma geometrica. Para que incrementos constantes de QL produzcan incrementos constantes de calidad.

El motivo para escoger una razón geométrica es el siguiente: cada vez que subimos de nivel, estaremos multiplicando por r al factor de compresión CF y por lo tanto estaremos añadiendo un cierto porcentaje de muestras sobre las ya existentes. Si no usásemos una razón, y dividiésemos el intervalo de valores de CF en partes iguales, cada nivel no implicaría una mejora porcentual, por lo que los niveles más altos apenas producirían mejoras significativas, mientras que los niveles bajos producirían cambios enormes de calidad.

En cuanto a CFmin es una constante que depende de PR'minimo (0.125)y de PPP max y cuyo valor aparece desarrollado en la tesis

NO DECIR: Un ajuste adicional: si pr'=0 ponemos PPP=PPPmax en lugar de usar la ecuacion

El significado fisico de QL es que dos imágenes con el mismo QL se codifican igual de bien o de mal, es decir, la misma relevancia perceptual calculada se va a traducir en los mismos PPP. Esto lo ilustra la imagen de mikey, donde las flores hacen que la cuenta de errores aumente y el PSNR baje, pero la calidad de mikey es la misma en ambas imágenes pues tiene las mismas metricas de relevancia perceptual en las esquinas de la malla que lo contiene. Por ello digo que ambas imágenes tienen la misma calidad equivalente, aunque el PSNR sea diferente

3 | LHE avanzado: traducción de PR' a PPP

Ajuste de gradiente de PPP

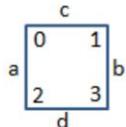
$$l'' = \text{INT}(0.5 + l')$$

$$\alpha' = \frac{PPP_2 - PPP_1}{(l'' - 1)}$$



Figura 106. Downsampling Elástico de la imagen Lena sin restricciones

Restricción rectangular



teniendo en cuenta que $l' = \frac{2l}{PPP_1 + PPP_2}$
(Siendo PPP_1 y PPP_2 los valores PPP en los extremos de un segmento), entonces si estamos ante un rectángulo:

$$PPP_{0x} + PPP_{1x} = PPP_{2x} + PPP_{3x}$$

$$PPP_{0y} + PPP_{2y} = PPP_{1y} + PPP_{3y}$$



Ecuación 29. Condición rectangular

37

Queda un ultimo paso antes de poder realizar el doewnsampling elastico y es lo que he llamado restriccion rectangular.

En la ecuacion aparece reflejado el gradiete de PPP, (α) que permite evolucionar los ppp de una esquina a otra y debe considerar un numero entero de pixels en la scanline submuestreada. Si aplicamos dicho gradiente, aunque cada scanline trenga un numero entero de pixels, van a ser de diferente tamaño unas de otras, dando lugar a una imagen submuestreada que no es un rectangulo y que por lo tanto no se le va a poder aplicar el segundo LHE, ya que LHE requiere recorrer la imagen en scanlines

Aplicar restriccion rectangular implica quitar PPPs de una esquina y darselos a otra de forma proporcional para conseguir que la forma final sea un rectangulo. Este reparto proporcional no se muestra aquí pero se detalla en ecuaciones desarrolladas en la tesis p151 (no aquí)

L' es $2l/(p1+p2)$ y sale de una serie aritmetica

Se deben hacer 2 cosas: ajustar los PPP para que l'' sea un numero entero de pixeles pero tambien ajustar los PPP para que quede un rectangulo

El gradiente de PPP debe tener en cuenta que vamos a reducir a un numero entero de pixels (l'')

Pero ademas,

3 | LHE avanzado: Downsampling Elástico

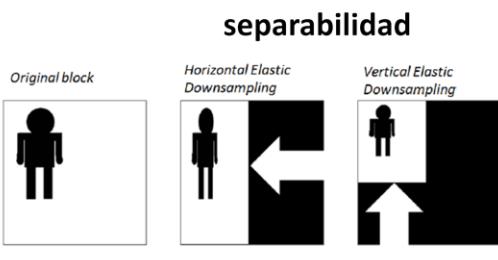
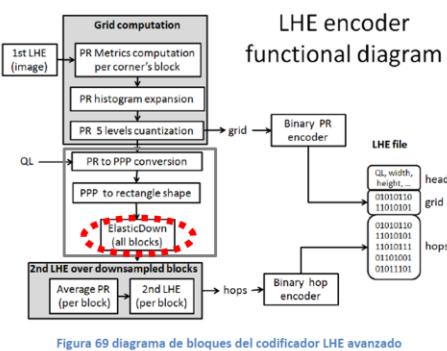
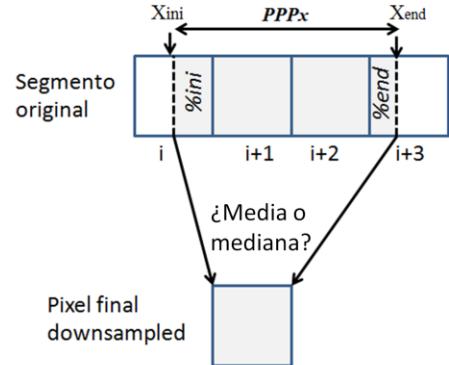


Figura 113. Downsamplig Elástico separable



Una vez que tenemos los PPPs de cada esquina podemos hacer el downsampling elastico y lo vamos a hacer de forma separable. El downsampling convencional u homogeneo es una operación matematicamente separable y el elastico tambien.

Al hacer este down debemos calcular la luminancia de cada pixel submuestreado como la media de las luminancias de los pixels enteros o parciales que representa.

Esta estrategia como veremos en los resultados difumina un poco los bordes y seria mas adecuado una mediana, al menos para conservar bordes abruptos en la version submuestreada de la imagen. Esto es algo que esta contemplado como mejora futura

Si hablamos de imágenes en color, el procedimiento es el mismo solo que los PPP son la mitad en YUV 422 y la cuarta parte en YUV420

3 | LHE avanzado: Downsampling Elástico



Original image: Kodim20



Conventional (detail)



Elastic downsampling



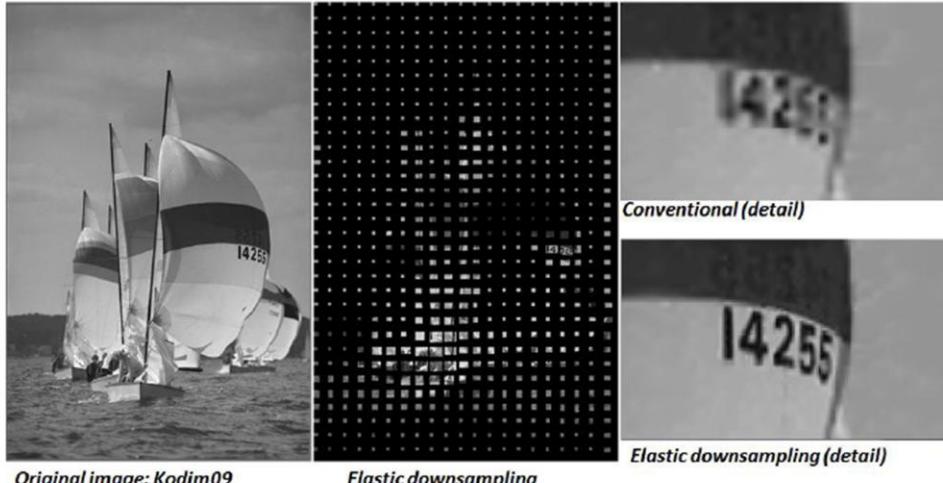
Elastic downsampling (detail)

39

Esto es un ejemplo de lo que se logra con el downsampling elastico.comparado con un downsampling convencional

Podemos ver como en el cielo es submuestreado de un modo mas intenso por tener poca relevancia perceptual y todas esas muestras que nos ahorramos en el cielo las empleamos en las zonas relevantes, obteniendo un resultado de una calidad superior al downsampling convencional, a igual numero total de muestras

3 | LHE avanzado: Downsampling Elástico



En este otro ejemplo podemos observar un resultado similar y nuevamente con el mismo numero de muestras logramos una calidad superior con el downsampling elastico que con el convencional

3 | LHE avanzado: Downsampling Elástico

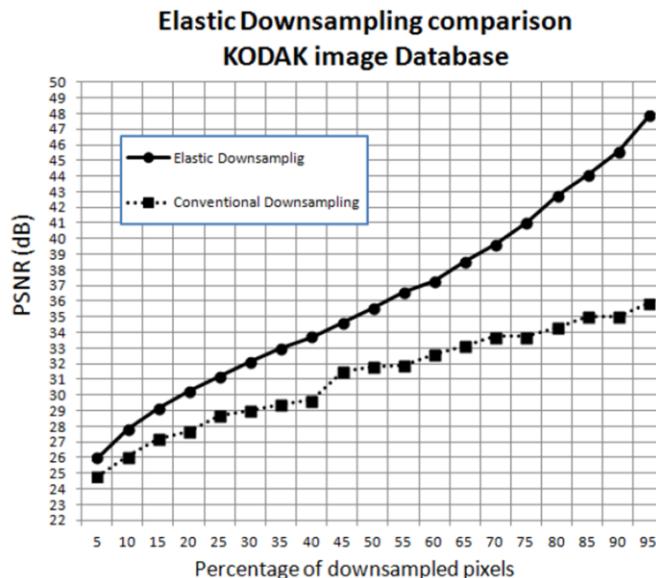


Figura 115. Comparación de Downsampling Elástico vs Downsampling homogéneo

41

Esta es una curva comparativa entre el PSNR obtenido con downsampling convencional y el elastico a igualdad de muestras.

