ROBUST LIGHT FIELD WATERMARKING WITH HIGH SPATIAL AND ANGULAR FIDELITY

Amir Ansari

Vniversitat de València

Doctoral Programme in Physics

SUPERVISORS: Dr. Manuel Martinez-Corral
Dr. Genaro Saavedra-Tortosa

A THESIS
SUBMITTED TO THE FACULTY OF PHYSICS
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

SEPTEMBER 2021

Declaration

We, Drs. Manuel MARTINEZ-CORRAL and Genaro SAAVEDRA-TORTOSA, Full Professors at the Optics and Optometry&Vision Sciences Department of the Universitat de València, hereby

CERTIFY

that this manuscript, *Robust Light Field Watermarking with High Spatial and Angular Fidelity*, is a record of the original research work undertaken, under our supervision, by Mr. Amir ANSARI, for the award of the PhD Degree within the Doctoral Programme in Physics at Universitat de València.

Valencia (Spain), September 1, 2021

Signature:	Signature:
Dr. Manuel Martinez-Corral	Dr. Genaro Saavedra-Tortosa

This dissertation is kindly dedicated to my parents who
have been an endless source of support and courage.

همه اش یکجا تقدیم به پدر عزیزم به استواری کوه و مادر نازنینم
اسطوره مهربانی که همواره در همه عرصه ها مشوق من بوده اند.

“Sometimes I think the surest sign that intelligent life exists elsewhere in the universe is that none of it has tried to contact us.”

- Bill Watterson, cartoonist

Acknowledgements

This work would not have been possible without the support of all the individuals who helped me. I would like to take this opportunity to express my gratitude for the ones who helped me with my research and wish all of them the best.

First and foremost, I am extremely grateful to my supervisors professor Manuel Martinez-Corral and professor Genaro Saavedra-Tortosa for their constant guidance, assistance and support. They showed a great level of helpfulness and enthusiasm by spending a lot of time to answer my numerous questions in detail. Their training in all aspects of being a good research scientist is immensely appreciated and will not be forgotten. Apart from scientific side, I am specially thankful to them on personal level for all their kind support that they provided me in literally every aspect. They are more than PhD supervisors and they never hesitated to offer their kindest support at any moment.

I would also like to extend my deepest gratitude to Adrian for his tremendous kindness specifically his helps upon my arrival at Spain. Many thanks to Jorge for being so nice and supportive. I am indebted to Alex for his continuous help in bureaucratic procedures. I am also grateful to all the other staff of the 3D Imaging and Display lab (3DID) like professor Juan Carlos Barreiro, Emilio, Angel, Gabriele, Min, Anabel, Przemysław, Nico, and Hui. I owe you many thanks for every advice that you provided in the work as well as the friendly atmosphere that you created.

Always and forever, my sincere thanks go to my lovely family. I have an amazing family, unique in many ways, and the stereotype of a perfect

family in many others. I have always received their unconditional love and attention. They caused me to feel that my most valuable wealth is my family. Doing my PhD abroad has been harder for them than me, but their eternal courage and endless desire of my succeed, finally paid it off. I am infinitely thankful to my awesome father, lovely mother and supportive brother who exhibited their highest care and love since ever. Undoubtedly, if it was not for their enormous support and adoration, I could never reach this point. Many thanks for not leaving me alone for one single moment. Vahid jan! One million thanks for being my brother since ever! Having a brother as you is fabulous. Since the happiest days of our childhood, you have always stood next to me. The geographical distance has never made me feel far from you.

پدر و مادر عزیزم سپاس بیکران برای لطفی که همیشه در طول زندگی به من داشته اید و همواره از محبت های بی پایان و حمایت های بی دریغ تان بهره مند بوده ام. از همان روزهای آغازین ورود من به مدرسه، شما خستگی ناپذیر تمام تلاش تان را نمودید تا در بهترین محیط های آموزشی و به بهترین شکل ممکن تحصیل کنم. بدون حمایت های بی دریغ شما و توجه و تلاشی که در تحصیل من داشتید ، هرگز به این نقطه نمی رسیدم. بی شک، حمایت های شما در لخظات سخت و دشوار بسیار باغث دلگرمی و شادی خاطر من بود. کلمات از بیان عمق قدردانی و سپاسگزاری من ناتوانند ولی شما همیشه بدون اینکه من لب از لب بگشایم، خود سخن مرا خوانده اید.

وحید عزیز، سپاس از تو به خاطر همه محبت و پشتیبانیت در تمام طول زندگی. داشتن چنین برادری حقیقتا باعث مباهات من است. همواره بدون هیچ چشمداشتی یار و یاور من بوده ای. حمایت های بی دریغت و همه لطف و مهربانیت با هیچ کلمه ای قابل وصف نیست.

This work was funded from the European Union’s Horizon 2020 Research and Innovation Program under the Marie Skłodowska-Curie grant agreement No 676401, European Training Network on Full Parallax Imaging.

Abstract

The Light Field (LF) has been increasingly used for 3D representation of the scene. The LF incorporates tremendous amount of spatial and angular information which is of paramount usefulness in 3D reconstruction. The high dimensionality of the LF makes it incompatible with most of the current watermarking methods proposed for conventional images. Even if such methods are applicable to the LF data, there is still a huge risk of ruining the spatial or angular information. Given the utmost essence of the watermarking for ownership protection, fingerprinting, authentication and tamper detection, two watermarking methods have been proposed for the LF. The proposed methods thoroughly protect both spatial and angular information of the LF.

The first proposed method employs Discrete Cosine Transform (DCT) and Singular Value Decomposition (SVD) to embed the watermark. The fidelity of the watermarking approach has been verified objectively and subjectively. Both assessments show that the watermarked LF is imperceptible from the host one. If no attack occurs, the embedded watermark can be extracted error-free. The proposed method also shows a high amount of robustness against common image processing methods as Gaussian noise, JPEG compression and median filtering.

The second proposed method gains the enormous correlation of the LF along all dimensions. Exploiting the 4D wavelet transform significantly improves the robustness and capacity of the first method. The inter-channel correlation of the LF pixels has been substantially reduced by conversion of the color space. Using 4D wavelet transform and DCT jointly, results in gaining the advantages of both transforms. To acquire a robust watermarking feature, SVD is used in the second proposed method as well. The objective and subjective measurements approve the high transparency of this method making the watermarked LF indistinguishable from the host one. Additionally, the proposed method also shows great robustness against some image processing attacks as Gaussian noise, JPEG compression, median filtering and JPEG 2000. Principally, this method uses a secret key to embed the watermark in some selected coefficients of the wavelet transform. The location of such coefficients is specified by the secret key. The impact of the block size and number of DCT coefficients used to embed the watermark, have been investigated. Additionally, the contribution of each transform to the robustness of the watermarking method has also been measured to check if it is possible to attain optimal performance without some of the utilized transforms. The simulations demonstrate that the optimal results are obtained only when all the transforms are used and the block size and number of DCT coefficients are chosen properly.

Resumen extendido (Spanish)

El término *ocultación de información* se refiere típicamente a la inserción secreta de datos en una *señal anfitriona*. La señal anfitriona puede ser una imagen, un archivo de audio, un video,... Las técnicas de ocultación de información se dividen generalmente en *marca de agua digital*, *esteganografía* y *criptografía*. Si la propia existencia del mensaje secreto incrustado debe permanecer en secreto, entonces el método de ocultación de información se conoce como esteganografía. Por el contrario, en la marca de agua digital, el usuario es consciente de la existencia del mensaje secreto. A diferencia de la esteganografía y la marca de agua, existe otra categoría de ocultación de información que cifra el mensaje secreto sin insertarlo en una señal anfitriona. Estos métodos se conocen como criptografía en la literatura técnica especializada. Los métodos de ocultación de información se han utilizado durante milenios. A modo de ejemplo, es conocido que Heródoto (485-525 a.c.) ya cita que Histiaeus, el gobernante de Mileto por designación del rey de reyes persa Darío El Grande estaba conspirando para derrocar el imperio persa. Sin embargo, nunca quiso levantar ninguna sospecha entre los que eran leales al rey de reyes ni perder la confianza que el rey Darío había depositado en él. Por ello, para instigar la revuelta Histiaeus afeitó la cabeza de uno de sus esclavos y tatuó un mensaje secreto sobre su cuero cabelludo. Tras dejar crecer el pelo del sirviente, éste viajó sin despertar sospechas hasta el destinatario del mensaje. En la era reciente, la esteganografía se usa ampliamente para la comunicación encubierta. En la esteganografía, la señal anfitriona se usa simplemente para transmitir un mensaje secreto importante. La señal anfitriona no importa por sí misma, pero es de suma importancia no llamar la atención de los expertos en seguridad. La señal anfitriona generalmente se elige entre los medios típicos que no causan sospechas. Es por eso que el mensaje transmitido no está encriptado en esteganografía. En otras palabras, un mensaje cifrado hace sonar inmediatamente las alarmas, pero es menos probable que un mensaje sin cifrar llame la atención. Como ejemplo, se cuenta que en 1966, el comandante en jefe estadounidense Jeremiah Denton se vio obligado a participar en una entrevista televisiva que se transmitió en Estados Unidos. Fingiendo sentirse incómodo con las luces cegadoras de la televisión, parpadeó en código Morse deletreando la palabra ”T-O-R-T-U-R-E”. Al igual que la marca de agua, el rápido crecimiento de la comunicación por Internet ha proporcionado un medio perfecto para que los sistemas de esteganografía transmitan los datos ocultos sin causar sospechas graves.

