

“Transmisión progresiva de imágenes, basada en *wavelets*, para reducir el uso de ancho de banda en aplicaciones compartidas”

Pablo Roncagliolo B.
pronca@atmlab.utfsm.cl

Agustín González V.
agv@elo.utfsm.cl

Departamento de Electrónica, Universidad Técnica Federico Santa María
Casilla 110-V, Valparaíso Chile. Tel: (56-32) 654196, Fax: (56-32) 797469

Resumen: Una estrategia para implementar sistemas de colaboración distribuida ha sido la transmisión de la interfaz gráfica de la aplicación compartida. En este ámbito, para lograr una transmisión eficiente de secuencias de imágenes, se desarrolló un *codec* especializado, que permite el envío progresivo de diferentes capas de resolución. El *codec* desarrollado se basa en la transformada *wavelets* para implementar algoritmos de transmisión progresiva de imágenes. El método propuesto permite una mayor ‘usabilidad’, pues actualiza con mayor rapidez las ventanas de las aplicaciones compartidas y reduce el ancho de banda requerido en la transmisión de ventanas que presentan cambios constantes. El sistema responde adecuadamente a algunos de los vacíos que existen en la transmisión eficiente de ventanas, que ni los algoritmos de compresión de imágenes, ni los *codec* de video abordan directamente.

Palabras claves: Compresión, Procesamiento de Imágenes, *codec*, Wavelets.

1. Introducción

En el ámbito de los sistemas de colaboración distribuidos, se han desarrollado diversas herramientas tendientes a facilitar la interacción de múltiples usuarios, como son las videoconferencias, la transmisión de contenido, la sincronización de eventos y datos, y la ejecución de aplicaciones compartidas, entre otras. En particular, la implementación de sistemas para compartir aplicaciones, ha seguido dos estrategias fundamentales; la primera, consiste en la sincronización de dos aplicaciones remotas, mediante la captura e intercambio de los eventos generados por el usuario; la segunda, se basa en la captura y transmisión de la interfaz gráfica de una aplicación compartida. En el contexto de esta segunda estrategia, es fundamental una transmisión eficiente de las secuencias de imágenes generadas por la aplicación, pues constituye uno de los factores principales en el uso de ancho de banda de estos sistemas. Las diferentes estrategias para reducir estos costos han determinado la aparición de diversos estándares en las etapas de compresión, codificación y transmisión, de imágenes y video. Por ejemplo, los últimos desarrollos en el ámbito de la compresión de imágenes, como JPEG2000 [1], lo constituyen como uno de los algoritmos de mejor rendimiento. En el otro extremo, los *codec* de la familia MPEG, son un buen ejemplo del gran desarrollo que han logrado los sistemas de codificación de video. Pero existe un tipo de aplicación particular que no está cubierto ni por los algoritmos de compresión de imágenes ni por los *codec* de video, como son las secuencias de imágenes sintéticas generadas en los sistemas de compartición de aplicaciones.

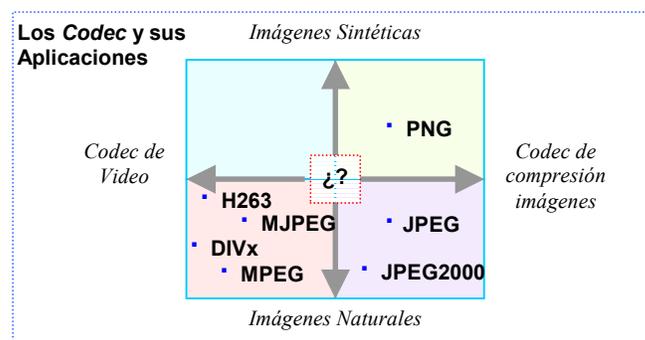


Figura 1: Diagrama de focalización del sistema desarrollado.

Este caso ha sido abordado por numerosos proyectos, que han dado lugar a conocidos software como VNC [2], que permite la ejecución remota de una escritorio compartido, o Netmeeting [3] de Microsoft que también realiza la misma función. Otras aplicaciones comerciales conocidas y de gran rendimiento son las desarrolladas por la empresa Citrix [4]. Sin embargo, tanto la escalabilidad, como la relación entre la compresión-calidad de las imágenes, en muchos de estos sistemas, no está diseñada para aplicaciones específicas y con restricciones de calidad.

La problemática descrita, se puede representar, en primer lugar, como la intersección de los requerimientos de codificación de video, donde la sensación de movimiento es importante, versus los requerimientos de codificación de imágenes estáticas, donde los detalles de interés para el observador deben ser preservados. En segundo lugar, como la intersección entre los requerimientos de codificación de imágenes naturales (fotos) versus la codificación de imágenes sintéticas (formas geométricas, interfaces de software, etc.). La figura 1 representa gráficamente esta problemática, la cual constituye el foco principal de esta investigación y del sistema propuesto. Un escenario posible de aplicación, es la transmisión de ventanas de un software de imágenes médicas, en un sistema de Teleasistencia; en este caso la interacción de uno o más médicos sobre el software requiere de las ventajas de la transmisión de video y a la vez de la calidad de la codificación de imágenes; y dichas imágenes compuestas pueden contener regiones de tipo sintético (interfaz del software) y regiones de tipo natural (imágenes médicas).