Aquí no interviene el 2do LHE, solo la señal original submuestreada en función de las métricas de relevancia perceptual

El down convencional tiene una caída dramática desde el principio porque elimina muestras en zonas críticas desde el principio mientras que el elastico elimina muestras donde menos se necesitan, manteniendo un nivel de calidad superior en todo momento

3 | LHE avanzado: segundo LHE

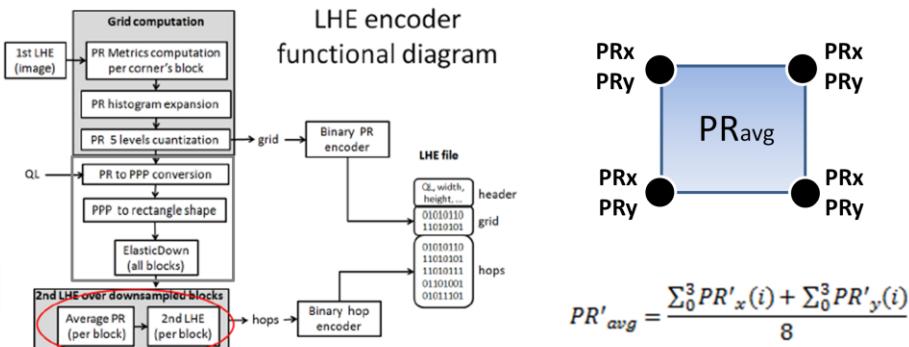
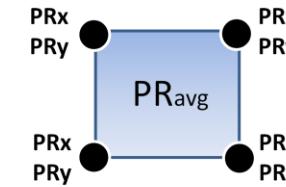


Figura 120. Ubicación del contenido que se va a desarrollar



$$PR'_{avg} = \frac{\sum_0^3 PR'_x(i) + \sum_0^3 PR'_y(i)}{8}$$

Ajuste de parámetros de LHE en función de PR'avg

- Razón geométrica
- Valor máximo de h1



42

Hemos llegado al cuarto y ultimo de los bloques del codificador en el que vamos a realizar un segundo LHE sobre los bloques ya submuestreados elasticamente

Conociendo las métricas de PR de cada esquina de cada bloque, y calculando su media, podemos ajustar mejor los parametros de la cuantizacion logaritmica, es decir tanto la razón geométrica de los hops, que en el primer LHE era de 2.5, como el valor maximo de h1 que se corresponde con el umbral mínimo de contraste lumínico para detectar cambios, lo que llamabamos fraccion de weber

un bloque con un PR elevado se codifica mejor con una razón grande que separe los hops y un h1 maximo también grande que permita cubrir mejor todo el intervalo disponible.

Por el contrario un bloque suave con un PR bajo se codifica mejor compactando los hops, es decir con un h1 maximo menor y una razón menor, de modo que queden mas compactados los hops y puedan representar con mayor fidelidad variaciones de tonos cercanos entre si.

No decir :La tabla de valores se ha obtenido de modo experimental aunque los motivos expuestos son la explicación de estos valores obtenidos

3 | LHE avanzado: segundo LHE

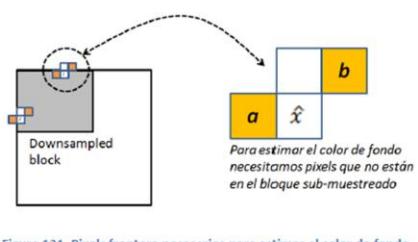
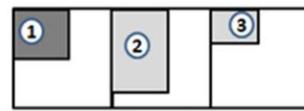
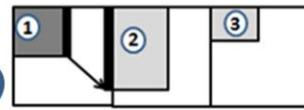


Figura 121. Pixels frontera necesarios para estimar el color de fondo



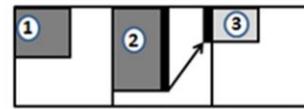
Aplicamos LHE en el bloque 1



Interpolamos la frontera del bloque 1



Aplicamos LHE en el bloque 2



Interpolamos la frontera del bloque 2

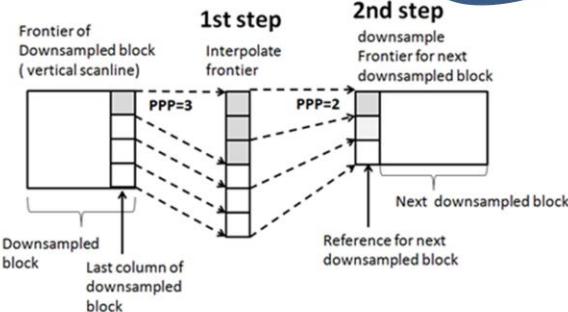


Figura 124. Interolar es adaptar longitud pero tambien adaptar los PPP

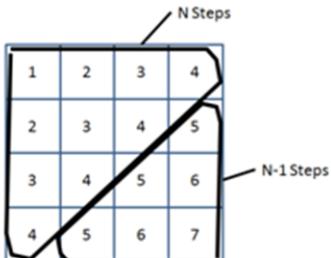
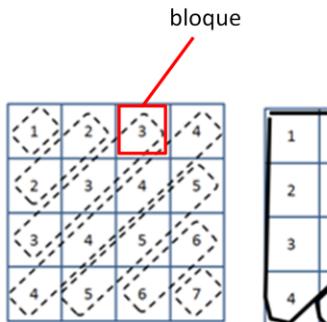
43

Para realizar el segundo LHE podemos considerar cada bloque como una nueva imagen y aplicarle un LHE basico o bien podemos hacerlo un poco mejor, considerando las muestras disponibles en la fronteras de cada bloque, mejorando el calculo la estimacion del brillo de fondo en las fronteras de los bloques y en consecuencia mejorando la calidad

Para hacerlo debemos estirar la frontera de cada bloque anterior hasta el lado del siguiente. Ello limita nuestra capacidad de paralelizacion porque para empezar a aplicar LHE un bloque necesitamos la frontera del bloque anterior, la cuales solo tenemos tras aplicarles el LHE. Esto significa que no se pueden procesar todos los bloques simultaneamente pero a cambio ganamos algo de calidad. Dependiendo del tipo de aplicacion puede ser interesante una u otra opcion. Por ejemplo si queremos codificar fotogramas en cloud gaming puede ser interesante codificar cada bloque de forma independiente y ganar asi capacidad de paralelizacion.

Estirar la frontera consiste en adaptar la longitud de la frontera al siguiente bloque pero tambien adaptar los PPP pues estamos ante un submuestreado elastico. Por ello se puede primero estirar y despues submuestrear con los valores adecuados de PPP

3 | LHE avanzado: paralelización del segundo LHE



N : número de bloques de lado

$$2N - 1 \quad vs \quad N^2$$

Incremento de velocidad (lado =32 bloques)

Usando fronteras \uparrow 16 veces (63 pasos frente a 1024)

sin usar fronteras, se pueden paralelizar todos los bloques \uparrow 1024 veces

44

Ahora vamos a ver como se puede paralelizar usando fronteras entre bloques.

En la figura de la izquierda he representado un ejemplo con los bloques en que se descompone una imagen. Bloques con el mismo numero se pueden ejecutar a la vez. Una imagen de $N \times N$ nos llevaria $2N - 1$ pasos

En lhe basico la paralelizacion se hace en terminos de pixels y el factor limitante es el calculo de la predicion o brillo medio de cada pixel. En el 2º lhe el factor limitante son las fronteras.

Sin usar fronteras en una imagen 1280 x 720p (32x18 bloques) hay 576 bloques y eso es el incremento de velocidad respecto de una ejecucion secuencial: 576 veces mas rapido. En una imagen cuadrada (32x32 bloques) hay 1024 bloques y eso seria el incremento

Usando fronteras el incremento de velocidad respecto de una ejecucion secuencial es 16 veces

3 | LHE avanzado: codificadores entrópicos

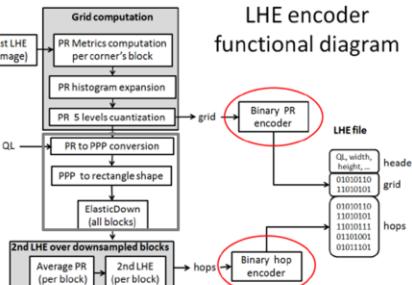


Figura 127. Ubicación del contenido que se va a desarrollar

contribución

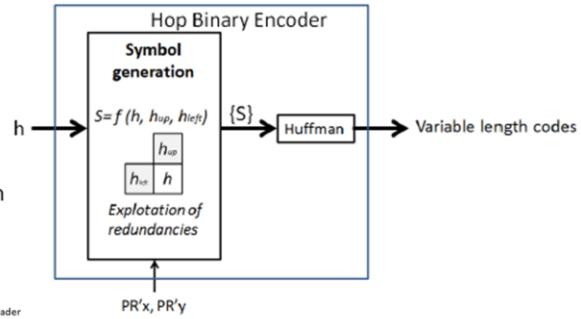


Figura 129. Las dos etapas del codificador binario de hops

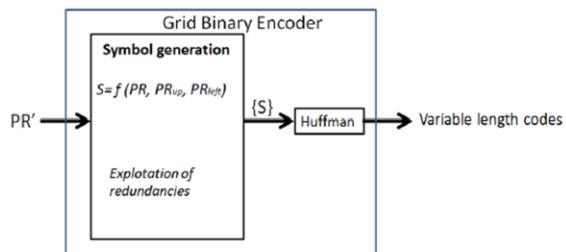


Figura 132. Las dos etapas del codificador binario de malla

45

Ahora voy a pasar a explicar Los codificadores binarios, los cuales son muy similares al de LHE basico

Primero transforman hops en simbolos aprovechando las redundancias espaciales. Se hacen comprobaciones sobre cada uno de los hops contra una tabla de valores ordenada por probabilidad y el numero de comprobaciones es el simbolo. En este caso ademas, si los valores de la metrica de PR nos indican que el bloque es mas horizontal que vertical podemos poner en primer lugar la si el hop es igual al superior (comprobacion UP) y luego la de hop nulo , mejorando el proceso de transformacion.