A diferencia de la esteganografía, los métodos de marca de agua digitales pueden no tener ningún deseo de ocultar la existencia del mensaje incrustado. La marca de agua se define como la inserción imperceptible del mensaje secreto en la señal anfitriona. Esto es exactamente lo contrario de lo que ocurre en la esteganografía, en la que la señal anfitriona no tiene importancia real y se usa simplemente como cobertura. La marca de agua digital se usa ampliamente para la protección de derechos de autor, autenticación, detección/corrección de errores, comunicación encubierta y monitoreo de transmisiones. Se espera que cada plataforma de marca de agua:

- Incruste tanta información como sea posible. El envío de información secreta es el principal motivo de explotación de las técnicas de marca de agua. Esto es especialmente importante en la comunicación encubierta.

- Genere una marca de agua lo más imperceptible posible sobre la señal anfitriona. La diferencia detectable entre la propia anfitriona y la anfitriona tras el marcado anula el propósito de la marca de agua.

- Sea lo más robusto posible contra ataques sobre la señal anfitriona. En el contexto de las marcas de agua, el ataque se refiere a cualquier alteración intencionada o no de los valores de la señal marcada.

Obviamente, la realización perfecta de estas tres características sigue siendo un desafío y, dependiendo de la aplicación, se puede priorizar una o dos de estas características. El rápido crecimiento de la demanda de marcas de agua puede contribuir razonablemente a la creciente preocupación por la protección de los derechos de autor en las últimas décadas. A pesar de las enormes oportunidades que ofrece Internet para compartir la información a gran escala, la duplicación ilegal, la manipulación y el intercambio de información ha aumentado sin descanso. Esto impone serias preocupaciones a los autores y editores que dedican mucho tiempo y esfuerzo a la creación de contenidos. El rápido desarrollo de los métodos de marca de agua fue una respuesta prevista a la implacable tendencia al alza de la piratería. La marca de agua ha desempeñado un papel activo en la protección de los derechos de autor, la detección de manipulaciones, la autenticación y la comunicación encubierta. El número de artículos de investigación publicados sobre marcas de agua muestra la importancia absoluta de las marcas de agua en nuestra era.

Otra categoría de ocultación de información es la criptografía, que se define básicamente como un método para proteger la información y las comunicaciones mediante el uso de códigos, de modo que solo los lectores autorizados pueden decodificar y leer el mensaje. Así, en criptografía el mensaje secreto se implementa sin usar señal de cobertura. La mayoría de los sistemas criptográficos utilizan conceptos matemáticos y un conjunto de cálculos basados en reglas. El contenido se cifra y se proporciona una clave de descifrado solo a los receptores autorizados. El contenido cifrado se transmite a través de Internet, pero los receptores no autorizados difícilmente pueden descifrar el contenido codificado. A diferencia de la marca de agua, el cifrado no tiene ningún control sobre la redistribución del contenido descifrado por parte del usuario autorizado. Puede ser que un cliente compre una clave de descifrado válida y, después del descifrado, redistribuya el contenido de forma masiva. Por lo tanto, la criptografía puede proteger el contenido antes del descifrado, pero una vez descifrado, el contenido no tiene más protección. Cabe mencionar que los sistemas de cifrado cifran el mensaje secreto y la existencia del mensaje secreto es clara. Por el contrario, los sistemas esteganográficos están optimizados específicamente para ocultar la existencia del mensaje secreto. Dependiendo de la aplicación, los sistemas de marca de agua pueden ocultar la existencia de la marca de agua o en algunos casos hacer pública la existencia de la marca de agua. Como el ámbito de esta tesis pertenece a la marca de agua, la esteganografía y la criptografía no se tratan más a fondo. Además, centraremos el contenido en el uso de señales anfitrionas tipo imagen.

Según el dominio en el que se realiza la marca de agua, los métodos de marca de agua se dividen en métodos de dominio espacial y métodos de dominio de transformación. Los métodos de dominio espacial alteran los valores de los píxeles en el dominio espacial y, en comparación con el dominio de transformación, normalmente implican una complejidad computacional mucho menor. Por el contrario, el dominio de transformación primero convierte los píxeles de la imagen en el dominio de transformación. Los píxeles transformados a menudo se denominan *coeficientes* en la literatura. Aparentemente, dicha transformación puede ser costosa desde el punto de vista computacional, pero el compromiso es que la robustez suele ser mayor que la de los métodos de dominio espacial. Normalmente, se aplica una transformación directa en la imagen y, después de la inserción de la marca de agua, se aplica una transformación inversa para recuperar la imagen con marca de agua en el dominio espacial. Algunas transformaciones comunes en la literatura de marcas de agua son (pero no se limitan a) la transformada de coseno discreta (DCT), transformada de ondícula (wavelet) discreta (DWT), Contourlet, Curvelet, Ridgelet, análisis de componentes principales (PCA), transformada de Karhunen-Loeve (KLT) y descomposición en valor singulares (SVD). Algunos otros métodos utilizan tanto el dominio espacial como el dominio de transformación para implementar la marca de agua. Estos enfoques a menudo se denominan *métodos híbridos* en la bibliografía. Si no se requiere información previa de la imagen anfitriona para la extracción de la marca de agua, entonces el método de marca de agua se conoce como *ciego*; de lo contrario, se denomina *no ciego*. Si se utiliza alguna información secundaria (no la imagen anfitriona) para la extracción de la marca de agua, el método de marca de agua se denomina *semi-ciego*. Si la imagen anfitriona se puede recuperar después de la extracción de la marca de agua, el método se denomina de marcado *reversible*; de lo contrario, se conoce como método de marca de agua *irreversible*.

En los últimos años, el concepto de *campo luminoso* (lightfield, LF) se ha utilizado cada vez más para la representación de imágenes 3D. Básicamente, el LF es una función escalar que describe la cantidad de luz que fluye en todas direcciones a través de cada punto del espacio. Michael Faraday fue el primero en proponer (en una conferencia de 1846 titulada *Pensamientos sobre las vibraciones de los rayos*) que la luz debería interpretarse como un campo, muy parecido a los campos magnéticos en los que había estado trabajando durante varios años. La denominación *campo luminoso* fue acuñada por Andrey Gershun en un artículo clásico de 1936 sobre las propiedades radiométricas de la luz en el espacio tridimensional. Desde un punto de vista óptico-geométrico, todo lo que percibimos visualmente, está iluminado por los rayos provenientes de fuentes de luz que se propagan a través del espacio hasta llegar a nuestro ojo. Básicamente, el LF describe la intensidad de cada rayo de luz en la escena en función del ángulo visual, la longitud de onda, el tiempo y la posición de visualización. Así, registra todo lo que potencialmente puede ser visto por un dispositivo óptico omnidireccional que es (supuestamente) capaz de capturar cada rayo del espacio. Levoy y Hanrahan definieron el LF como la función que describe la totalidad de los rayos de luz que atraviesan un volumen 3D dado. En otras palabras, el LF puede entenderse como la descripción de un conjunto denso de rayos de luz, cada uno de los cuales se origina en el vértice de un cono. Cada punto de un volumen 3D se considera como el vértice de un cono que transmite un número infinito de rayos con diferentes inclinaciones. Así, aparte del tiempo y la longitud de onda, el LF se representa típicamente usando cinco parámetros: posición del punto considerado (3 coordenadas espaciales) y dirección del rayo (2 ángulos directores). En realidad, la invariancia en propagación de los rayos (de acuerdo con la Óptica Geométrica), permite reducir su dimensional a 4D. Convencionalmente, a los valores obtenidos para un punto fijo del espacio en función de las 2 coordenadas angulares se le denomina imagen elemental (EI). Si (idealmente) se proporciona el LF de una escena, entonces es posible reconstruir la misma escena 3D sin pérdida de información. En la práctica, lo que realmente se captura en el mundo real es una submuestra del LF, no el conjunto completo de todos los rayos de la escena. Los dispositivos usados en esta captura se denominan de modo genérico cámaras LF. La principal diferencia entre una cámara LF y una convencional es que la primera captura los rayos individuales que inciden en un punto determinado del sensor de captura, mientras que la segunda registra la suma de todos los rayos que inciden en un punto específico del sensor. Esto facilita la reconstrucción 3D precisa de la escena recuperando los rayos individuales. El LF se puede adquirir de varias formas. En la configuración multicámara, se usa una matriz de cámaras 2D. En este caso, las dimensiones espaciales del LF están determinadas por las características intrínsecas de las cámaras, mientras que las dimensiones angulares están determinadas por el número y la disposición de las cámaras. Las cámaras pueden estar distribuidas en superficie plana, circular, rectangular o esférica. Esta configuración suele ser costosa y voluminosa. Además, la calibración de las cámaras puede llevar bastante tiempo. Otra alternativa es capturar el LF deslizando una sola cámara horizontal y verticalmente. A diferencia del sistema multicámara, la configuración de una sola cámara es mucho más barata y puede grabar el LF con mayor densidad. Sin embargo, la adquisición de LF por una sola cámara lleva mucho más tiempo que la de varias cámaras, lo que prácticamente hace que sea imposible grabar escenas dinámicas. Las cámaras LF estáticas también se pueden utilizar para capturar el LF. En ellas se emplea un único sensor estático y alguna distribución espacial de lentes (típicamente, una matriz de microlentes) para muestrear el LF.