Este foco ha sido abordado también, por el proyecto ODUst[5], el cual permite compartir aplicaciones en ambientes heterogéneos y utiliza protocolos de transmisión UDP *multicast*, lo que facilita la escalabilidad en el número de usuarios. Inicialmente ODUst se basó en el estándar JPG, lo que no permite su utilización en aplicaciones que requieran una codificación sin pérdida de las imágenes. En este contexto, el presente trabajo aborda el desarrollo de un nuevo tipo de *codec* para el sistema ODUst, que permite transmitir de manera progresiva las ventanas de la aplicación, aumentando así la usabilidad del sistema, pues transmite con mayor velocidad las ventanas, y mejorar su resolución continuamente, en la medida que el usuario mantenga su atención sobre alguna ventana en particular. De lo contrario, si el usuario cierra una ventana o actúa sobre el software, se reduce el ancho de banda utilizado, pues se abortan las transmisiones que contienen el resto de información para completar la resolución deseada.

Las limitaciones del sistema psicovisual del hombre explican por qué no notamos cambios rápidos y percibimos el movimiento como continuo, cuando se presentan a lo menos 15 cuadros por segundo. Así también no procesamos instantáneamente una imagen. El tiempo en enfocar y fijar la atención en una imagen puede ser utilizado para mejorar progresivamente su resolución. Para lograr este objetivo, en este trabajo se utilizó la transformada *wavelets* de imágenes, que permite separar una imagen en diferentes sub-bandas de resolución, las cuales pueden ser transmitidas en forma independiente, y decodificadas a medida que arriban a sus destinos. Esta característica permitió implementar el sistema de transmisión progresiva que se presenta a continuación.

2. Descripción del sistema

2.1 Fundamentos del algoritmo de compresión implementado

El desarrollo de las técnicas de compresión de imágenes ha desencadenado la formulación de una serie de estándares, entre los que destacan JPEG basado en la codificación de la transformada de coseno, y a partir del año 2000, la consolidación de un nuevo estándar basado en la codificación de la transformada *Wavelets* denominado JPEG2000. Este algoritmo presenta una gran eficiencia, pues permite lograr grandes tasas de compresión con menor distorsión en las imágenes respecto de otros algoritmos. Estas ventajas se observan en la figura 2. donde para un mismo tamaño de archivo, JPEG2000 presenta notables diferencias respecto de JPEG.



Figura 2: imágenes de 512x512 píxeles. a) JPG b) JPEG2000. Ambas imágenes pesan 2,7Kb, sin embargo la imagen b) presentan distorsiones suaves, que son más naturales.

Dado este contexto, el estándar JPEG2000 fue el punto de partida en el desarrollo de esta investigación. Para ello se evaluó la utilización de algunas bibliotecas disponibles del estándar JPEG2000 (ISO/IEC 15444-5:2003), como JJ2000 desarrollada en Java [6]; o Jasper desarrollada en C [6]. Sin embargo, se consideró más apropiado implementar directamente las funciones específicas de la transformada *wavelets*, pues las bibliotecas mencionadas corresponden a implementaciones completas del estándar JPEG2000, e incluyen una serie de funcionalidades que no están directamente relacionadas con el objetivo de este trabajo, por lo cual, generarían un gran *overhead* en el sistema. Finalmente se optó por escribir las rutinas para el manejo progresivo de imágenes mediante *wavelets*, manteniendo la filosofía central del JPEG2000, pero alejándose levemente del estándar para optimizar su rendimiento en la transmisión progresiva, más que en el almacenamiento de imágenes comprimidas.

2.2 Arquitectura del *Codec*

La estructura de esta adaptación de JPEG2000 contempló las etapas fundamentales del estándar, descartando las etapas que constituyen refinamientos orientados a maximizar la compresión, pero que en el contexto de las transmisión secuencial de imágenes en la red, constituyen retardos innecesarios en el procesamiento. La figura 3 presenta las etapas del *codec* implementado.

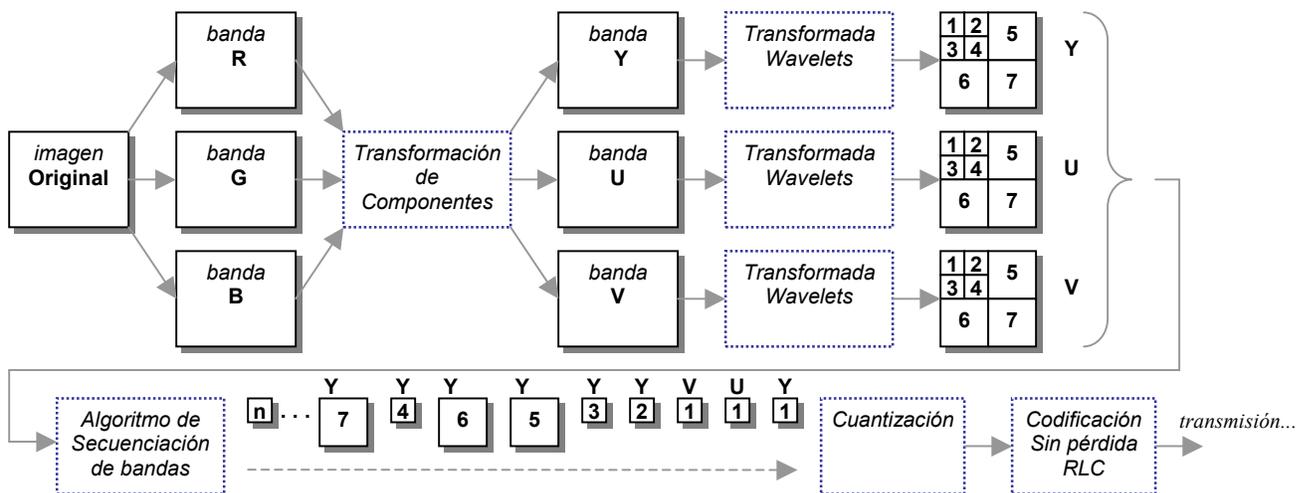


Figura 3: arquitectura del *codec* desarrollado.