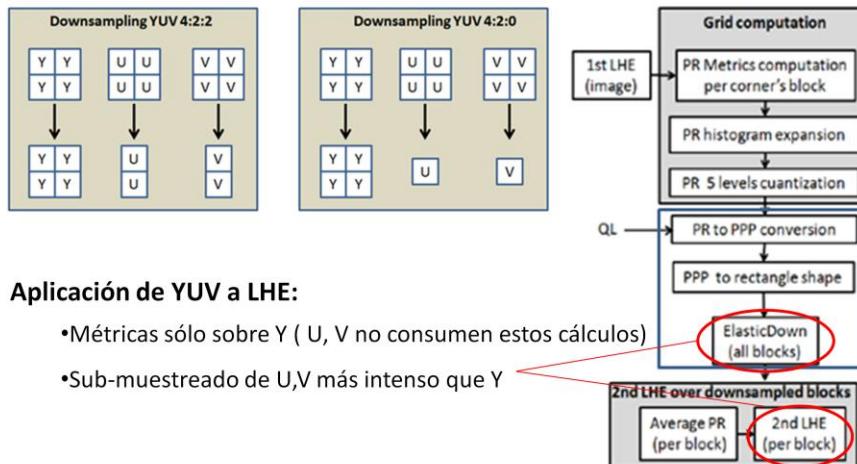
Despues aplicamos huffman estatico

En cuanto a la malla, se pueden aprovechar redundancias espaciales porque bloques suaves suelen estar cerca de bloques suaves y bloques abruptos suelen estar cerca de bloques abruptos. Esto significa que la metrica de PR se va a parecer a la de la esquina anterior y/o superior. Empleando la misma filosofia que con los hops comprimiremos los cuantos de PR

3 | LHE avanzado: modelo de color

Modelos de color: CIE, RGB, HSB, CMYK, YUV

YUV: es el modelo de los “lossy compresores” (JPEG, JP2K, h264 intra, webp...)



46

Existen muchos modelos de color como el CIE, RGB, CMYK, pero Modelo de color en LHE es YUV, al igual que en muchos otros compresores de imágenes

El ojo es mucho más sensible al cambio de luminancia que al de crominancia, por este motivo es más eficiente invertir más bits en la luminancia que en la crominancia. Es más adecuado que otros modelos para comprimir, pues en caso de compresión con pérdidas se puede degradar más las componentes de color que la luminancia. Dentro del modelo YUV hay dos posibilidades que son el YUV422 y el YUV420 que comprimen a la mitad o a la cuartaparte el color respecto de la luminancia

La aplicación de este modelo no implica nada mas que ejecutar el downsampling y el segundo LHE sobre las componentes de crominancia pues todo el proceso anterior de las metricas calculadas con la luminancia fijan el muestreo de la luminancia y el de crominancia siempre va a ser ese mismo a la mitad o a la cuarta parte. Esto significa que nos ahorraremos muchos pasos del proceso llevado a cabo con la luminancia

3 | LHE avanzado: complejidad y paralelización

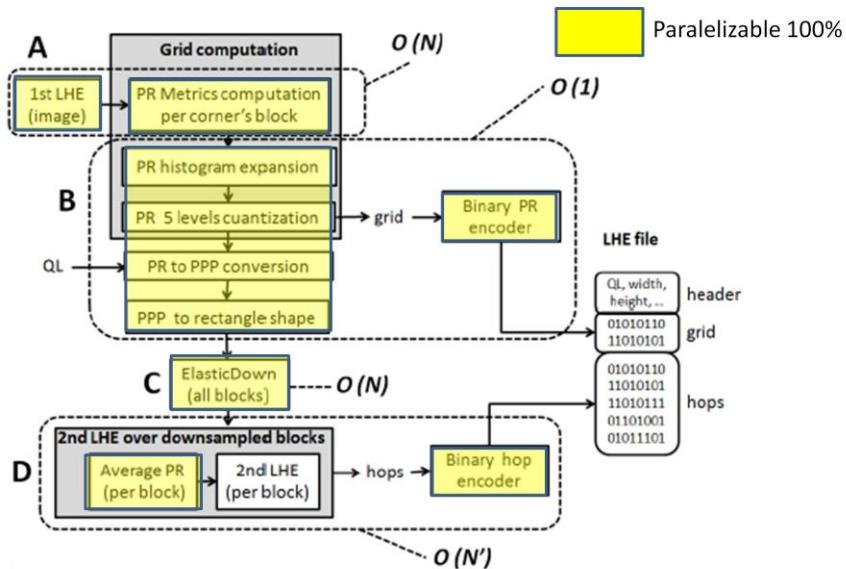


Figura 134. Macrobloques de LHE avanzado en función de su complejidad algorítmica

47

Aquí Voy a analizar el coste computacional del compresor

el primer LHE puede hacerse submuestreando por SPS la imagen como he demostrado en la memoria, obteniendo unos valores de PR idénticos, con variaciones en torno al 1%. Esto significa que aunque es complejidad N, el numero de operaciones puede ser N/4 por ejemplo o incluso un valor fijo pues podemos tomar una imagen submuestreada de ancho 256 para llevar a cabo el proceso.

Todos los calculos sobre la metrica de relevancia perceptual y PPPs son de complejidad 1 pues son calculos sobre nodos de la malla y como mucho hay 1024 nodos si la imagen es cuadrada o menos si es rectangular. Si comparamos este numero de nodos con los pixeles de una imagen que pueden ser millones, nos damos cuenta de que estos calculos son despreciables en el computo total de la imagen

A continuacion el downsampling tiene complejidad N y por ultimo el segundo LHE tiene N' pues trabaja con menos de N pixels ya que ha sido submuestreado

En cuanto a la paralelizacion, todos los pasos en amarillo son paralelizables al 100%. Pero ademas si se quiere se puede hacer el segundo LHE 100% paralelizable eliminando las dependencias entre bloques, es decir, no usando fronteras



0| Introducción

1| Estado del arte

2| LHE básico

3| LHE avanzado

4 | Decodificador LHE

5| Video LHE

6| Resultados

7| Conclusiones y trabajo futuro

4 | Decodificador LHE

LHE decoder functional diagram

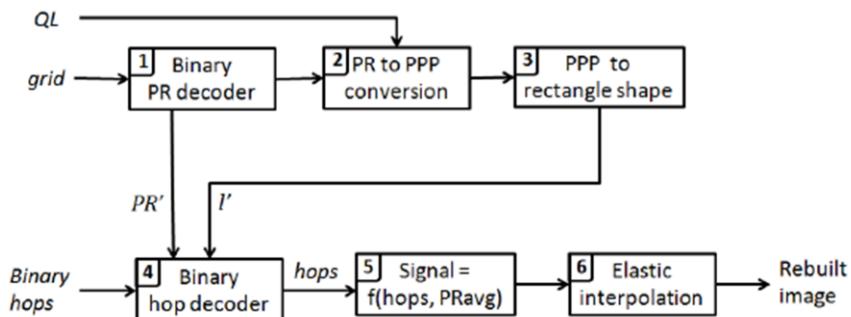


Figura 135 diagrama de bloques del decodificador LHE avanzado

49

El decodificador se compone de 6 pasos. Los 3 primeros son obvios y es aplicación directa de las formulas inversas a las aplicadas en el codificador. Estos 3 primeros pasos tienen un coste computacional despreciable porque trabajan con datos de malla y no con pixeles. La misión de estos 3 bloques es decodificar los datos de las métricas de malla y teniendo en cuenta el valor de QL traducirlos a las longitudes de cada uno de los bloques submuestreados.

Esto es necesario para poder colocar debidamente los hops que vamos a leer en el paso 4

En el paso 4 se transforman los símbolos en hops, en el 5 se transforman los hops en luminancias y por último en el paso 6 se realiza la interpolación de los bloques, recuperando la imagen original

Por ello pasamos directamente a explicar el paso 4:

decodificador binario de hops, al que entramos con I_x y I_y y las métricas cuantizadas de PR_x y PR_y

4 | Decodificador LHE: recuperación de la señal

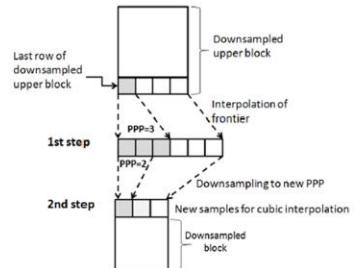
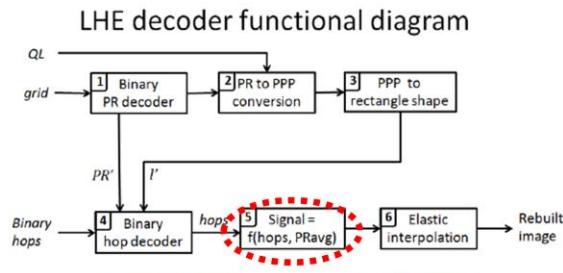


Figura 144. Dependencias entre bloques

Para decodificar el bloque "k" primero debemos interpolar las luminancias de la frontera vertical del bloque "i" y las de la frontera horizontal del bloque "j"

50

En el paso 5 vamos a traducir los hops en luminancia.

los hops son saltos respecto el brillo de fondo de modo que antes de interpretar un hop debemos calcular dicho brillo. Esto significa que hay ciertas dependencias entre pixels como ocurre en LHE basico y tambien hay dependencias entre bloques si usamos fronteras.

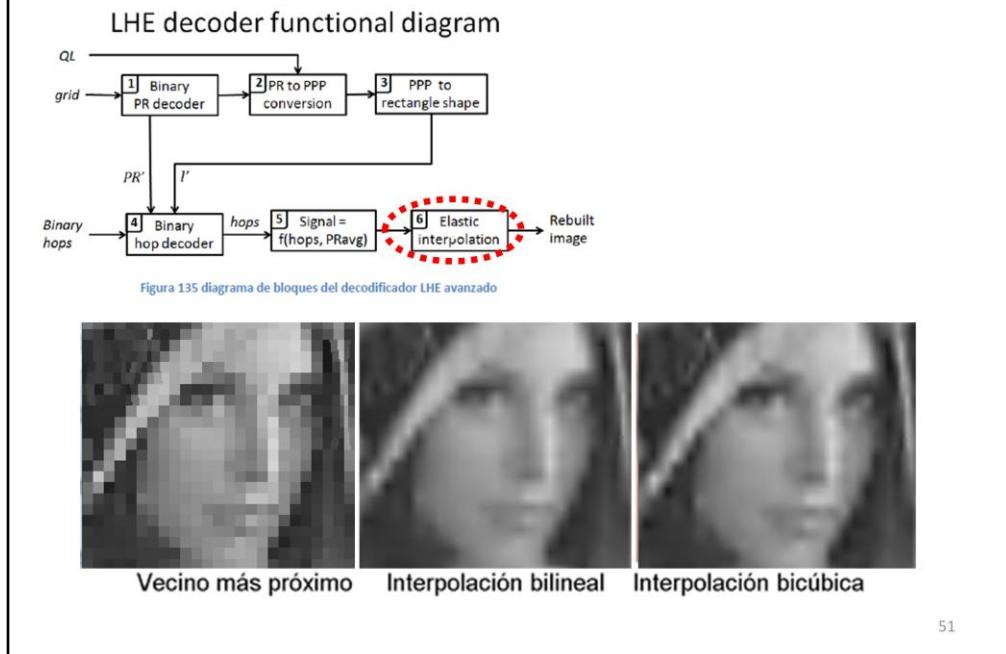
Para comenzar a interpretar un bloque primero debemos interpretar los bloques que limitan sus fronteras superior e izquierda.