A pesar de los numerosos métodos propuestos para la marca de agua sobre el LF, ninguno de ellos está adaptado para proteger la enorme cantidad de información angular incorporada en el LF. Se trata en todos los casos de aplicar los algoritmos ya desarrollados sobre imágenes 2D al LF con sus 4 dimensiones. El principal objetivo de esta tesis es lograr métodos de marca de agua LF maximizando la protección de la información espacial y angular al mismo tiempo. Según el conocimiento del autor, hay muy pocos trabajos que aborden los métodos de marca de agua personalizados para LF. Algunos artículos también han discutido la marca de agua de objetos 3D y el video de visualización libre, que, aunque con similitudes, es bastante diferente de la marca de agua sobre el LF. Cualquier método propuesto para la marca de agua del LF deberá tener sumo cuidado de no arruinar ni la información espacial ni angular del LF. A través de esta tesis se han propuesto dos métodos de marca de agua.

El primer método propuesto se basa en la DCT y la SVD, y trata de aprovechar el hecho de que los datos de LF generalmente tienen una correlación muy alta en las dimensiones espaciales y espectrales. Se supone que cualquier transformada como la DCT compacta la información en unos pocos coeficientes al proporcionar una descorrelación máxima. La transformada DCT es una aproximación de la KLT que descorrelaciona perfectamente los coeficientes. A diferencia de la base de funciones de la KLT, que dependen de la señal de entrada, las funciones base de la DCT están fijadas. Aunque la descorrelación de la DCT puede ser ligeramente menor que la de KLT y la descorrelación alcanzada es marginalmente menor, su costo computacional es menor debido a la eliminación del tedioso cálculo de las funciones básicas de la KLT. Además, en comparación con otras transformadas como la transformada de Fourier, los coeficientes transformados no tienen parte imaginaria y, por lo tanto, requieren menos datos para procesar. El hecho de que la DCT compacta la energía de la señal en pocos coeficientes lo hace muy interesante para la compresión y la marca de agua.

En este primer método propuesto, se parte del LF anfitrión y de una clave secreta como entrada. Según la clave secreta, para cada píxel de la marca de agua se seleccionan bloques de píxeles del LF original, a los que se aplica la DCT. Los coeficientes de los bloques transformados se ordenan en zigzag y se eligen los primeros coeficientes para incrustar la marca de agua. La razón de no incrustar la marca de agua en todos los coeficientes DCT es aumentar la robustez del método propuesto. Es bien sabido en la literatura que los coeficientes de baja frecuencia mejoran la robustez del método de marca de agua y los coeficientes de alta frecuencia son extremadamente propensos al ruido y otros ataques. Después de elegir los coeficientes DCT seleccionados, se factorizan utilizando la SVD. El valor singular correspondiente se utiliza para incrustar la marca de agua (el valor la marca de agua en el píxel considerado en la clave secreta). Cada bloque lleva un bit de la marca de agua. Según el bit de marca de agua, el valor singular aumenta o disminuye. El incremento o decremento del valor singular se determina mediante el factor de ganancia. Se requerirá el valor singular en el procedimiento de extracción para que se guarde en la *imagen de referencia*. Luego, se realiza la SVD inversa para obtener los coeficientes DCT del LF con marca de agua. Para generar el LF con marca de agua en el dominio espacial, se lleva a cabo la DCT inversa. Este proceso se repite hasta que todos los bits de marca de agua se incrustan en el LF del host.

Para extraer la marca de agua incrustada, se necesitan el LF con marca de agua, la imagen de referencia y la clave secreta. La clave secreta utilizada para la extracción de la marca de agua tiene que ser idéntica a la del procedimiento de incrustación, de lo contrario, la extracción de la marca de agua incrustada fallará. Si se introduce la clave secreta correcta en el sistema de extracción, los bloques correspondientes se ordenan a partir de los píxeles del LF marcado. La DCT y la SVD se realizan exactamente de la misma manera que para el procedimiento de inclusión. A continuación, el valor singular se compara con el valor correspondiente de la imagen de referencia. Si el valor singular es mayor que el valor correspondiente de la imagen de referencia, el bit de marca de agua extraído se considera uno; de lo contrario, se asume que es cero. La lógica detrás de este argumento es que si el bit de marca de agua incrustado es cero, entonces el valor singular ha disminuido por el factor de ganancia. Por el contrario, si el bit de marca de agua incrustado es uno, entonces el valor singular se ha incrementado en el factor de ganancia. Por tanto, el bit de marca de agua incrustado se puede extraer comparando el valor singular y el píxel correspondiente de la imagen de referencia. Después de extraer la marca de agua incrustada, los coeficientes DCT del bloque con marca de agua se obtienen mediante la SVD inversa. Antes de aplicar la SVD inversa, el valor singular del bloque con marca de agua se reemplaza con el píxel correspondiente de la imagen de referencia. Posteriormente, se aplica la DCT inversa a los coeficientes DCT para obtener el LF del anfitrión recuperado. Este proceso se repite hasta que se extraen todos los bits de la marca de agua.

La transparencia del LF con marca de agua se ha verificado objetiva y subjetivamente. Subjetivamente, el LF con marca de agua y el anfitrión parecían idénticos y no se detectó ninguna diferencia visual entre los dos campos de luz. Para garantizar la transparencia absoluta del LF con marca de agua, las partes de alta frecuencia del LF se han ampliado y no se encontraron diferencias visuales. Desde una perspectiva objetiva, la relación señal pico-ruido PSNR de la imagen con marca de agua fue mucho más que suficiente para permitir la detección de cualquier diferencia por el sistema visual humano (HVS) de acuerdo con los criterios objetivos establecidos en la literatura especializada. A diferencia de la mayoría de los métodos de marca de agua, la tasa de error de bits (BER) sobre la marca de agua recuperada permanece en cero independientemente de la intensidad del marcado y la marca de agua incrustada se puede extraer sin errores. Otra métrica que se utiliza para evaluar el rendimiento del método propuesto de manera objetiva es la similitud estructural media (MSSIM). La premisa básica de la MSSIM es que la percepción por el HVS de la calidad de la imagen se ve muy afectada por la similitud estructural del contenido de la imagen en lugar de los valores absolutos de los píxeles. También incorpora la intensidad media y el contraste de la imagen, que desempeñan un papel clave en la percepción de la calidad de la imagen por parte del HVS. La MSSIM siempre se mantiene por encima del 99% en los experimentos realizados. La robustez del método propuesto se ha medido frente al ruido gaussiano, la compresión JPEG y el filtrado de mediana. El método propuesto muestra una buena robustez frente a los ataques antes mencionados. Las simulaciones realizadas confirman la absoluta necesidad de utilizar pocos coeficientes DCT. Aunque el LF con marca de agua puede degradarse predominantemente por el ruido, la marca de agua se puede extraer. Como la mayor parte de la energía de la señal se concentra en coeficientes de baja frecuencia de la DCT, proporcionan más robustez frente al ruido gaussiano. Esta hipótesis se confirma completamente con las simulaciones. Las simulaciones también mostraron la importancia absoluta de la explotación de la DCT. La exclusión de la DCT conduce a resultados catastróficos. El método propuesto también exhibe una buena robustez contra el filtrado de mediana y la compresión JPEG, específicamente para el factor de calidad más común de 100%.