Todas las imágenes o *tiles* procesados por el *codec* siguen la misma secuencia de operaciones. Inicialmente la imagen en color es separada en sus bandas RGB, sobre las cuales se aplica una transformación que permite agrupar la información según la codificación YUV. A partir de estas nuevas bandas, se obtiene la descomposición *Wavelets* de dos niveles, generando así 7 sub-bandas de información por cada componente. Posteriormente se aplica un algoritmo de secuenciación de las sub-bandas según diferentes criterios. Por último, antes de la transmisión, se realizan dos procesos, la cuantización que es el factor principal de “compresión con pérdida” de información, y la aplicación del algoritmo *Run Length Code* [7], que no introduce pérdida de información. Las principales etapas del *codec* se detallan a continuación.

ETAPA TRANSFORMACIÓN DE COMPONENTES: La transformación de componentes permite agrupar la misma información contenida en las bandas RGB en otros planos o “espacio de colores”, tales que, representen de mejor manera la forma en que el sistema visual humano decodifica las imágenes. Esta transformación logra que la información de la intensidad esté reflejada principalmente en la banda Y, y la información de color y saturación en las bandas V y U. Por lo tanto, la simple transmisión de la banda Y permite integrar gran parte de la información de contornos, bordes y gradientes, la cual puede ser completada posteriormente con la información de color. Esta propiedad es utilizada por el sistema de transmisión progresiva desarrollado. La Tabla 1 presenta la transformación utilizada

TABLA 1	
Transformación de Componentes	Transformación Inversa de Componentes
$Y = \text{floor}((R+2G+B)/4)$	$G = Y - \text{floor}((U+V)/4)$
$V = R-G$	$R = V+G$
$U = B-G$	$B = U+G$
* $\text{floor}(x)$ es el mayor entero que no excede a x	

Considerando que el sistema visual humano tiene mayor sensibilidad a los cambios de intensidad que a los cambios de color o saturación, las bandas V y U puedan ser cuantizadas con menor resolución. Esta propiedad es muy utilizada en los codec de video e incluso el clásico algoritmo JPG. Sin embargo, en la codificación basada en *Wavelets*, no se hacen submuestreos fijos de las bandas V y U, como es el caso de JPG y muchos otros *codec*, permitiendo su procesamiento según los requerimientos específicos de la aplicación.

ETAPA TRANSFORMADA WAVELETS: La implementación más común de la transformada *Wavelets* en el procesamiento de imágenes [9], consiste en la aplicación recursiva de filtros pasa altos y filtros pasa bajos, de manera alternada, sobre las columnas y filas de la imagen, respectivamente. Este proceso se presenta en la figura 4.

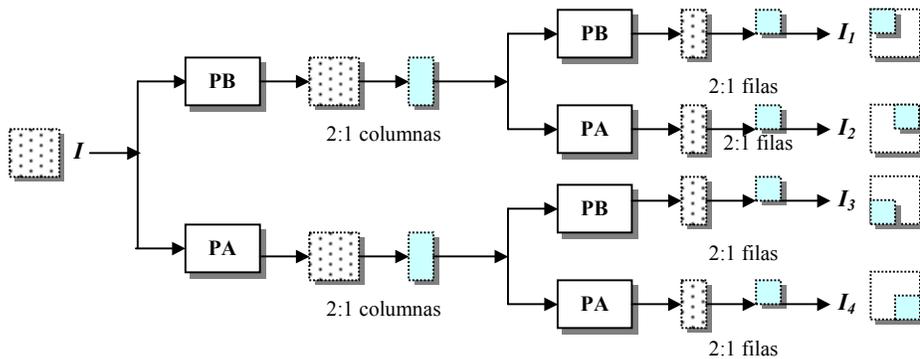


Figura 4: Multiresolución o descomposición *Wavelets*. Los filtros “pasa bajos” PB y “pasa altos” PA se aplican sobre filas y columnas de la imagen de manera alternada.

La imagen original I es descompuesta en cuatro sub-bandas de un cuarto de resolución, lo que se denomina el “primer nivel de descomposición”. Los algoritmos de compresión aplican de manera recursiva este proceso sobre la sub-banda I1, es decir la banda de bajas frecuencias, pues contiene la mayor cantidad de información.

Desde el punto de vista de la compresión no se ha ganado nada aún. En vez de la imagen inicial, se obtienen 4 sub-bandas de $1/4$ de resolución. Lo interesante en la compresión basada en *wavelets*, es que el enfoque se centra ahora en analizar las características de estas sub-bandas finales. En ellas se distinguen claramente los detalles y fondos de la imagen original. Así por ejemplo, una imagen que tiene grandes bloques de color homogéneo debería obtener muchos valores cercanos a cero en las sub-bandas de detalles (I2,I3,I4). Por otro lado, en la sub-banda de baja frecuencia (I1), se debería observar claramente los valores promedios de fondo de los bloques de color homogéneo.

La compresión basada en *wavelets*, no consiste en la simple eliminación de las sub-bandas de alta frecuencia, sino en un proceso de cuantización variable de éstas. Para la mayoría de las imágenes (naturales y por sobre todo sintéticas), muchos de los valores de las sub-bandas de alta frecuencia pueden ser muy pequeños y parecidos unos de otros, por lo que pueden ser redondeados o aproximados. Esta nueva cuantización produce que muchos de los valores sean cero o sean iguales unos de otros, lo que aumenta la eficiencia de la etapa posterior de compresión sin pérdida.