Como los bloques submuestreados no solo son de diferente tamaño sino que tienen diferente PPP, debemos primero interpolar y despues submuestrear las fronteras de cada bloque adyacente, ajustando los PPP.

Esto implica que este paso no es 100% paralelizable cuando usamos fronteras ya que no se pueden ejecutar todos los bloques simultaneamente. Se darian $2N-1$ pasos

Si no se usan fronteras, este paso es mas sencillo, simplemente se interpretan los hops sin reescalar los lados ni ajustar PPPs y todos los bloques serán paralelizables

4 | Decodificador LHE: interpolación elástica



Entramos en el paso numero 6 : interpolacion elastica

la interpolación es una operación separable, del mismo modo que lo era el submuestreo

Vamos a contemplar los casos de interpolacion por vecino, bilineal y bicubico .

El caso de interpolacion por vecino cercano es el mas sencillo y solo utiliza un pixel submuestreado para calcular cada pixel interpolado

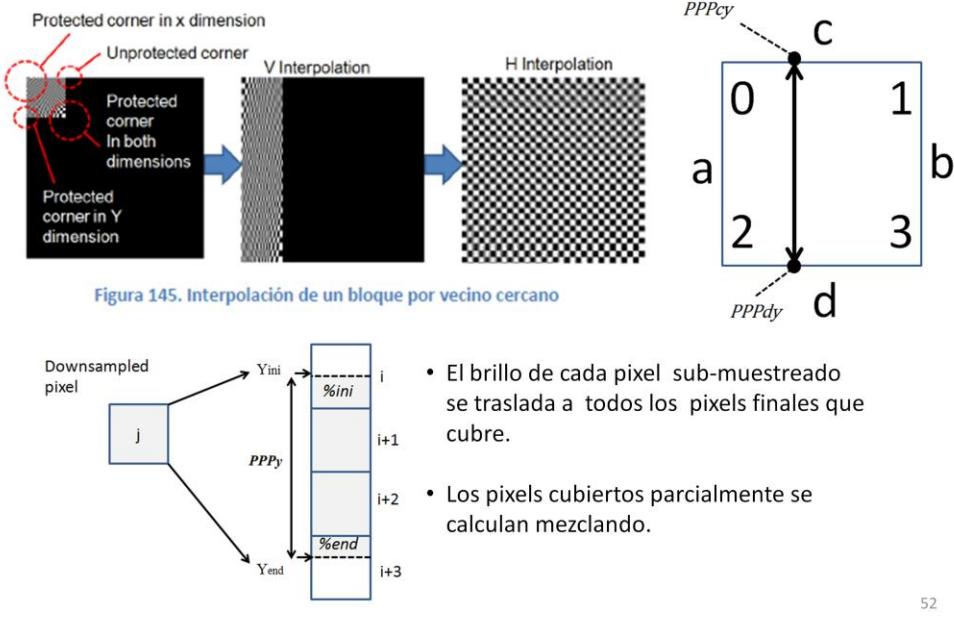
Por el contrario, los otros dos metodos utilizan mas de un pixel submuestreado para calcular cada pixel interpolado por ello producen resultados de mayor calidad, aunque poseen una problemática común que veremos a continuacion

Aunque en la ppt solo voy a contar los dos primeros pues las dificultades del bicubico se parecen al bilineal.

empezamos por el vecino cercano.

4 | Decodificador LHE: interpolación elástica vecino

Interpolación por vecino cercano (separable)



Vamos a ver la interpolacion por vecino cercano

la interpolación es una operación separable, del mismo modo que lo era el submuestreo. Esto significa que podemos interpolar en la dirección vertical y luego en la horizontal o a la inversa, con lo que los cálculos se simplifican mucho.

Los PPP evolucionan entre esquinas de modo que tendremos que manejar unos gradientes de PPP para saber a cuantos pixels debemos expandir cada pixel original

Tras realizar la interpolacion el bloque resultante queda totalmente lleno. Esto no es así en los casos bilineal y bicubico y ahora explicaré porque

4 | Decodificador LHE: interpolación elástica bilineal

Interpolación bilineal

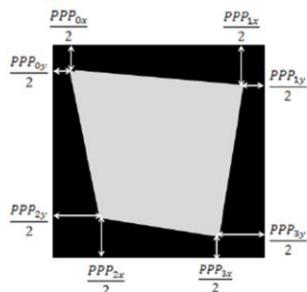


Figura 148. Área pintada por una interpolación bilineal

- Pixeles parcialmente cubiertos. Dos opciones:
- Pintar si se recubre mas del x% (óptimo con 25%)
 - Interpolan su luminancia

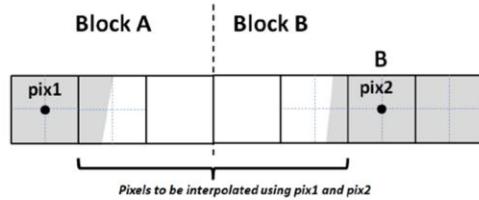


Figura 151. Estrategia de interpolación con pixeles ya interpolados

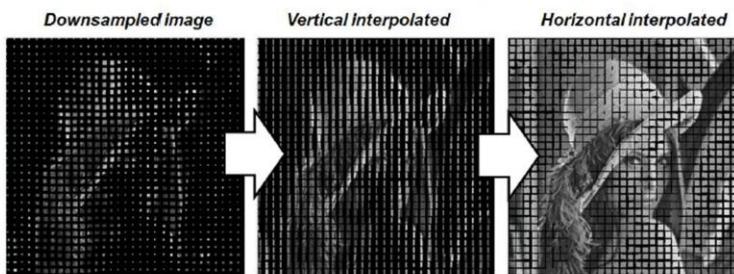


Figura 154. Resultado de la interpolación

53

La interpolacion bilineal debe resolver dos problemas:

- En primer lugar puesto que cada pixel se debe interpolar entre las dos muestras mas cercanas, las costuras de los bloques quedan sin llenar, ya que no podemos empezar a pintar antes de $PPP/2$. El primer pixel abarca PPP pixels de modo que su coordenada central es $PPP/2$. podemos interpolar entre el primer pixel y el segundo pero no entre "nada" y el primer pixel.
- En segundo lugar, los laterales presentan inclinaciones y muchos píxeles no estan totalmente cubiertos. Queda la decision de pintar o no en funcion de cuanto esta recubierto un pixel . Pues bien los mejores resultados se obtienen pintando con al menos un 25% de recubrimiento

4 | Decodificador LHE: interpolación elástica bilineal

Interpolación bilineal de costuras

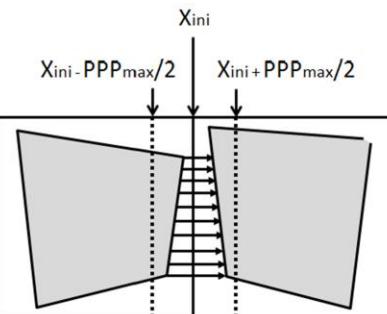
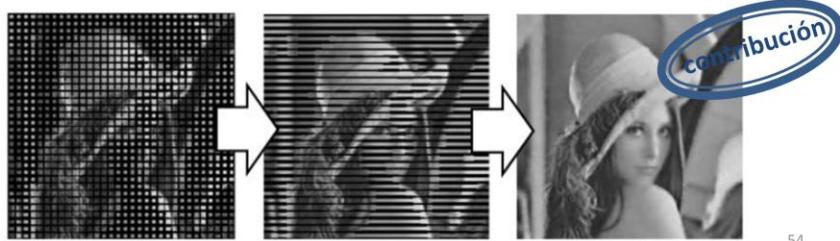


Figura 157. Interpolación horizontal de costuras



54

La interpolacion de costuras entre bloques es un proceso delicado porque los bloques **no son paralelos, ni estan alineados**, la interpolacion de costuras requiere hacerse en dos pasos, primero en horizontal , en la zona comun y despues en vertical

Ademas, la vertical debe contemplar los huecos que a veces quedan

4 | Decodificador LHE: complejidad y parallelización

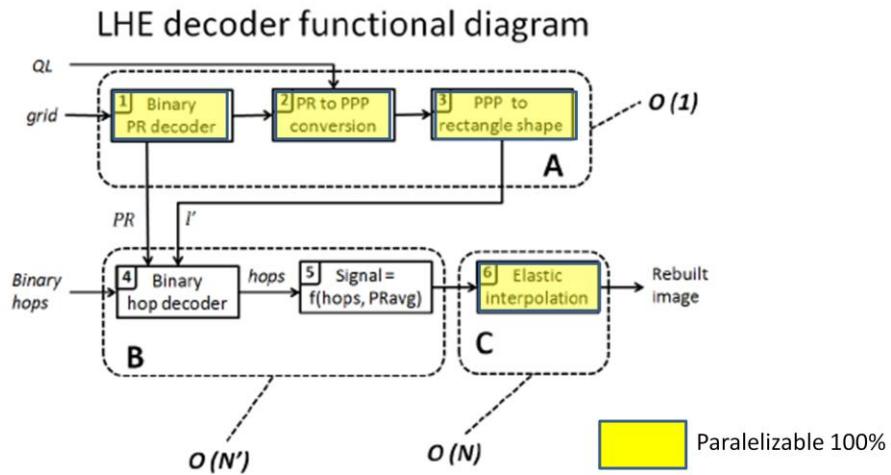


Figura 166. Macrobloques de LHE avanzado en función de su complejidad algorítmica

55

La complejidad de los 3 primeros pasos del decodificador es 1, debido a que son operaciones sobre malla

El macrobloque B que integra los pasos 4 y 5 son de orden N' siendo N' el numero de muestras que por lo tanto es inferior a N y el bloque de interpolacion eslatica tiene complejidad lineal N

En cuanto a la paralelización si no se usan fronteras todos los bloques son paralelizables pero usando las serían 100% paralelizables los que aparecen en amarillo

N' puede ser muy inferior a N . la complejidad es estrictamente N pero esto reduce mucho el numero de operaciones

0| Introducción

1| Estado del arte

2| LHE básico

3| LHE avanzado

4| Decodificador LHE

5| Video LHE

6| Resultados

7| Conclusiones y trabajo futuro

56

Pasamos al video LHE.