Para la justificación de la propuesta del segundo método de marcado, es interesante seguir el siguiente razonamiento. Aunque la DCT se usa ampliamente en la compresión de imágenes/video y marcas de agua, el supuesto subyacente es la independencia de los bloques adyacentes, ya que se comprime cada bloque por separado. Esto provoca artefactos notables, especialmente en velocidades de refresco bajas. Por el contrario, la DWT se aplica globalmente y no introduce artefactos de bloque. Como era de esperar, existe una similitud visual sustancial entre las EI vecinas en las direcciones horizontal, vertical y diagonal. En otras palabras, cada EI tiene una correlación mucho más alta con las EIs vecinas que con las demás. Nos referimos a la correlación de los píxeles de la misma EI como intracorrelación, mientras que la correlación entre las IE se denomina intercorrelación. La enorme intercorrelación de las IE trae la idea de obtener ventajas tanto de la intracorrelación como de la intracorrelación del LF. Es por eso que presentamos la DWT 4D para introducir la marca de agua en el LF. La razón subyacente de emplear la DWT 4D para la marca de agua LF es exactamente la misma que usar la descorrelación 3D de secuencias de video (descorrelacionar las filas y columnas de cada fotograma y los fotogramas cercanos). El uso de la DWT 4D para la marca de agua LF garantiza la obtención del máximo beneficio de una tremenda intercorrelación de píxeles LF. Incluso si los numerosos métodos de marca de agua de imágenes 2D son directamente aplicables al LF, siempre existe la preocupación de que tales métodos puedan arruinar la valiosa información angular. Por el contrario, la DWT 4D gana la tremenda redundancia del LF a lo largo de las dimensiones espaciales y angulares. Como la DWT es separable, cada dimensión se descompone primero mediante un filtro de descomposición. Aunque el orden de descomposición de las diferentes dimensiones no importa, el LF se descompone selectivamente primero en los dominios espaciales horizontal y vertical y luego en el dominio angular horizontal y vertical.

Por otro lado, en el segundo método propuesto se explota también la enorme redundancia entre los canales cromáticos del LF. Teóricamente, una correlación tan enorme puede reducirse mediante DCT, DWT, Fourier o cualquier otra transformación. No obstante, dada la corta longitud de los vectores de píxeles en el dominio RGB, no parece prudente utilizar filtros que tengan más de tres coeficientes. En cambio, un enfoque muy común para la descorrelación de canales es realizar una transformación de espacio cromático de RGB a YUV, y operar sobre el canal de luminancia (Y). Esta técnica se usa con frecuencia en muchos estándares de compresión. Por lo tanto, el segundo método de marca de agua propuesto, primero convierte el LF anfitrión de RGB a YUV. Después de la conversión de color, la transformada de ondas 4D se aplica al componente Y. Entre dieciséis subbandas generadas en la DWT, la subbanda de frecuencia más baja (LLLL) es la elegida para la inserción de marcas de agua. La principal razón para elegir la subbanda LLLL es la existencia de baja frecuencia de información espacial y angular. La incrustación de la marca de agua en una parte de frecuencias tan bajas del espectro de LF proporciona la máxima robustez para la marca de agua. Después de la DWT 4D, la DCT se aplica a algunos coeficientes de esta subbanda. Los bloques se adquieren a partir de coeficientes LLLL en función de la clave secreta. En otras palabras, la clave secreta especifica qué coeficientes de la subbanda LLLL se explotarán para crear un bloque que lleve la i-ésima fila y la j-ésima columna de la marca de agua binaria. La utilización conjunta de la DCT y la DWT dará como resultado la obtención de las ventajas de ambas transformaciones. Específicamente, la suposición de independencia de los bloques DCT adyacentes es mucho más precisa en el dominio de ondículas. Es de notar que la DCT no cambia las dimensiones de la matriz de entrada. Como se mencionó anteriormente, los coeficientes de alta frecuencia de la DCT son extremadamente propensos al ruido. El uso de tales coeficientes conduce al deterioro de la robustez de la plataforma de marca de agua frente a varios ataques que incluyen, entre otros, el ruido gaussiano. Por tanto, no todos los coeficientes DCT se utilizan para la inserción de la marca de agua. Para recoger los coeficientes DCT seleccionados, primero se reconfigura la matriz de los coeficientes DCT a un vector en orden en zigzag. Los primeros elementos de dicho vector son seleccionados y luego factorizados por la SVD. Dependiendo del bit de marca de agua, el valor singular aumenta o disminuye según el factor de ganancia especificado. Estos valores singulares serán necesarios para extraer la marca de agua incrustada. En consecuencia, el valor singular se registrará en la imagen de referencia. Luego, se realiza la inversa de todas las transformaciones realizadas para reconstruir el bloque con marca de agua. Primero, los coeficientes DCT se reconstruyen a partir de los valores singulares modificados. Asimismo, se aplica DCT inversa para reconstruir los coeficientes DCT con marca de agua. La DWT inversa se utiliza para producir la marca de agua en el dominio de luminancia. Finalmente, el LF con marca de agua se construye convirtiendo el LF con marca de agua del dominio de YUV al dominio RGB.

Ahora también se sigue el mismo procedimiento para extraer la marca de agua incrustada. En otras palabras, la conversión de color, DWT 4D, DCT y SVD se llevan a cabo tal como se describe en el procedimiento de inclusión. Después de aplicar la SVD, el valor singular se compara con el píxel correspondiente de la imagen de referencia y de esta manera, se detectará el bit de marca de agua incrustado. Si el valor singular es mayor que el píxel correspondiente de la imagen de referencia, se supone que el bit extraído es igual a uno; de lo contrario, se considerará cero. Con el fin de verificar la transparencia del marcado tras las operaciones citadas, se ha comparado subjetiva y objetivamente la similitud del LF del anfitrión y del LF con marca de agua. Además, tanto el LF anfitrión como el LF con marca de agua se han proyectado en un monitor de imagen integral (IIM), preparado para reconstruir ópticamente las escenas 3D a partir de ambos LFs, para verificar la protección de la información angular del LF y comprobar que no se afecta significativamente la reconstrucción 3D de la escena original. Para comprobar el impacto del método propuesto en la percepción visual en 3D, se proyectaron en el IIM tanto el LF anfitrión como el LF con marca de agua. No se encontró ninguna diferencia (ni siquiera menor) entre las dos escenas reconstruidas. La paralaje obtenida con el anfitrión y el LF con marca de agua parecían idénticos, lo que indica la fidelidad de la información angular tras el marcado. El PSNR del LF con marca de agua siempre superó los $48dB$ en los experimentos realizados, lo que supera con mucho la capacidad típica de HVS de detectar cualquier diferencia. Aún para valores de la intensidad de la marca de agua menores que 145, la BER siempre permaneció cero. La BER cero, independientemente de la intensidad de la marca de agua, es bastante alentadora, ya que este no es el caso de la mayoría de los métodos de marca de agua en la literatura. El MSSIM también se mantuvo por encima de 97% independientemente del factor de ganancia. Estos resultados objetivos son totalmente consistentes con la fidelidad objetiva del LF proyectado con marca de agua a IIM. El uso conjunto de DCT y DWT mejoró considerablemente la solidez del método propuesto contra el ruido gaussiano, la compresión JPEG, el filtrado de mediana y la compresión JPEG2000.

Se observó que la exclusión de la conversión de color, DCT y DWT 4D conduce a una desviación significativa del rendimiento óptimo. Como el método propuesto incorpora conversión de color, DWT 4D, DCT y SVD, uno puede preguntarse si es realmente posible lograr el rendimiento óptimo incluso cuando se eliminan una o más transformaciones. Cada una de las transformaciones individuales sirve para descorrelacionar los píxeles del LF a lo largo de una dimensión específica que no puede lograrse con ninguna de las otras transformaciones. La conversión de color elimina la alta correlación entre los canales R, G y B de la imagen en color que es imposible de lograr con DCT, DWT 4D o SVD. La transformada de ondas 4D se emplea para disminuir la inmensa intercorrelación e intracorrelación en el LF. Obviamente, este tipo de descorrelación nunca puede ser conseguida por otras transformadas. Finalmente, el DCT se utiliza para compactar la energía de la señal dentro de unos pocos coeficientes. Tal compactación de los coeficientes nunca se logrará si se elimina el bloque DCT. Los resultados experimentales indican que el rendimiento óptimo solo se puede lograr si se utilizan todas las transformadas mencionadas anteriormente. También se ha investigado el impacto del tamaño del bloque y el número de coeficientes DCT. Se observó que reducir el tamaño del bloque de seis a tres generalmente deteriora la robustez del método propuesto frente a los ataques estudiados. Asimismo, el uso de muy pocos o demasiados coeficientes DCT provoca una caída significativa de la robustez del método propuesto.