El número de aplicaciones recursivas de la transformada *Wavelets* o número de “niveles”, queda determinado por el grado de compresión que se desea lograr y por los recursos de procesamiento que se dispongan para su ejecución. La figura 5 presenta un proceso de multiresolución de tres niveles. El sistema desarrollado en esta investigación utiliza una descomposición de dos niveles, pues el objetivo del codec no es la razón de compresión, sino la transmisión progresiva de imágenes. De esta manera se logran 7 sub-bandas como se observa en la figura 6.

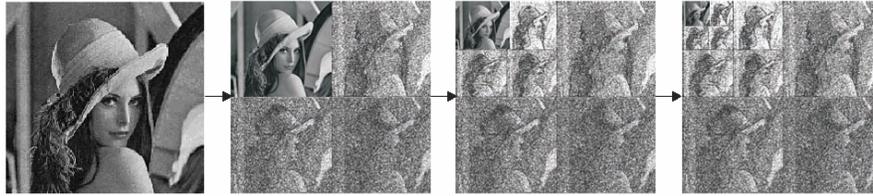


Figura 5: Descomposición *wavelets* en 3 niveles de la imagen clásica “lena”.

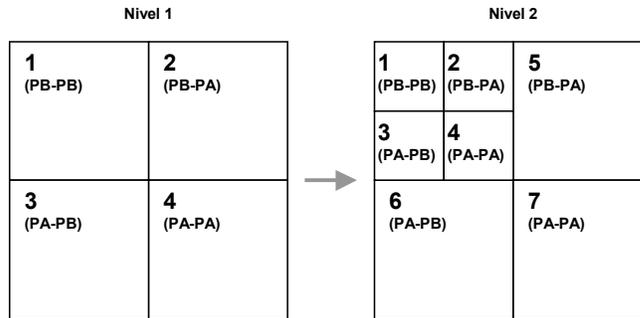


Figura 6: Aplicación de 2 niveles de descomposición *wavelet*. PB indica la aplicación de filtro pasa bajos; PA, filtro pasa altos

Respecto de los filtros utilizados, el estándar JPEG2000 [1] se basa en la utilización de los coeficientes de filtros desarrollados por Daubechies, conocidos como 9/7, para la transformada irreversible. Para la transformada reversible JPEG2000 propone los filtros Le Gall 5/3. Sin embargo, el estándar también define un método simple de filtrado denominado “*lifting*”, el cual consume muy pocos recursos de procesamiento, respecto de la implementación de funciones de convolución para los otros filtros de Daubechies. Dado que el objetivo del sistema desarrollado es transmitir imágenes en “tiempo real”, se privilegió la implementación del método *lifting*. El filtrado basado en *lifting* se aplica a filas y columnas de la imagen, representadas por el vector x en las ecuaciones siguientes:

$$y(2n+1) = x(2n+1) - \text{floor}\left(\frac{x(2n) + x(2n+2)}{2}\right)$$

$$y(2n) = x(2n) + \text{floor}\left(\frac{y(2n-1) + y(2n+1) + 2}{4}\right)$$

ETAPA DE SECUENCIACIÓN DE SUB-BANDAS: La etapa de secuenciación aquí descrita, no es parte del estándar JPEG2000 y es un aporte de este trabajo. Específicamente se aprovechó esta flexibilidad para buscar secuencias más convenientes de transmitir progresivamente las imágenes. Este módulo corresponde simplemente a una etapa de ordenamiento de la secuencia en la cual serán transmitidas las diferentes sub-bandas, basándose en criterios como: i) priorización de las sub-bandas que aportan un mayor factor en la SNR; ii) priorización de las sub-bandas que requieren menor ancho de banda; iii) priorización según el usuario. Para estandarizar el primer tipo de priorización se realizó un estudio para diferentes tipos de imágenes y se obtuvo un vector de ordenamiento básico de las sub-bandas según el aporte de información que contienen. El segundo tipo de priorización corresponde a un ordenamiento por tamaño de las sub-bandas codificadas. Finalmente, un tercer tipo de priorización en la transmisión, puede ser definida por el usuario. Ejemplos de los diferentes tipo de ordenamiento y los resultados asociados se presentan en la sección 4.2 de este trabajo.

Esta funcionalidad es muy simple, pero puede tener un impacto notable en el uso de ancho de banda. Por ejemplo, un usuario puede elegir priorizar la transmisión de las bandas intensidad por sobre las de color. Si la aplicación compartida presenta cambios constantes en sus ventanas, en muchos casos, sólo se alcanzará a transmitir las primeras sub-bandas de los *tiles*. La transmisión de las restantes sub-bandas se aborta si se detecta un cambio en el tile, por lo tanto es muy importante la elección de cuales deben ser las primeras sub-bandas en la transmisión. Este criterio depende del tipo de aplicación, por lo cual este módulo de secuenciación del *codec*, puede adaptarse a los requerimientos mencionados en el párrafo anterior.

Es importante notar, que en los tres casos, la simple estabilización de las ventanas en la aplicación compartida genera la posibilidad que el sistema complete la progresión, reconstruyendo íntegramente toda la información de la imagen original, en los tres tipos de priorización. Esta “estabilización” de las imágenes corresponde simplemente a un proceso manual, donde el usuario evita gatillar cambios en la aplicación compartida. Esto permite un uso eficiente del ancho de banda, pues se aborta la transmisión de muchas sub-bandas de detalles, que no tiene sentido transmitir las, dado el nivel de cambios en las ventanas de la aplicación, en un instante en particular.