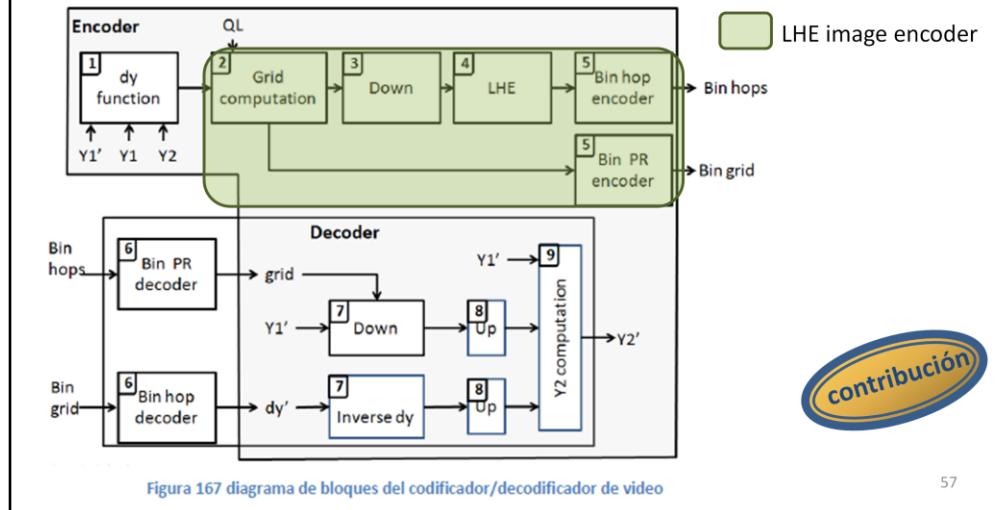
El objetivo del video es diseñar un codificador capaz de comprimir usando información diferencial temporal, es decir, en lugar de codificar cada fotograma por separado se va a aprovechar la redundancia temporal y se va a codificar logarítmicamente las diferencias de cada fotograma con el anterior.

Esto es lo que podríamos llamar un frame P, aunque solo voy a hacer referencia al frame inmediatamente anterior mientras que un frame P por definición puede hacer referencia a uno o más frames anteriores. El cálculo de vectores es una parte fundamental de un codificador de video, sin embargo en este codificador solo voy a dar el primer paso, es decir, compresión por LHE de la información diferencial temporal aunque si diré la estrategia que puede usar LHE para calcular los vectores de forma instantánea, lo cual se suele llevar 2/3 del tiempo total de cálculo en los compresores de video convencionales. Esto podría ser un gran avance

5 | Video LHE

- Retos
- Los errores de cuantización de LHE
 - El downsampling elástico y variable de un frame a otro.

VideoEncoder/decoder functional block



57

En este diagrama está representado el codificador y el decodificador de video. solo voy a explicar lo nuevo. Por ejemplo todos los bloques en verde son identicos al codificador LHE de imágenes fijas

El primer bloque del codificador de video es precisamente el cálculo de la información diferencial a la que he llamado función "dy". Tiene en cuenta el nuevo fotograma cuya luminancia he llamado Y2 y el anterior decodificado, al que he llamado Y1', siendo el verdadero fotograma anterior Y1. Una vez que se calcula esta diferencia, codifica con LHE y se transmite o se almacena

El decodificador es similar al decodificador de LHE pero modificado para trabajar con info diferencial

Los dos problemas que debemos resolver son (solo enumerar y decir que los voy a explicar después)

• Primero Los errores de cuantización de LHE : a continuación explicaré porque esto es un problema

• y en segundo lugar El downsampling elástico y variable de un frame a otro.: hemos de sumar información diferencial que nos va a llegar con una resolución diferente a la de nuestro fotograma. Pensemos que cada bloque se codifica con más o menos muestras según su relevancia y en cada fotograma el contenido de los bloques va cambiando, la resolución está cambiando.

• cosas que en principio tienen distinta resolución y eso es un problema

5 | Video LHE: error de cuantización



Figura 170. Valor de la función dy



contribución

VideoEncoder/decoder functional block

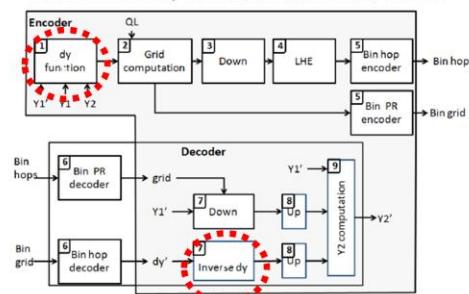


Figura 167 diagrama de bloques del codificador/decodificador de video

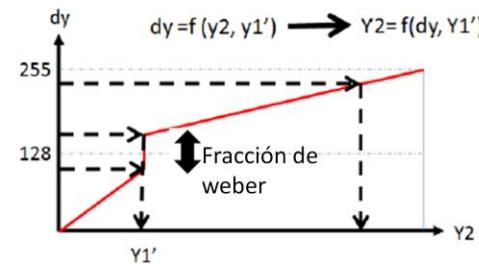


Figura 174. La función inversa de dy

En estas dos imágenes aparece el videojuego de mario y a la derecha la informacion diferencial temporal

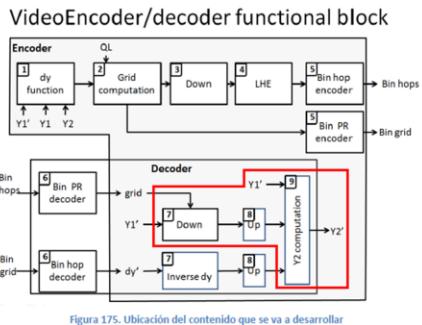
En el videojuego de mario lo que no se mueve desaparece, como por ejemplo los marcadores de manera que la informacion diferencial temporal nos va a permitir ahorrar bits. La informacion diferencial puede ser positiva o negativa de modo que situaremos el cero en el valor 128 que es un tono gris medio.

El error minimo de LHE es el tamaño de su hop mas pequeño h_1 , que es lo que llamamos fraccion de weber. Como h_1 minimo es 4, se produce un error de ± 2 sobre el valor cero. Si contemplasemos ese error como incremento o disminucion de brillo, seria un error pues en verdad no esta ocurriendo. Y nuestra imagen sin movimiento podria incrementar su brillo indefinidamente hasta llegar a ser completamente blanca

La funcion dy permite “absorber” los errores de codificacion de LHE mediante un tramo vertical en el eje horizontal tenemos valor de la luminancia del fotograma nuevo Y_2 , y en el lugar donde $Y_2 =$ al fotograma anterior o Y_1' , situamos el tramo vertical de la funcion. Si no hay movimiento, la funcion dy puede tomar cualquier valor entre $128+2$ y $128-2$, es decir, hay que codificar un 128 pero podemos hacerlo con error y no pasa nada

Si tras una secuencia de mas de 3 fotogramas la imagen permanece inmovil, aunque haya diferencia entre la luminancia del nuevo fotograma y el fotograma decodificado, dejaremos de transmitir dy , porque lo que esta ocurriendo en ese caso es que la informacion diferencial que estamos enviando se comprime tanto que no permite afinar suficientemente el valor al que queremos llegar y en ese caso no merece la pena seguir enviando el diferencial

5 | Video LHE: downsampling elástico y variable



Diagnóstico y solución al problema de las “estelas”

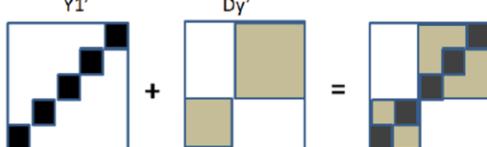


Figura 176. Suma de señales de distinto nivel de sub-muestreo

contribución

Reglas para sumar información diferencial temporal

No sumar imágenes de diferente resolución

$$Y2' = Y1'' + Dy, \text{ siendo } Y1'' = UP(DOWN(Y1'))$$

Preservar la información sin apenas cambios (movimiento o brillo)

$$\text{if } Dy < \text{umbral} (=1), \text{ then } Y2'' = Y1'$$

Ante pequeños cambios de brillo sumar de forma ponderada

$$\text{if } Dy < \text{umbral}', \text{ then } Y2' = f * (Y1'' + Dy') + (1-f) * Y1' \text{ siendo } f = (Dy' - u1) / (u2 - u1)$$

59

Ahora vamos a ver el segundo de los problemas que tiene que resolver el video

Tenemos que tener en cuenta que la imagen diferencial no es lo que llega al decoder. Lo que llega es una versión que además de tener errores de cuantización logarítmica, está submuestreada. Y por ello antes de sumarla hay que interpolarla. Es lo que he llamado Dy' . Supongamos que la imagen diferencial posee diferente (=menor) resolución al fotograma previo, se verán restos del primero, creando un efecto de estelas en el movimiento de objetos.