Publication List

I) **Ansari, A.**, Dorado, A., Saavedra, G., and Martinez-Corral,. M. (2017). Plenoptic image watermarking to preserve copyright., Proceedings of the SPIE 10219, Three-Dimensional Imaging, Visualization, and Display 2017, 102190A.

II) **Ansari, A.**, Hong, S., Saavedra, G., Javidi, B., and Martinez-Corral, M. (2018). Ownership protection of plenoptic images by robust and reversible watermarking. Optics and Lasers in Engineering 107, 325-334.

III) **Ansari, A.**, Saavedra, G., and Martinez-Corral, M. (2020). Robust light field watermarking by 4D wavelet transform. IEEE Access 8, 203117-203133.

IV) **Ansari, A.**, Hong, S., Martinez-Corral, M., and Saavedra, G., (2017), Plenoptic Image Watermarking for Ownership Protection, 16th Workshop on Information Optics (WIO), Interlaken (Switzerland), 2017.

Abbreviations

BER	Bit Error Rate
CR	Compression Ratio
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
EI	Elemental Image
EPI	Epipolar Image
HVS	Human Visual System
IIM	Integral Imaging Monitor
LF	Light Field
MR	Multiresolution
MSSIM	Mean Structural Similarity
μI	Microimage
PSNR	Peak Signal to Noise Ratio
QMF	Quadrature Mirror Filter
SVD	Singular Value Decomposition

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Information Hiding

Information hiding is a general term covering several realms. As the name implies, information hiding is the *secret* transmission of the information of a message by embedding and retrieving it into and from a digital medium. Information hiding, classically refers to imperceptible insertion of the secret information into content [1]. The information hiding basically refers to imperceptible embedding of the information or keeping the existence of the information secret. The former approach is often known as watermarking while the latter is commonly refereed to as steganography. The rapid growth of the digital infrastructures has facilitated the production, storage, modification and transmission of the media. Despite the enormous opportunities provided by the internet and other networks, the risk of copyright infringement has rosen. The researchers have investigated the amount of the illegal media exchange among twelve nations of Europe, South America and Asia. Poort et al. found out that up to 60% of the audio-visual books, music and other digital content is acquired illegally [2]. This single example shows the absolute importance of copyright protection which is only one of the numerous applications of information hiding. In a broad sense, the information hiding can be divided into i) steganography ii) cryptography, and iii) watermarking.

1.2 Information Hiding Techniques

1.2.1 Steganography

Steganography can be defined as the mechanism of concealing secret data within a non-secret medium. The main purpose of steganography is covert communication. The medium can be image, audio, video or text document. The transmission of an ordinary message causes no suspect by third party.

The term ”steganography” was originally derived from Greek words *steganos* meaning ”covered” and *graphia* which means ”writing” [3]. Steganography has been practiced for millennia. A well-known example is cited by Herodotus (485-525 B.C.) who is one of the first Greek historians. He elaborates the war between the great Persian empire and Ionian city of Miletus. According to his book [4], Histiaeus shaved his slave’s head and tatooed a secret message on his scalp. The message was about encouraging *Aristagoras*, the city ruler to revolt against Persian Achaemenid empire. He then waited for the slave’s hair to grow. The slave, obviously causing no suspect, freely traveled to Miletus and delivered his master’s message.

Another ancient example is Pliny the Elder’s work. He explained that the milk of thithymallus plant goes invisible when used to write on the paper. It can become visible when exposed to heat. Hence, he proposed one of the earliest recepies for invisible ink [5].

Theoretically, any kind of image, video, audio or text may be concealed within any kind of image, video, audio or text. Nevertheless, most of the steganography literature concerns images [6, 7]. The principal reason of employing steganography techniques is to transmit a secret signal without letting the third party knowing the very existence of the secret message. The media which carries the secret message is a common media typically drawing no attention. It is worth mentioning that the host (i.e. the carrier media which the secret message is concealed within) signal is not encoded in steganography. Encoding the host signal used for steganography basically defeats the most underlying purpose of not drawing the attention of the third party.

1.2.2 Cryptography

The term *cryptography* is originally from Greek words of *crypt* meaning ”hidden” or ”secret” and *graphia* which means ”writing” [8]. Cryptography is the act of ensuring the imperceptibility of a *host* signal by encoding its whole content to make it meaningless. Unlike the steganography and watermarking, the whole transmitted message is ciphered in cryptography and the secret message is not embedded within another media.

The cryptography is divided into two major categories. The first kind of cryptography is symmetric in which both the sender and receiver have the very same key. Conversely, asymmetric encryption system utilizes a public key to encrypt the message and a private one for the decryption. As the relation between the public and private key are extremely complicated, it highly improves the security of asymmetric encryption systems. On the other hand, the symmetric encryption systems are usually faster mainly due to the shorter length of the key [9, 10]. The encoded host should be recovered via decryptography [11]. Cryptography is used when the existence of the communication is not clandestine but the exchanged message shall remain secret. SSL and TLS are some common examples of encryption protocols for email services [12].

1.2.3 Digital Watermarking

In the literature, the generally accepted definition of the digital watermarking is to as imperceptible insertion of the *hidden information* into the *host* signal which can be image, audio, video or text [13]. The application of watermarking can be originated to the ancient times. Paper watermarks have been widely used in Italy about 1282. By the 18th century, watermarking techniques began to be utilized to combat forgery on money and other important documents [14]. A more classical example is Margaret Thatcher’s approach to identify which member of her cabinet would reveal the confidential documents to the newspapers. She used an invisible watermarking method of 80s. After several cabinet documents had been leaked to the press, Thatcher ordered that the word processors being used by government employees to encode their identity in the word spacing of the document.

She provided each of her ministers with a unique document in which the word spacing of the documents were identifiable. This allowed for disloyal ministers to be quickly found out. It is not easy to find the first instances of the digital watermarking. In 1979, Zepanski defined the digital watermarking as machine-detectable pattern that can be embedded into documents for anti-counterfeiting [15]. As opposed to watermarking and steganography, the existance of the secret message is known in cryptography. In watermarking, the existence of the watermark sometimes may be announced but in steganography the existence of the embedded message always remains secret. On the other hand, in watermarking, the host can be more important than the secret message while the main purpose of steganography is to transmit a secret message within the host signal. As this dissertation concerns watermarking, the scope of the literature review is narrowed down to the watermarking. Additionally, as this research addresses light field (LF) watermarking, the rest of this manuscript is confined to image watermarking even though most of the arguments also hold for other types of watermarking media.

1.3 Literature Review of Image Watermarking

The relentless development of the internet, network bandwidth and cost-effective digital devices, have served to ease of creation, modification and transmission of the digital contents. As promising as such opportunities are, there has been serious concerns about breach of the rights of the intellectual property [16]. In 1998, the US congress proposed the Digital Millennium Copyright Act bill which would strictly prohibit copyright violation of any kind [17]. As mentioned in section 1.1, Poort et al. reported that more than 60% of the audio-visual books, music and other digital content have been exchanged illegally in their observed nations. Their comprehensive and diverse investigation highlights the tremendous infringement of copyright law all over the globe [2]. Such widespread breach of copyright law substantiates the absolute necessity of watermarking research. Copyright protection is only one of the numerous applications of the digital watermarking. Besides, watermarking can also be used for authentication [18], error detection/correction [19], tamper detection [20] and broadcast monitoring [21]. Despite the motivation that the watermarking method is employed for, it is always desired to meet three major requirements namely [22]

(i) Transparency: The embedded watermark should cause no perceptible difference in the watermarked image. Every endeavour should be made to ensure delivering a watermarked image as identical to the host image as possible. The transparency should be taken into account on all steps of the watermark insertion such as choosing color channels or frequency components that carry the watermark.

(ii) Robustness: The embedded watermark should survive the various *attacks* that may occur. In watermarking context, the attack denotes any operation/event resulting in modification of the pixel values. Some common examples of attacks are noise, compression and filtering. The embedded watermark should be difficult to remove or alter without imposing substantial degradation to the host image.

(iii) Capacity: The higher the capacity provided by the watermarking platform, the more information can be embedded to the host image. The capacity of the watermarking scheme refers to the amount of the information that can be embedded into the image.