ETAPA DE CUANTIZACIÓN: Si bien, la Transformada de *Wavelets*, es la base de este método de compresión, no es ésta la que realiza la compresión en sí, sino que facilita o permite que un proceso posterior, en este caso la Cuantización, logre realizar una compresión eficiente y con un grado de distorsión de la imagen controlado y en concordancia con el modo de percepción visual del hombre.

Las sub-bandas de detalles, que genera la transformada *Wavelets*, generalmente poseen valores muy pequeños, y para imágenes sintéticas o imágenes sin muchos cambios, numerosos “píxeles” de estas sub-bandas son cero o cercanos a cero. Entonces, un primer análisis permite notar que las sub-bandas de detalles pueden ser eliminadas o degradadas, sin afectar notablemente la calidad de la imagen. La cuantización es simplemente una nueva escala para los píxeles de las diferentes sub-imágenes generadas.

A diferencia de JPEG2000, el codec desarrollado en este trabajo, incorpora una nueva optimización en la etapa de cuantización. La aplicación de la transformada *Wavelets* genera valores específicos en las sub-bandas mayores y menores al rango dinámico original de las imágenes. Para las imágenes originales cuyos elementos se representan con 1 byte, la transformación aplicada obliga a utilizar un tipo de variable mayor a 1 byte para su almacenamiento, lo cual repercute en duplicar el tamaño ocupado por las sub-bandas. Para evitar este aumento en el uso de ancho de banda y minimizar la distorsión que pueda generar una recuantización, se implementó el siguiente algoritmo para el modo de operación “con pérdida”: i) se define un umbral de corte en cada sub-banda. ii) todos los valores menores a ese umbral se hacen cero. iii) se resta el umbral de corte de la sub-banda. iv) se aplica el proceso de cuantización de 0 a 255 v) se transmite el umbral de corte antes de transmitir la sub-banda. De esta manera se mantiene el tamaño de las sub-bandas en la transmisión y se minimiza la distorsión generada por la cuantización.

ETAPA DE CODIFICACIÓN SIN PÉRDIDA: El sistema de codificación sin pérdida utilizado, también difiere del estándar JPEG2000. Se privilegió un algoritmo que utiliza pocos recursos de procesamiento y memoria. Aunque existen otros algoritmos de mejor razón de compresión, un algoritmo de buen desempeño para los requerimientos planteados es el de Codificación por Longitud de Cadenas o RLC (*Run length Code*) [7] que consiste en transmitir el número de valores iguales adyacentes. La ventaja de este método de codificación, es que no requiere preprocesar los datos como por ejemplo el método de Huffman [10]. El método RLC se puede implementar directamente en la función de salida del codec, sin introducir mayores costos o retardos en la transmisión.

2.3 El *codec* como parte del sistema ODUst

El sistema ODUst [5] considera la división de la imagen total de la ventana de la aplicación, en pequeñas sub-imágenes denominadas *tiles*. Inicialmente todos los *tiles* deben ser transmitidos a los clientes del sistema. Esta fase de ‘puesta en marcha’ del sistema consume muchos recursos, tanto en el uso de capacidad de procesamiento en la codificación de cada uno de los *tiles*, como el uso de ancho de banda en la transmisión de los *tiles* por la red. Cuando un *tile* ha sido transmitido, sólo se deberá reenviar si éste ha cambiado. Para ello se deben ejecutar periódicamente algunas funciones para verificar qué zonas de la pantalla han cambiado y deben ser retransmitidas. En la práctica, se implementa un vector de chequeo que indica cuáles *tiles* han cambiado y por lo tanto serán nuevamente procesados (comprimidos) y transmitidos. No obstante, dado que ODUst contempla la transmisión UDP, existe un segundo periodo de muestreo que permite reenviar *tiles* independientemente de los cambios detectados, para soportar de esta manera la posible pérdida de paquetes UDP.

Inicialmente ODUst [8] utilizó la compresión JPEG, o PNG, sin embargo, dada las nuevas características y la mayor eficiencia de la compresión basada en *wavelets*, se consideró su incorporación, para obtener así las ventajas comparativas respecto de JPG, principalmente en la posibilidad de transmisión progresiva, la codificación de regiones de interés (ROI), y la mayor razón de compresión.

3. Características y ventajas del sistema

Se presentan a continuación algunas de las principales características del sistema desarrollado, que desde el punto de vista cualitativo, permiten clarificar las ventajas del sistema implementado respecto de los sistemas tradicionales basados en el envío de *tiles* de tipo JPG.

MEJOR RAZÓN DE COMPRESIÓN: Se logra una mejor calidad de imagen en comparación con otros estándares, sobre todo al comparar imágenes con alto grado de compresión (i.e. bajo los 0.25 bpp). En otras palabras, es posible lograr una mejor razón de bit por píxel para un mismo nivel de calidad de imagen. Sin embargo, el *codec* implementado permite codificar imágenes bajo métodos de compresión con o sin pérdida. Existen muchas aplicaciones donde es necesario poseer ambos tipos de codificación.

TRANSMISIÓN PROGRESIVA POR RESOLUCIÓN Y/O TAMAÑO: En muchas aplicaciones, no es necesario transmitir más información de la que se puede desplegar. En ese sentido, el *codec* desarrollado ofrece la capacidad de “extraer” de manera eficiente, la información de la imagen original para un tamaño o resolución específica. Así, por ejemplo un dispositivo móvil y un computador de escritorio pueden acceder la misma imagen, según las propiedades gráficas específicas de cada equipo. Por otro lado, el *codec* ofrece la posibilidad de transmitir la imagen en diferentes grados de resolución, donde cada transmisión sólo considera el envío de nueva información. En la figura 7 se presentan las dos posibilidades de progresión: la primera basada en disminución de la distorsión; la segunda, basada en aumento de resolución.