Las reglas para sumar información diferencial se resumen en esta tabla

- 1) No sumar imágenes de diferente resolución: antes de sumar se deben ajustar las resoluciones de la información diferencial y la del fotograma anterior. Por eso hay un bloque down en el decoder, para ajustar las resoluciones
- 2) Preservar info sin cambios: si no hay movimiento o es casi nulo, no sumo
- 3) Ante pequeños cambios sumar de forma ponderada el resultado de la suma la información diferencial y el fotograma anterior, para no correr el riesgo de perder detalles por utilizar con demasiado peso la información diferencial, que puede tener menos resolución y por tanto menos detalle que el fotograma previo

5 | Video LHE: complejidad y paralelización

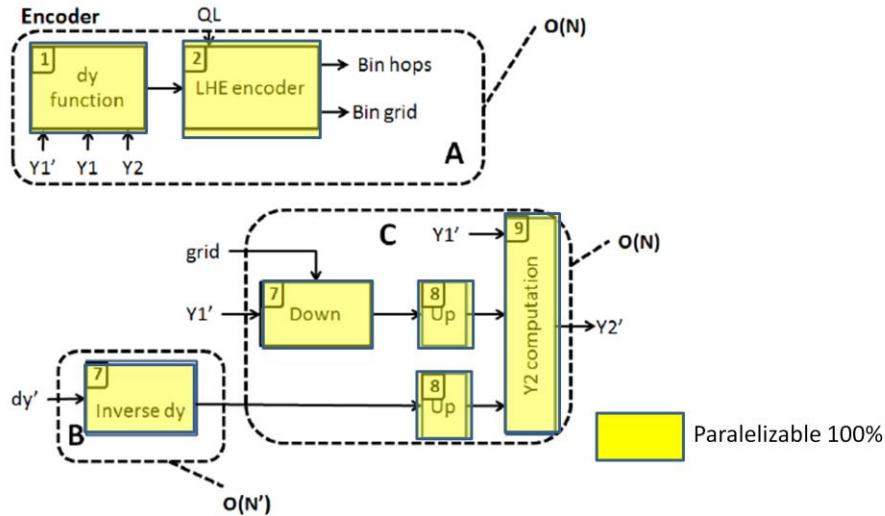


Figura 177. Macrobloques del codificador de video LHE en función de su complejidad algorítmica

60

En esta figura tenemos el codificador de video con los bloques etiquetados según su complejidad computacional.

En la parte del encoder se compone de dos pasos lineales

En decodificación tenemos un primer paso de orden n' donde se interpreta el diferencial siendo N' el número de muestras a las que se ha comprimido. Y una segunda parte de orden n .

N' es muy inferior a N pues es información diferencial, y en caso de video quieto es máxima compresión LHE

En cuanto a la paralelización, todo es 100% paralelizable incluido el codificador LHE pues en video Interesa un LHE sin fronteras, para máxima paralelización



0| Introducción

1| Estado del arte

2| LHE básico

3| LHE avanzado

4| Decodificador LHE

5| Video LHE

6 | Resultados

7| Conclusiones y trabajo futuro

6 | Resultados

Kodak

24 imágenes



 USC University of
Southern California

Signal and Image Processing Institute
Ming Hsieh Department of Electrical Engineering

44 imágenes



Dos de las galerías de imágenes más reconocidas para esta labor son:

- **Kodak Lossless True Color Image Suite** [52]: 24 imágenes fotográficas con diferentes motivos de tamaños 768x512 y 512x768
- **USC-SIPI Image Database** [53]: consiste en 4 galerías de imágenes entre las que se encuentra la galería "mísceleña", que contiene las famosas imágenes de "lena", "baboon", "peppers", etc. Las resoluciones de las imágenes son variadas, 256x256, 512x512 y 1024x1024

Como método de medida de la calidad he escogido dos: el PSNR (Peak Signal to Noise Ratio) y el SSIM. De estas dos medidas la más conocida es PSNR si bien SSIM presenta una mayor similitud con el HSV⁶² (Human Vision System) [42]

Como conclusiones puedo decir que:

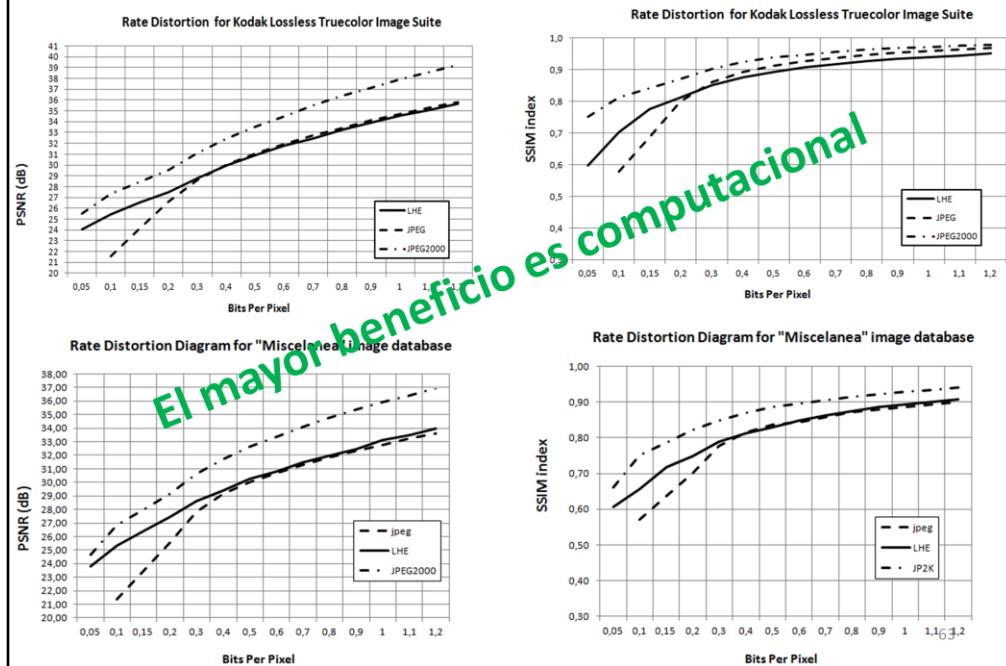
-LHE proporciona una calidad intermedia entre JPEG y JPEG2000 mientras que a medios y altos bit-rates es similar en calidad a JPEG.

-Añadida a esta ventaja se encuentra su velocidad, superior a JPEG y JPEG2000 debido a su menor complejidad.

-Otra interesante característica de LHE es la posibilidad de alcanzar tasas de bit de 0.05bpp, alcanzables por JPEG2000, pero no por JPEG.

- Por último, tras analizar visualmente las diferencias con JPEG2000, se aprecia como los bordes en LHE están más difuminados que en JPEG2000, debido a los efectos del downsampling y de la interpolación. Es esperable que LHE pueda mejorar en este aspecto mediante **downsampling por mediana y mecanismos avanzados de interpolación**, mediante los que se puedan reconstruir bordes abruptos incluso con fuertes compresiones, con estrategias similares a los algoritmos "HQX" o "XBR" [55] aunque deberían ser adaptados o reformulados para poder actuar de forma "elástica".

6 | Resultados: imagen fija



Estas curvas representan el diagrama de distorsión de LHE, comprado con JPEG y JP2k, usando PSNR y de SSIM

Las conclusiones son

- LHE proporciona una calidad intermedia entre JPEG y JPEG2000 mientras que a medios y altos bit-rates es similar en calidad a JPEG.
- Otra interesante característica de LHE es la posibilidad de alcanzar tasas de bit de 0.05bpp, alcanzables por JPEG2000, pero no por JPEG.
- el mayor beneficio es computacional:comprime mas rapido que JPEG y JPEG2000 debido a su menor complejidad.

- Por último, tras analizar visualmente las diferencias con JPEG2000, se aprecia como los bordes en LHE están más difuminados que en JPEG2000, debido a los efectos del downsampling y de la interpolación. Es esperable que LHE pueda mejorar en este aspecto mediante **downsampling por mediana y mecanismos avanzados de interpolación**, mediante los que se puedan reconstruir bordes abruptos incluso con fuertes compresiones, con estrategias similares a los algoritmos “HQX” o “XBR” [55] aunque deberían ser adaptados o reformulados para poder actuar de forma “elástica”.