It is obvious that these requirements conflict with each other. Figure 1.1 shows the conflicts of the watermarking requirements. For example, as will be discussed in section 3.1, most of the times the higher transparency implies lower robustness and vice versa. A very similar conflict pertains between the capacity and the two other requirements. Achieving higher capacity might hinder a higher transparency/robustness and vice versa.

Depending on the robustness of the watermarking method against different attacks, there are different categories of watermarking methods. The robust watermarking demonstrate good resilience against attacks and is often used for copyright protection [22–24]. Conversely, the fragile watermarking is easily collapsed even if exposed to any minor attack and is typically demanded in authentication and tamper detection [20, 25–29]. The

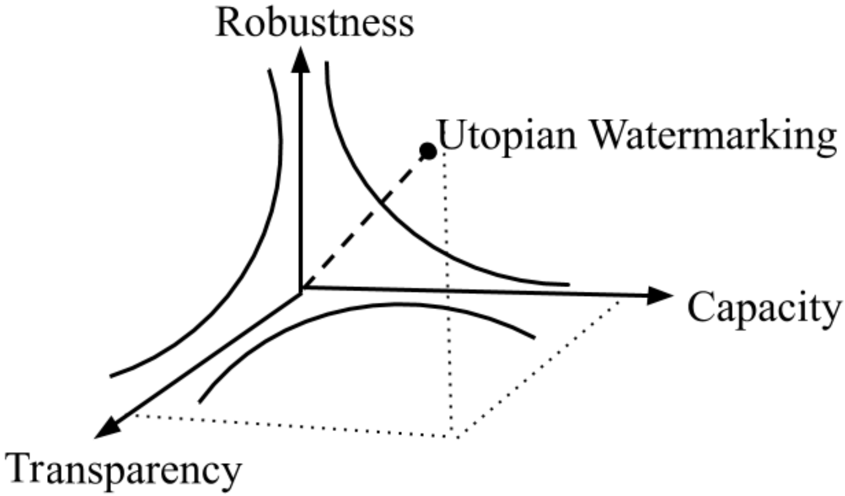


Figure 1.1: The trade-off among the main requirements of the watermarking platform [33]

semi-fragile watermarking provides good robustness against some specific attacks and remains vulnerable against some others [30–32].

Depending on the domain that the watermarking is realized, the watermarking methods are categorized to spatial domain and transform domain. The spatial domain methods basically embed the watermark by alteration of the pixels without carrying out any transform. In contrast, the transform domain methods technically embed the watermark into coefficients of one or several transforms. The spatial domain methods typically have lower computational complexity whereas the transform domain methods deliver considerably higher robustness [34–41]. Some of the commonly used transforms in watermarking area are Discrete Cosine Transform (DCT) [42–45], Discrete Wavelet Transform (DWT) [46–50], Contourlet [51–54], Curvelet [55, 56], Ridgelet [57, 58], Principal Component Analysis (PCA) [59–61] and Singular Value Decomposition (SVD) [28,62–64]. There are other watermarking methods which utilize both spatial and transform domain for watermark insertion and are often known as *hybrid* methods [65–70].

The watermarking methods can also be classified based on the necessity of the host image in the extraction procedure. If the host image is not required for the extraction procedure, the watermarking method is referred to as blind, otherwise it is known as non-blind [71–76]. If some side information (not the host image) is used for watermark extraction, then the watermarking method is called *semi-blind* [77]. Depending on the possibility of recovering the host image from the watermarked image, the watermarking methods are divided into reversible and irreversible methods. As it can be understood from the terms, the former makes it possible to recover the host image through the extraction procedure. In contrast, the irreversible watermarking does not deliver the recovered host image after watermark extraction [78–82]. Watermark may be embedded into chrominance or the luminance component. Even though the luminance component delivers higher robustness, the Human Visual System (HVS) is extremely sensitive to the modification of the luminance information. Hence, the chrominance component provides higher transparency [83–86].

1.4 The concept of the Light Field (LF)

The LF is a scalar function that describes the amount of light flowing in every direction through every point in space. In other words, it considers the usual three-dimensional space to be penetrated by light that propagates in all directions. The light can be blocked, attenuated or scattered while doing so [87]. The *intensity* of all possible light rays is given by the five-dimensional plenoptic function, and the light magnitude of each ray is given by the radiance. Michael Faraday was the first to propose (in an 1846 lecture entitled ”Thoughts on Ray Vibrations” [88]) that light should be interpreted as a field, much like the magnetic fields on which he had been working for several years. The phrase light field was coined by Andrey Gershun in a classic paper on the radiometric properties of light in three-dimensional space (1939) [89, 90].

Everything we visually perceive, is illuminated by light rays coming from a light source (e.g. lamp or sun) which are propagated through space

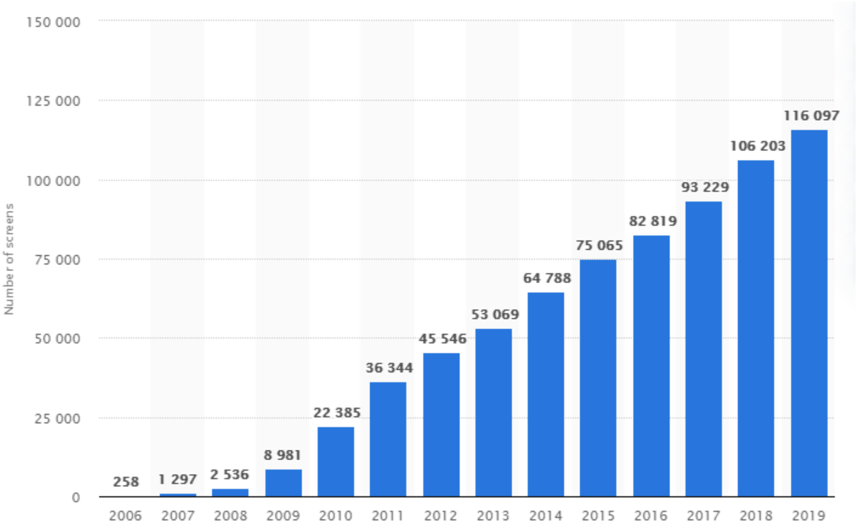


Figure 1.2: Number of digital 3D cinema screens worldwide from 2006 to 2019 [91]

reaching our eye. It can also be that we see an object which is illuminated by another light source. An illuminated surface, partly absorbs and partly reflects (or refracts) the light to another surface, where it finally reaches our eyes after several bounces. Human visual perception vastly depends on the subsample acquired from LF by moving in it forming an idea of relative position and color of the objects of the scene.

Over the past few years the demand for 3D LF-based reconstruction has continuously increased. The upward trend of market of LF display is a sign of increasing demand of such displays. No matter if the end user is watching a movie or is playing a videogame, giving a realistic sense of depth is highly demanded. Over the last years, both academia and industry have shown great interest to the 3D content production. Figure 1.2 shows the monotonous increasing trend of 3D screens since 2006. This graph fully supports the argument of upward trend in popularity of 3D screens. LF displays reproduce the light from objects of the scene. Therefore, unlike the common 3D displays, no specific glasses would be required to gain 3D perception. To attain this goal, a dense field of light rays are reconstructed giving the viewer a glasses-free 3D perception [92]. Figure 1.3 shows two different commercial LF displays manufactured by Holografika.