Figura 7: a) Escalabilidad en distorsión (SNR). b) Escalabilidad en resolución.

PROCESAMIENTO Y ACCESO DIRECTO A SUB IMÁGENES Y REGIONES DE INTERÉS (ROI): Esta característica indica la capacidad de acceder a una porción específica de una imagen y extraerla bajo diferentes resoluciones o tamaños. Las regiones de interés, permiten que alguna parte específica de la imagen pueda ser transmitida con mejor resolución o menos distorsión que el resto de la imagen.

MEJOR RESPUESTA A GRANDES CAMBIOS EN LA IMAGEN: Los cambios en grandes bloques de las ventanas en las aplicaciones compartidas, son muy comunes y por múltiples razones: los desplazamientos o *scroll*; el despliegue de una nueva ventana de opciones o un diálogo; una nueva aplicación; un cambio de página; el despliegue de animaciones o videos; un retorno de línea al final de un procesador de texto, entre muchos otros ejemplos puntuales. La transmisión progresiva minimiza estos efectos, pues acelera la previsualización de las ventanas, aumentando así la sensación de cambio y movimiento.

MEJOR RESPUESTA A CAMBIOS DE ALTA FRECUENCIA DE *TILES* ESPECÍFICOS: Otro problema indeseado, es el cambio en zonas puntuales de la ventana con una alta frecuencia. Esto genera un efecto molesto, por ejemplo, al tratar de seguir el cambio en un reloj, la actualización fluida de un texto que se está escribiendo, parpadeos o efectos en animaciones flash, gif, juegos, etc. En todos estos casos ocurre que la frecuencia de cambio en el origen es mayor a la frecuencia de actualización del *tile* en el destino. La transmisión progresiva minimiza estos efectos, por el tiempo de respuesta de las primeras actualizaciones.

FUSIÓN DE IMÁGENES CONSECUTIVAS DURANTE PROGRESIÓN: Un efecto indirecto que se logra con el sistema de transmisión progresiva de imágenes, es la fusión que se genera entre *tiles* sucesivos. Esto resulta muy interesante, pues muchos de los cambios, cuando una aplicación está activa, son muy breves y por lo general retornan a un estado inicial. Por ejemplo, al presionar un botón, o al abrir y cerrar un menú o una ventana de mensajes, etc. En todos estos casos, ni el método tradicional, ni el método progresivo lograrán actualizar las imágenes completamente. Por el contrario, si el cambio es muy rápido sólo se logra transmitir una sub-banda de baja resolución. El receptor actualiza dicha sub-banda, pero el resto de las sub-bandas del *tile* corresponden al *tile* anterior. Por lo tanto, cuando se vuelve

al estado anterior, sólo basta transmitir la sub-banda que logró ser alterada, el resto aun se mantienen en el receptor. La actualización de este tipo de “eventos” se logra a una mayor velocidad y haciendo un buen uso del ancho de banda. Este efecto se puede visualizar en la secuencia de imágenes presentada en la figura 8.

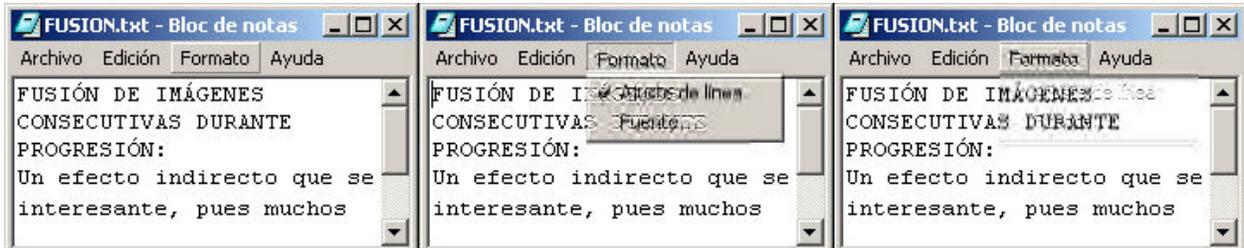


Figura 8: Secuencia: fusión de imágenes

Esta secuencia refleja un evento muy rápido en el cual un usuario desplaza el mouse sobre un menú, sin la intención de mantenerlo abierto. Sin la transmisión progresiva y la fusión de imágenes que se observa, el texto bajo el menú habría desaparecido y se obliga al sistema a transmitirlo nuevamente. En la figura 8, se observa como los *tiles* del sector conservan sub-bandas “antiguas” y “nuevas”, por lo tanto la retransmisión sólo debe hacerse en las sub-bandas que lograron ser alteradas, lo que minimiza las retransmisiones inútiles, generadas por eventos tan comunes, como el de este ejemplo.

4. Resultados

Las diferentes características presentadas en el punto anterior, se ven reflejadas desde el punto de vista cuantitativo en los resultados que se presentan a continuación.

4.1 Ancho de Banda Requerido

Se definió el siguiente protocolo de evaluación: se comparó el rendimiento del sistema desarrollado respecto del sistema de transmisión de *tile* JPG; se implementaron ambos métodos en el sistema ODUst; se definió una misma aplicación a compartir, en este caso un navegador con una página Web; finalmente se determinó el número de *tiles* actualizados por ambos sistemas, para un total de bytes previamente definidos.