6 | Resultados: imagen fija

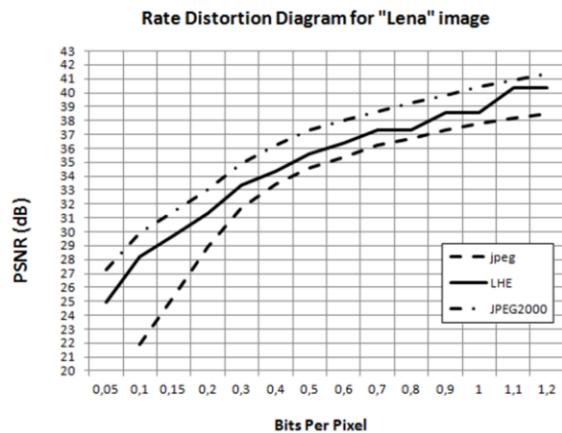
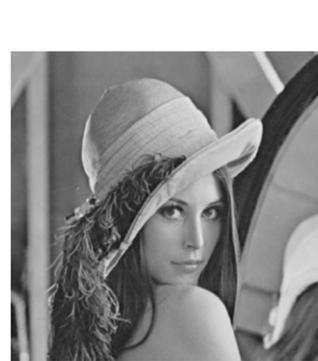
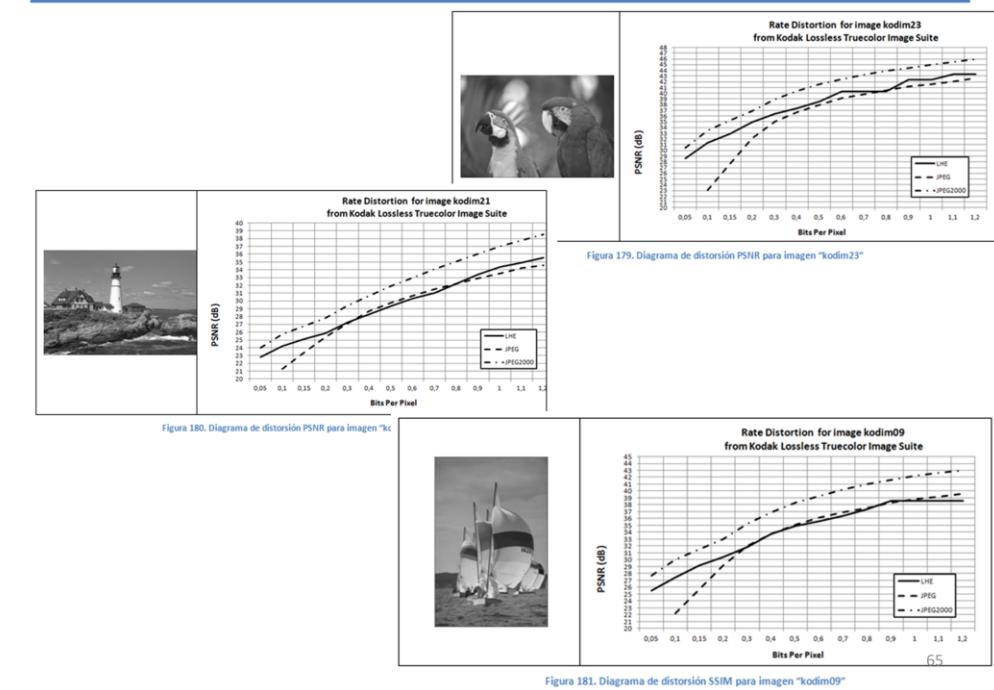
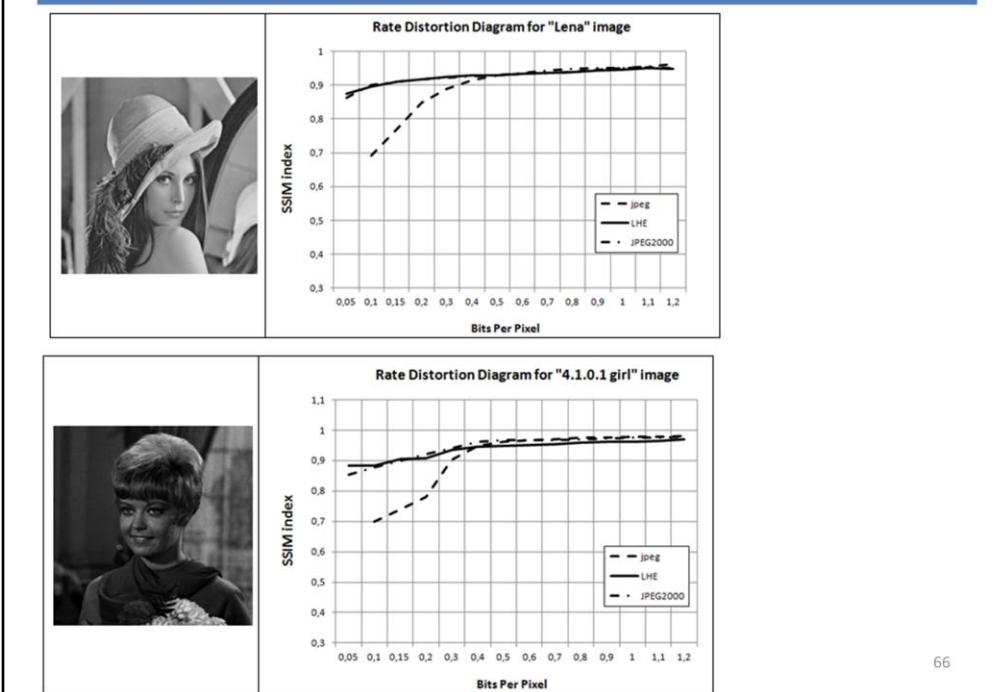


Figura 189. Diagrama de distorsión PSNR para la imagen “Lena”

6 | Resultados: imagen fija



6 | Resultados: imagen fija



6 | Resultados: imagen fija



Lena @ 0.1bpp JPEG, LHE, JP2K



JPEG @ 0.1 bpp



LHE @ 0.1 bpp



JP2K @ 0.1 bpp

67

LHE y JPEG2000 son comparables. Puede ser mejor JPEG2000 aunque son calidades comparables. JPEG esta muy lejos de esta comparacion

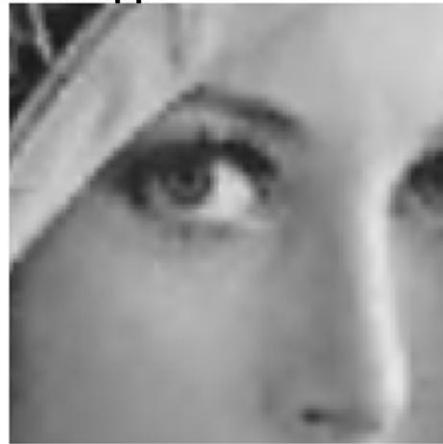
- Por último, tras analizar visualmente las diferencias con JPEG2000, se aprecia como los bordes en LHE están más difuminados que en JPEG2000, debido a los efectos del downsampling y de la interpolación. Es esperable que LHE pueda mejorar en este aspecto mediante **downsampling por mediana y mecanismos avanzados de interpolación**, mediante los que se puedan reconstruir bordes abruptos incluso con fuertes compresiones, con estrategias similares a los algoritmos “HQX” o “XBR” [55] aunque deberían ser adaptados o reformulados para poder actuar de forma “elástica”.

6 | Resultados: imagen fija

JPEG y LHE a 0.3bpp



hay efecto de bloques en JPEG porque hay saltos de cuantización de las componentes de las distintas frecuencias entre bloques



No hay efecto de bloques en LHE porque los bloques comparten los PPP en las esquinas y no hay saltos de resolución

No hace falta deblocking filter como se hace en h264 intraframe

68

La ausencia de bloques vamos a percibirla muy bien en los videos

Los bloques comparten los PPP en las esquinas eso significa que en las adyacencias entre bloques no hay saltos de resolucion y por lo tanto no hay efecto de bloques. En JPEG se producen debido a los cambios en la cuantizacion sufrida por las distintas componentes de frecuencia, los cuales son inevitables a menos que se eliminen posteriormente usando un filtro de deblocking que logicamente tiene un coste computacional adicional y ademas aunque subjetivamente pueda mejorar en realidad empeora el PSNR

6 | Resultados: imagen fija



El modelo de color YUV y en concreto la aplicación del YUV422 y YUV420 esta validado usando la primera version de LHE que utilizaba un downsampling mas primitivo y rígido, no elástico. Estos experimentos han validado el mecanismo de submuestrear aun mas intensamente las componentes de crominancia.

La aplicación de color basado en el modelo YUV ha sido validado de este modo, y dicha validez es extensible al downsampling elástico pues se basa en el mismo principio: más muestras en las zonas con más relevancia perceptual, aunque sea un mecanismo de downsampling más avanzado

No hace falta calcular las metricas de relevancia para la crominancia de modo que el 1er LHE y las metricas solo se calculan en luminancia

6 | Resultados: imagen fija



70

6 | Resultados: video



UNIVERSITÄT
MANNHEIM

DEPARTMENT OF COMPUTER SCIENCE

15 Videos licencia creative commons



Una efecto general observable y mejorable son los **bordes difuminados** de LHE, sobre todo a bajos bit-rates, debidos a los algoritmos de downsampling e interpolación utilizados. Es esperable que con el uso de la mediana en downsampling y algoritmos “pixel-art” en interpolación se puedan conseguir bordes bien definidos a cualquier ratio de compresión

Otra característica interesante es el **paralelismo de la tendencia** de la curva de distorsión PSNR entre h264 y LHE. Son paralelas, lo cual es prometedor. Esto no siempre ocurre entre diferentes algoritmos, como es el caso de las curvas de PSNR de JPEG y JPEG2000

Cuanto mas se saca partido de los vectores, mas se aleja h264

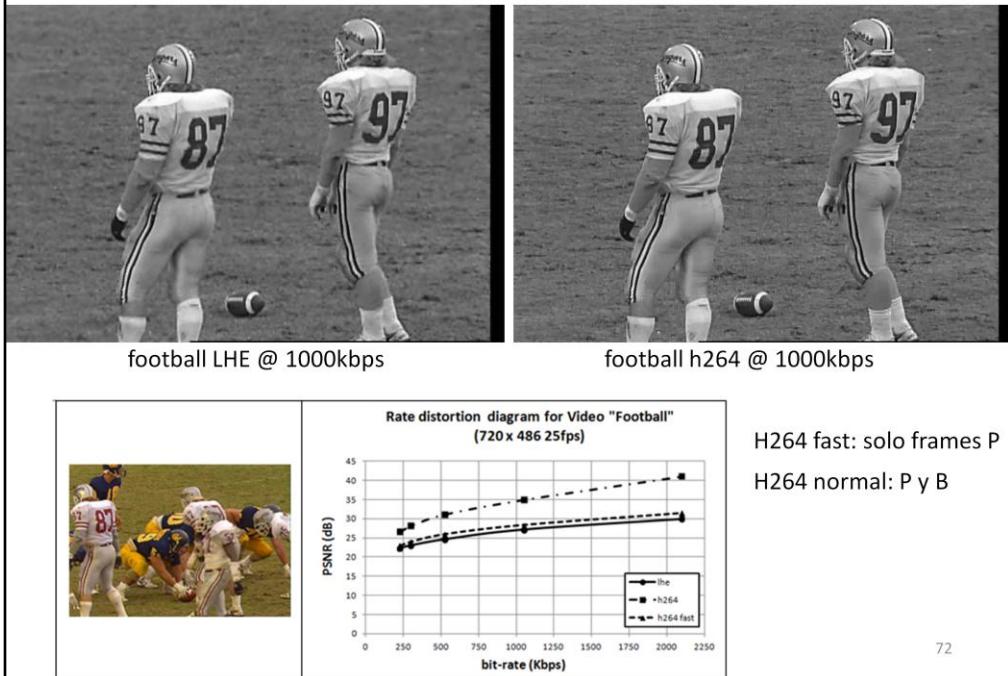
71

Videos con licencia creative commons. 15 videos para testear codecs de video

Vamos a comparar LHE con H264 normal y un H264 al que le hemos quitado la posibilidad de utilizar frames bidireccionales, que es el que emplearíamos en una aplicación de cloud gaming