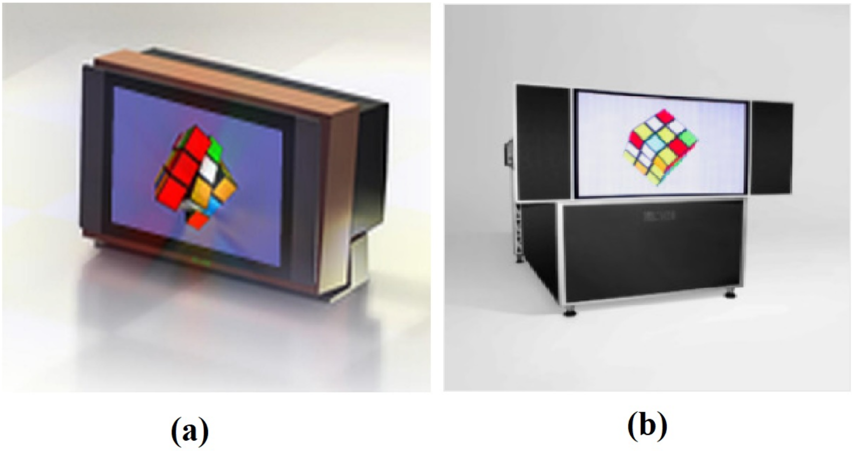


Figure 1.3: HoloGrafika LF displays a)HoloVizio 81WLT b) HoloVizio 722RC

1.5 Parametrization of the LF Rays

In conventional 2D cameras, a pixel is created by acquiring 2D projection of light rays on the sensor plane omnidirectionally. In other words, the common 2D camera captures the summation of all the rays passing through a given point. Conversely, the LF camera captures individual rays hitting a point. Figure 1.4 visualizes different capturing methods of the conventional and LF cameras. The image capturing sensors usually measure the information only from two dimensions. Hence the multiple samples have to be captured along angular dimensions [92]. As the name implies, LF (also known as plenoptic function) refers to a function which assigns a single value to each ray of the scene. At any point of the space, the LF gives the value assigned to each ray passing through it; in other words, at

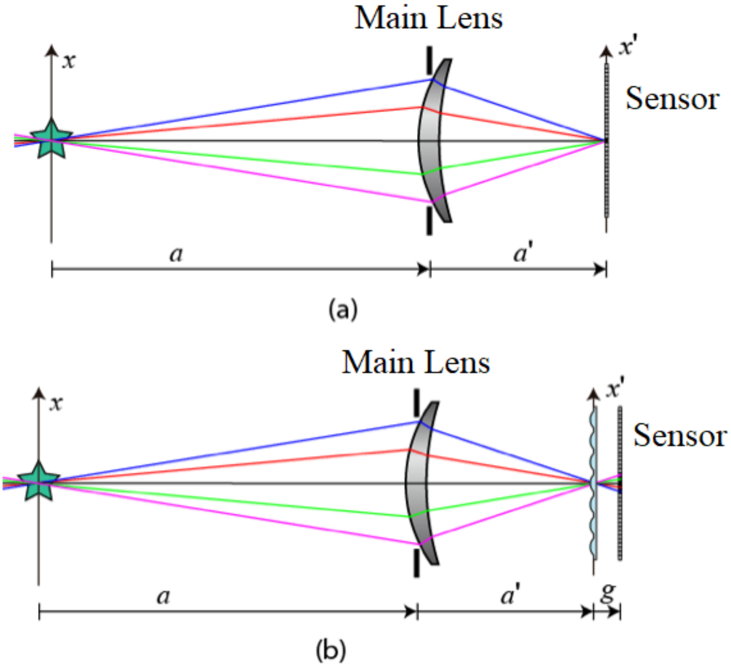


Figure 1.4: (a) Summation of the rays falling onto a pixel of a conventional 2D camera; and (b) Capturing of individual rays by LF camera. [33]

a given point it accounts for the assigned value to the ray traveling at a given direction and passing through that particular position. Even though the origin is backed to Leonardo da Vinci's works, the plenoptic function was not mathematically formulated until E.H. Adelson et al. published their groundbreaking paper in 1991 [93]. Levoy and Hanrahan defined LF as a way of describing all light rays passing through a given 3D volume [94,95]. In this way, the LF can be understood as a description of a dense array of light rays each originating from the vertex of a cone. Every single point of a 3D volume is considered as the vertex of a cone transmitting infinite number of the rays along slanted directions. Figure 1.5 shows two eyes located at different points of the space each forming a different cone. Obviously, the eye cannot see the rays coming from behind and that is why the model employs a cone. Figure 1.5 shows two samples of what Leonardo da Vinci calls *radiant pyramid*:

> *The body of the air is full of an infinite number of radiant pyramids caused by the objects located in it. These pyramids intersect and interweave without interfering with each other during the independent passage throughout the air in which they are infused [96].*

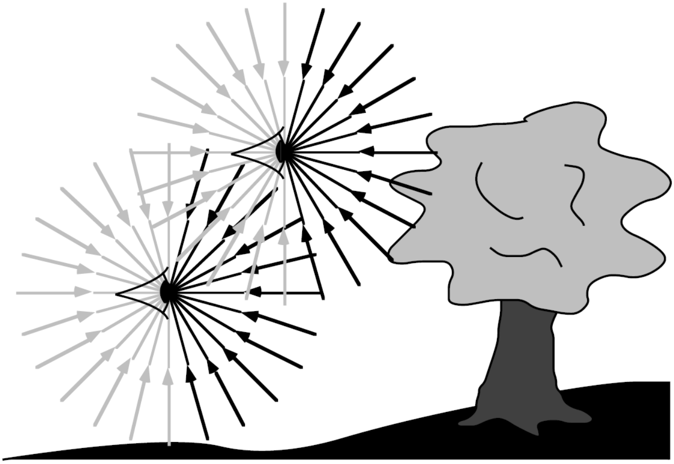


Figure 1.5: The plenoptic function describes all the rays of the scene even the ones coming from behind the eye. Only some slanted directions converging to the vertex of the cone are visualized. [93]

The terms *plenoptic function* and LF are often used interchangeably and represent the very same concept. The plenoptic function is the 7D function representing the intensity or chromacity of the light observed from every position and direction in 3D space. The radiance along all such rays at any moment and for any wavelength in a region of three-dimensional space is called plenoptic function. Ideally, from geometrical optics perspective, the plenoptic function should be a multidimensional function describing the set of light rays traveling in every direction through every point in 3D space [92]. The 7D plenoptic function is parametrized as

$$P = LF(V_x, V_y, V_z, x, y, \lambda, t) \,, \tag{1.1}$$

where V_x, V_y and V_z represent the 3D coordinates of a perfect eye, at any moment t, for every wavelength λ, that captures the rays passing through any possible spatial coordinates (x, y) of an imaginary picture plane placed at a unit distance from the pupil. In this contex, the perfect eye refers to an imaginary ideal eye capable of perceiving the continous range of wavelength over a continous period of time. Aditionally, perfect eye distinguishes all the rays regardless of the wavelength and the point the ray is being radiated from. Apparently, the LF is an idealistic description of the scene. If the plenoptic function of a scene is available, the scene can be reconstructed perfectly and losslessly. Unlike the 2D cameras, the plenoptic camera is supposed to capture the individual rays. From a practical point of view, no one can keep observing an infinite number of outcoming rays of the scene from all the viewpoints for any possible wavelength. Therefore, similar to other works in LF literature, through this dissertation a reduced decription of the plenoptic function will be used [92, 97–99]. Even though capturing the perfect plenoptic description of a static scene is a challenge, it can be even more complicated for videos in terms of capturing, storage, transmission and processing. Another common assumption of LF sampling is to consider the LF as monochromatic and remove the parameter λ. Additionally, as the LF is supposed to be static, the argument t is also eliminated.

1.6 LF Mathematical Description

The typical 2D RGB images are usually described using three parameters namely the row (x), the column (y) and the channel number (ch). As mentioned earlier in section1.4, the LF has huge dimensionality imposing enormous computational burden on employed hardware platforms. Besides, as mentioned in section 1.4, capturing perfect unsampled LF of the scene is not feasible for any manufacturer in the real world. This regarding, the

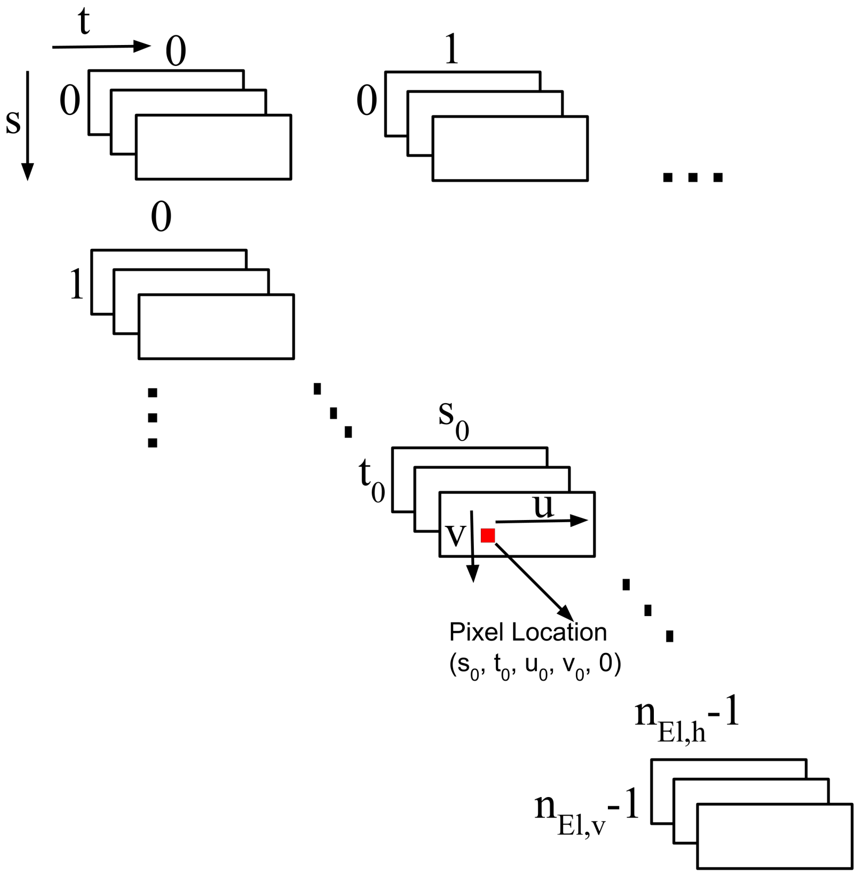


Figure 1.6: Multidimensional representation of LF

LF is usually described using five parameters [92]. The multidimensional nature of LF is illustrated in Fig. 1.6. The LF is written as

$$LF(s,t,u,v,ch). \tag{1.2}$$

Table 1.1 shows the parameters used in eq. 1.2. Each single image in Fig. 1.6 is known as an *elemental image (EI)*. The arrangement shown in Fig. 1.6 is often referred to as *integral image (II)*.