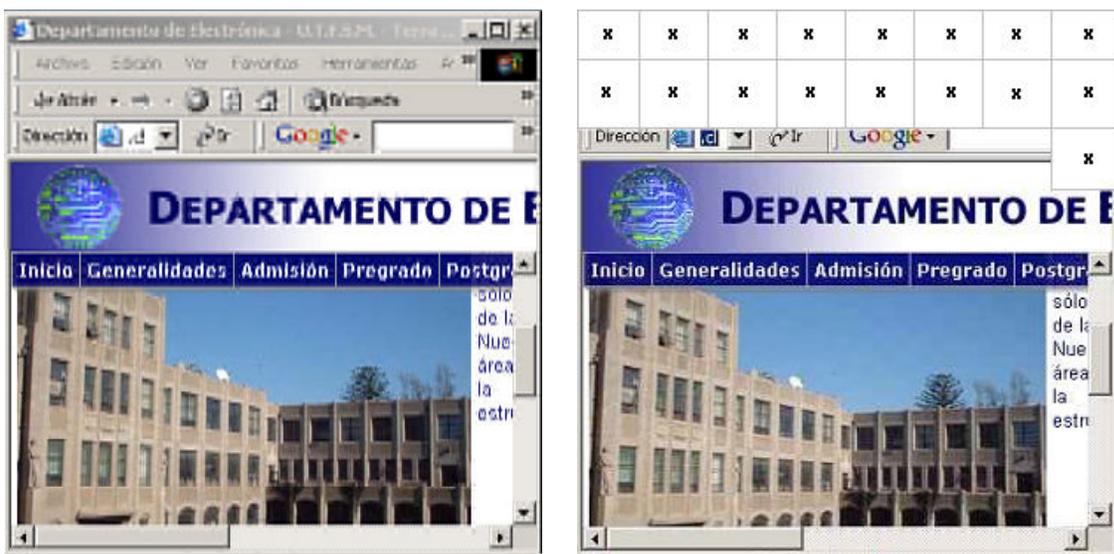
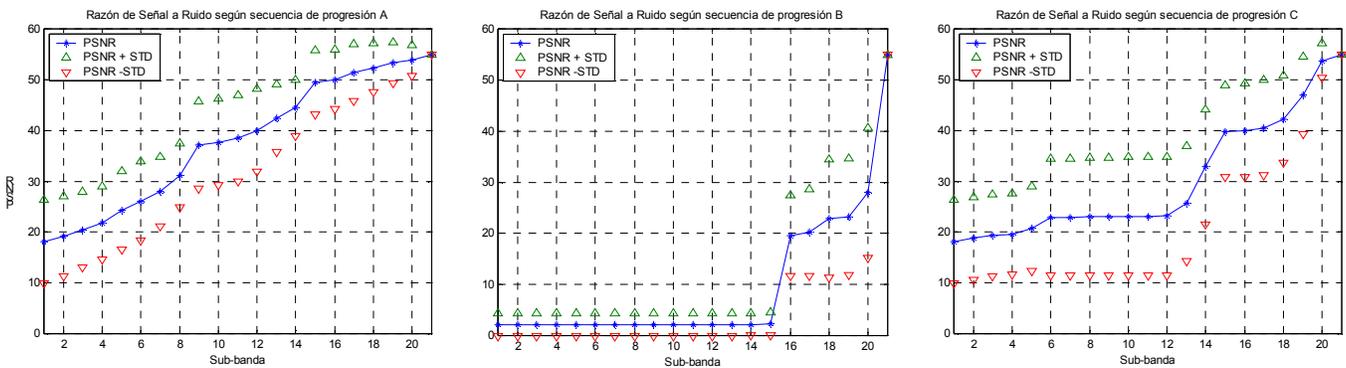


Figura 9: Ambas imágenes son capturas del sistema justo en el instante cuando se completan 32Kbyte de transmiten a) sistema con *tiles* progresivos b) sistema con *tiles* JPG, los *tiles* marcados con ‘x’ no alcanzaron a ser transmitidos.

Esta evaluación, refleja el mejor rendimiento cualitativo del sistema, pues permite que el usuario reciba una mayor cantidad de *tiles*, utilizando la misma cantidad de byte transmitidos. En la figura 9.a se observa la imagen del sistema progresivo, con las primeras sub-bandas transmitidas. La figura 9.b muestra que el número de tiles JPG que alcanzaron a ser transmitidos es menor. Ambas imágenes representan 32Kbyte de transmisión. Si bien la calidad de los tiles JPG presentan mejor resolución, esto es sólo momentáneamente, pues el sistema progresivo mejora la calidad posteriormente.

4.2 Criterios de Secuenciación de Bandas

Se evaluaron tres criterios para secuenciación de las sub-bandas: i) se determinó una secuencia de progresión A, que optimiza el cociente “aumento de SNR por byte transmitido”. ii) una secuencia B, según criterio de tamaño de las sub-bandas, transmitiendo en primer lugar las de menor tamaño. iii) una secuencia C, definida por el usuario, donde se prioriza el envío de las bandas de intensidad por sobre las bandas de color. En los gráficos se observa la progresión de la razón de señal a ruido para las tres secuencias seleccionadas.



Para determinar las secuencias se realizaron mediciones sobre 96 tiles de 64x64 píxeles, de la imagen capturada de la ventana de un navegador con la página <http://www.elo.utfsm.cl>. Las secuencias resultantes se presentan a continuación:

Secuencia Criterio A
A = [1 6 5 2 3 15 7 4 8 16 17 20 19 21 18 13 10 9 12 14 11]
Ap = [100 26 31 52 40 33 35 66 54 26 28 35 46 32 55 48 77 59 89 61 100]

Secuencia Criterio B
B = [9 10 11 12 13 14 16 17 18 19 21 20 2 4 3 1 8 15 7 6 7]
Bp = [62 70 78 69 65 78 35 51 59 21 22 32 33 33 50 46 45 45 73 65 100]

Secuencia Criterio C
C = [1 2 3 4 15 8 9 10 16 17 11 18 5 6 7 12 13 19 20 21 14]

Los números en los vectores A, B y C corresponden al índice de la sub-banda según la figura 6. La banda Y corresponde el rango 1-7, la banda V el rango 8-15 y la banda U el rango 16-21. Los vectores Ap y Bp presentan el porcentaje de tiles con el cual dicha sub-banda maximizaba el criterio.