Tengamos en cuenta que no estamos comparando dos codificadores en igualdad de condiciones. Uno usa compensación de movimiento y el otro no. No estamos comparando peras con peras, pero aun así el experimento es interesante

6 | Resultados: video



Cosas destacables:

Tendencia de la curva idéntica

Muy cerca de h264 fast y con mejoras previstas que lo acercarán aun más sin lugar a dudas

H264 fast: solo frames P: esta seria la configuracion para cloud gaming pues no se pueden usar frames B

H264 normal: P y B

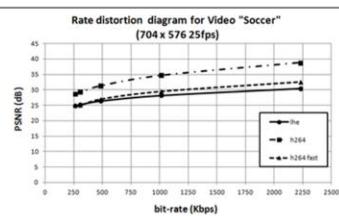
Algo mas borroso, debido al downsampling por media y no por mediana. Tambien por interpolación bilineal

6 | Resultados: video



Soccer LHE @ 930kbps

Soccer h264 @ 930kbps



H264 fast: solo frames P

H264 normal: P y B

73

Cosas destacables:

Tendencia de la curva idéntica

Muy cerca de h264 fast y con mejoras previstas que lo acercarán aun más sin lugar a dudas

Algo mas borroso, debido al downsampling por media y no por mediana. Tambien por interpolación bilineal



0| Introducción

1| Estado del arte

2| LHE básico

3| LHE avanzado

4| Decodificador LHE

5| Video LHE

6| Resultados

7 | Conclusiones y trabajo futuro

7 | Conclusiones y trabajo futuro

Conclusiones:

LHE BASICO

LHE AVANZADO

VIDEO LHE

- Es posible comprimir en el **dominio del espacio** reduciendo la complejidad algorítmica, mediante un modelado fisiológico del ojo
- La **información sensorial** es equiparable a hops logarítmicos, siendo 9 el número óptimo
- La **relevancia perceptual** se puede medir mediante el análisis de los hops de un modo independiente de la resolución de la imagen
- Es posible **traducir analíticamente la métrica de PR a valores de PPP** que permitan un sub-muestreado elástico, permitiendo así que la **métrica sea el criterio** de asignación de PPP.
- **El sub-muestreado elástico es superior en calidad** al sub-muestreado convencional
- El sub-muestreado elástico funciona de forma óptima con un **número fijo (malla) de bloques**
- La calidad de dos imágenes puede ser considerada **equivalente** cuando los mismos valores de relevancia perceptual son traducidos a los mismos PPP (parámetro “QL” de LHE)
- LHE permite aplicar el **modelo de color YUV**
- LHE **no presenta efecto de bloques** aunque procese la imagen dividiéndola en bloques debido a que los bloques comparten valores similares de PPP en sus esquinas
- LHE produce **bordes difuminados** a bajos bit-rates
- LHE permite **compresiones de hasta 0.05 bpp**
- LHE es **apto para comprimir información diferencial temporal**, base del codificador de video.
- La compresión de video LHE (sin compensación de movimiento) genera **diagramas de distorsión paralelos a H264** y muy cercanos a h264 sin frames B

75

7 | Conclusiones y trabajo futuro

Diseminación (artículos)

Revista: IET Image Processing.

Título “Logarithmical hopping encoding: a low computational complexity algorithm for image compression”.

Autores: J. J. Garcia Aranda, M. Gonzalez, M. Cao, J. Navarro y F. Gonzalez.

Accepted on 10th December 2014

Published: Volume 9, Issue 8, August 2015, p. 643 – 651

DOI: [10.1049/iet-ipr.2014.0421](https://doi.org/10.1049/iet-ipr.2014.0421), Print ISSN 1751-9659, Online ISSN 1751-9667

<http://digital-library.theiet.org/content/journals/10.1049/iet-ipr.2014.0421>

Published



Revista: IET Image Processing.

Título “Logarithmical Hopping Encoding: physiological fundamentals”.

Autores: J. J. Garcia Aranda, M. Gonzalez, M. Cao, J. Navarro y F. Gonzalez.

Submitted on 29 August 2015

Submitted



7 | Conclusiones y trabajo futuro

Diseminación (conferencias)

2014	Universidad de Valladolid. Master universitario de investigación en TIC. marzo de 2014 Titulo: "LHE: una nueva tecnología de codificación de imagen y video"
2013	Evento "GameMe5" organizado por HTML5 Spain , Diciembre 2013 (lugar campus UPM Vallecas). titulo :"LHE: una nueva tecnología de codificación para cloud gaming"
2013	UPM , Escuela superior de telecomunicación , Octubre 2013. Máster Universitario en Ingeniería de Redes y Servicios Telemáticos Programa de Doctorado en Ingeniería de Sistemas Telemáticos Titulo :" Codificación logarítmica: calidad wavelet a precio de DPCM" http://www.dit.upm.es/~doct/SI/2013-2014/2013-10-22-LHE_v003.pdf

7 | Conclusiones y trabajo futuro

Proyectos de I+D realizados con tecnología LHE

Las ideas y contribuciones de esta tesis han sido de gran utilidad en el desarrollo del proyecto de investigación “**VideoXperience**: Mejora Efectiva de la Experiencia de Usuario en la Nueva Era de Servicios Digitales mediante la Provisión de nuevas Tecnologías de Supercompresión en Streaming”.

Este proyecto forma parte del subprograma INNPACTO del Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica 2008-2011, y está financiado por el Ministerio de Ciencia e Innovación, actual Ministerio de Economía y Competitividad

Proyectos en curso de I+D con tecnología LHE

“**VINTAGE**” convocatoria MINETUR-RETOS 2015

“**ARQUEOPTERIX**”. Convocatoria CDTI- CIEN 2015 (7 empresas, 4 OPIs)



Convocatorias a proyectos futuros presentados con tecnología LHE

“**VIRTUOSE**” programa CELTIC, 2015 (ganado sello condicional)  **Celtic-Plus**
Smart Connected World

78

Minetur: ministerio de industria energía y turismo

CIEN: convocatoria del CDTI (centro para el desarrollo tecnológico industrial) se financian los 10 mejores proyectos de innovación nacional de todo el arco tecnológico

También ha sido presentado a los premios a la innovación de cincodías “Premio al proyecto empresarial más innovador en el campo de las nuevas tecnologías”

7 | Conclusiones y trabajo futuro

- Mejora de velocidad en el primer LHE: mediante un sub-muestreo por SPS. HD y UHD
- Mejora de la predicción: al estilo PNG
- Evolución del DownSampling Elástico . Malla jerárquica,...
- Mejora del sub-muestreo: experimentar con el uso de la **mediana** en lugar de la media
- Codificación aritmética de los hops:
- Capacidad de transparencias:
- Compresión sin pérdidas: capas de hops
- Mejora de la interpolación: con algoritmos pixel art
- Compensación de movimiento basada en evolución de la métrica de Relevancia Perceptual
- Mejoras de la implementación: introducir el color en el algoritmo actual, implementar la ejecución paralela del algoritmo y hacerlo todo como un plugin para la herramienta ffmpeg
- Difusión: impulsar el algoritmo en organismos de estandarización como W3C y creacion de proyecto github
- Audio LHE

contribución



79

7 | Conclusiones y trabajo futuro: contribuciones

Contribución original principal	Contribución derivada o secundaria	Descripción
Algoritmo basado en el modelado del ojo humano	Modelado del brillo medio: Predicción del color de fondo	Se utiliza una técnica original de predicción.
	Modelado de la ley de weber	Mecanismo de distribución de hops en el rango disponible basado en la ley de weber y cálculo de la razón geométrica que relaciona los hops.
	Número óptimo de hops	9 en total. Hop nulo, 4 positivos y 4 negativos.
	Modelado del "Umbral de detección"	Basado en la fracción de weber y afecta al menor hop (H1)
Métrica de Relevancia Perceptual	Modelado de la "acomodación al brillo medio local"	Mecanismo de compactación y expansión de hops
	Fórmula de la métrica basada en la no intercalabilidad de la información sensorial	Permite medir la relevancia de la información sensorial para la percepción del sujeto
	Umbral de saturación	Saturación sensorial y aplicación para proteger bordes frente a ruido
	Expansión de histograma de PR	Su aplicación en la métrica PR es una contribución original
Downsampling elástico	Cuantización geométrica	Demostable por la relación inversa entre PR y PPP
	Concepto de downsampling elástico	Constituye una contribución original, e introduce el concepto PPP.
	Transformación de PR a PPP	Relación entre PR, PPP y el nivel de calidad QL
	malla de bloques sin tamaño fijo	Sin tamaño fijo. Relaciona tanto la métrica PR como el downsampling elástico para su aplicación óptima
Arquitectura del codificador y decodificador de imágenes	Nivel de calidad QL y calidad equivalente	Otorga significado físico al nivel de compresión deseado QL
	Restricción rectangular	Permite la separabilidad de la operación de downsampling así como la posibilidad de ejecutar un segundo LHE
	Primer LHE reducido por SPS	Otorga más velocidad en alta resolución y los resultados de la métrica de PR son iguales.
	Mecanismo de fronteras para la codificación por LHE	Para codificar por LHE los bloques sub-muestreados
Arquitectura del codificador y decodificador de vídeo	Codificadores binarios de malla y hops	Son codificadores de entropía basados en redundancia espacial y distribución estadísticas
	Interpolación de costuras	Soluciones para las interpolaciones bilineal y bicúbica
	Función diferencial temporal por tramos	Absorbe los errores de cuantización de LHE
	Suma de información diferencial temporal de diferente resolución	Solución mediante 3 reglas
	Solución a la compensación de movimiento mediante el análisis de evolución de valores de la métrica de PR de la información diferencial temporal	Permitirá el cálculo instantáneo de vectores de movimiento basados en LHE