Estos resultados permiten definir una secuencia de progresión general para el *codec*. Es decir, el uso de secuencias según el criterio A, favorece un crecimiento continuo de la SNR durante la progresión. El uso de secuencias definidas por el usuario, aunque numéricamente represente un crecimiento menor de la SNR, puede priorizar otros criterios de visualización, por ejemplo, la intensidad por sobre los colores. No obstante, a medida que avanza la progresión obviamente todos los criterios convergen a una SNR idéntica.

5. Trabajos Futuros

En el ámbito de la computación móvil las restricciones de ancho de banda, las heterogeneidad de las capacidades gráficas de los equipos, las nuevas tecnologías de comunicación inalámbrica, y la menor capacidad de procesamiento de los dispositivos, obligan a redefinir muchas de las aplicaciones y sistemas distribuidos que no fueron diseñados para estos requerimientos. Específicamente, los sistemas orientados a compartir las ventanas de una aplicación, deben ser analizados a la luz de estas nuevas restricciones. En este contexto, se considera en trabajos futuros la

implementación de un *middleware* destinado a compartir ventanas de aplicaciones en redes de dispositivos heterogéneos, incluyendo dispositivos móviles, y que considere las siguientes restricciones o requerimientos: i.- el sistema debe permitir compartir las ventanas de una aplicación entre dos o más usuarios; ii.- el sistema debe funcionar sobre sistemas con hardware y software heterogéneos; iii.- el sistema debe utilizar eficientemente el ancho de banda, que en algunos dispositivos puede ser muy bajo; iv.- el sistema debe poseer la capacidad de adaptarse a diferentes resoluciones gráficas de los dispositivos clientes, siendo esta capacidad, transparente para la aplicación compartida; v.- la transmisión debe considerar las restricciones de las comunicaciones inalámbricas, en particular el nivel de pérdida de paquetes, las dificultades de establecer comunicaciones UDP, y las constantes desconexiones y conexiones en los enlaces; vi.- El sistema de codificación de las secuencias de imágenes, debe operar sobre dispositivos con baja capacidad de procesamiento, utilizando algoritmos simples pero eficientes, que permitan su ejecución en “tiempo real”.

6. Conclusiones

La transmisión progresiva de imágenes, es una herramienta efectiva para optimizar el uso de ancho de banda en aplicaciones distribuidas. Permite conjugar en un mismo esquema, la transmisión eficiente de imágenes de menor resolución cuando la aplicación presenta cambios frecuentes, con la transmisión sin pérdida de información cuando en la aplicación compartida disminuyen los cambios. El *codec* desarrollado responde adecuadamente a algunos vacíos existentes, pues la aplicación directa de los *codec* de compresión o los *codec* de video, no solucionan ciertos problemas característicos de la transmisión de ventanas en aplicaciones compartidas. Es por ello que el desarrollo de un *codec* específico, orientado a este tipo de aplicaciones, permite abordar muchos aspectos relevantes como el uso eficiente de ancho de banda, el tipo de compresión con o sin pérdida, la velocidad de actualización de las ventanas, la progresión de información según diferentes criterios y la posibilidad de adaptar el tipo de codificación según el tipo de información que las ventanas contienen en un instante dado.

Agradecimientos: Los autores expresan su agradecimiento al Fondo de Investigación UTFSM, proyecto N°23.04.26.

Referencias

- [1] A. Skodras, C. Christopoulos and T. Ebrahimi, “The JPEG2000 Still Image Compression Standard”, *IEEE Signal Processing Magazine*, pp. 36-58, September 2001.
- [2] T. Richardson, Q. Stafford-Fraser, K. R. Wood & A. Hopper, "Virtual Network Computing", *IEEE Internet Computing*, Vol.2 No.1, Jan/Feb 1998 pp33-38
- [3] Netmeeting, <http://www.microsoft.com/windows/netmeeting/>
- [4] Citrix, <http://www.citrix.com>
- [5] Agustín J. González, Hussein Abdel-Wahab, and J. Christian Wild, "Scalable and Resilient Application Sharing System for Internet Collaboration", *Journal of Internet Technology, Special issue on Distributed Multimedia Information Systems*. ISSN 1607-9264, Volume 2, No. 4, pp. 317-325, October, 2001.
- [6] “JPEG 2000 Part 5: Reference software”, norma ISO/IEC 15444-5:2003.
- [7] RLC, http://en.wikipedia.org/wiki/Run_length_encoding
- [8] Oscar R. Reyes y Agustín J. González, "Mejoras de ODUST: Un Sistema para Compartir Aplicaciones", Jornadas Chilenas de Computación 2004, Artículo 5 del VIII Workshop en Sistemas Distribuidos y Paralelismo 2004, Arica, Nov. 2004.
- [9] A. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, “Image coding using wavelet transform”, *IEEE Trans. on Image Processing*, v1, 1992
- [10] D. Huffman, “A Method for the Construction of Minimum Redundancy Codes”, *Proceedings of IRE*, v.40, N9, p.1098, Sept. 1952.