Table 1.1: Components of position vector in light field

s	The row where the elemental image lies among other views
t	The column where the elemental image lies among other views
u	The row of the the elemental image in which the pixel is located
v	The column of the the elemental image in which the pixel is located
ch	The channel number (from zero to two for RGB images)

LF has more dimensions than common RGB images. The LF can be subsampled along various dimensions. Throughout this dissertation, we will use the term *elemental image (EI)* for referring to the individual images (triple rectangle) shown in Fig. 2.5. As shown in Fig. 1.6, the LF can be represented as a matrix of EIs. Such arrangement is commonly known as *integral image* in the literature [22]. The EI is obtained by fixing the two spatial dimensions s_0, t_0. The first two components of the position vector of the illustrated pixel in Fig. 2.5 indicate the elemental image (s_0, t_0) where the pixel lies in. The $EI(s_0, t_0)$ is described as

$$EI(s_0, t_0) = LF(s_0, t_0, :, :, :). \tag{1.3}$$

Similarly, a *microimage* μI represents the same pixel of all the EIs by fixing the angular dimensions u, v

$$\mu I(u, v) = LF(:, :, u, v, :). \tag{1.4}$$

While an EI carries the spatial information of the LF, μIs are considered as angular subsamples of LF. Figure 1.7 shows the arrangement of EIs and μIs of a raw LF visualizing the concept of eq. 1.3 and eq. 1.4. Each color square in Fig. 1.7 shows one single pixel of the raw LF. The thick black lines drawn horizontally and vertically, separate the EIs. All the pixels of the same color, correspond to the same row and column of individual EIs. The combination of all the pixels on the row u and column v of EIs forms $\mu I(u, v)$. Another common representation of LF are Epipolar Images (EPI), which are LF subsample slices obtained by fixing one spatial and one angular dimension. The horizontal EPI is acquired by fixing the columns of

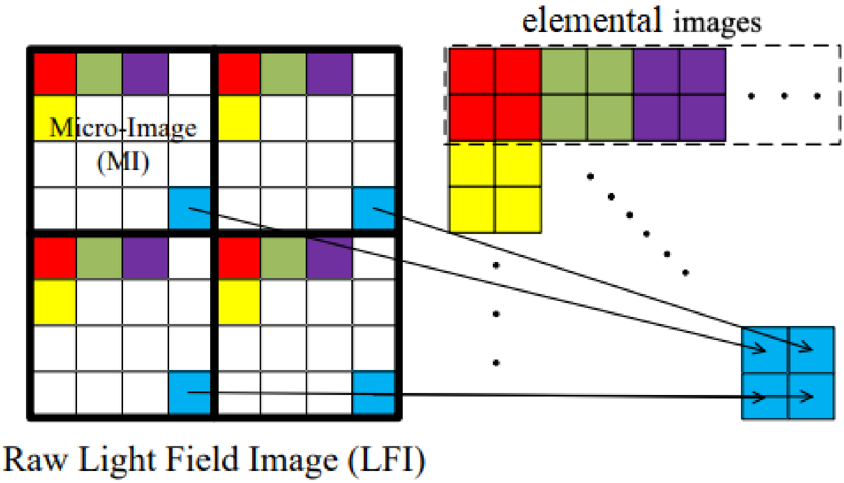


Figure 1.7: Arrangement of the EIs and µIs of the LF [100].

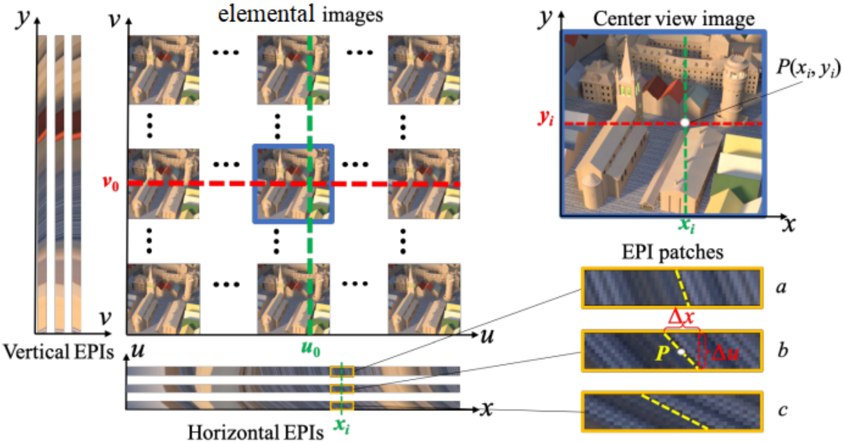


Figure 1.8: Arrangement of the EPIs [101].

the EIs and μI. Likewise, the vertical EPI is obtained by fixing the rows of the EIs and μIs. Figure 1.8 visualizes the horizontal and vertical EPIs.

As is obvious from Fig. 1.8, the horizontal(vertical) patches are created by concatenating multiple horizontal(vertical) *EPI*s.

1.7 LF Imaging

This section serves to give a brief review on possible methods of LF capturing. Generally speaking, the LF may be acquired either by common 2D camera or plenoptic cameras. Each of these methods has its own pros and cons. Besides, if 2D camera is used, it is possible to capture the LF by one or multiple cameras.

1.7.1 Multi-Camera Setup

Figure 1.9: Tightly packed cameras with telephoto lenses (Stanford multi-camera array setup) for LF capturing [102]

The multi-camera setup consists of an array of cameras distributed on planar, circular, rectangular or spherical surface to acquire the LF subsamples from different viewpoints. In this setup, the spatial dimensions of the LF are determined by the technical characteristics of the cameras whereas the angular dimensions are determined by the number and arrangement of the cameras [94]. In 2001, MIT Laboratory for Computer Science employed an 8×8 matrix of cameras to capture LF video. To minimize the required bandwidth, they used some specific rendering algorithm [103]. In 2002, Wilburn et al. applied six video cameras to record synchronized video. Each camera was controlled with a controller board and used MPEG compression scheme to reduce the redundancy [104]. This research was later improved to use 125 video cameras achieving several thousands frames per second (FPS) performance [105, 106]. Another LF acquistion platform was proposed using 6×8 cameras capable of estimating new views by reconfiguring the cameras position [107]. In 2005, a vector of eight video cameras was used for dynamic image renderring. This approach was called *plenoptic video* and would broadcast 15 FPS with the resolution of 256×256 [108]. Figure 1.9 shows a multi-camera array used for LF acquisition. The multi-camera array may employ planar or spherical surface for recording LF [94, 102, 109]. This assembly was designed through Stanford Immersive Television project which was funded by Intel, Sony, and Interval in late 90s. These were some of the most remarkable practical platforms of LF capturing which was ever launched by the time. Nevertheless, as can be seen from Fig. 1.9, Stanford camera array was quite bulky and would not fit to the portable devices as laptops, mobile phones and tablets. Multi-camera setup is typically expensive and voluminous. Furthermore, calibration of cameras demands a good deal of time and effort. The next noticeable multi-camera approach was released next year in which would capture the LF by an 8×8 matrix of cameras and stream the video through network [110].

1.7.2 Single-Camera Setup

As opposed to multi-camera setup, the *single-camera* setup employs only one single camera to capture the LF. The basic premise of the single camera setup is to slide the same camera along X, Y and Z axis. The camera is often placed on a gantry and its movement is precisely controlled. The gantry

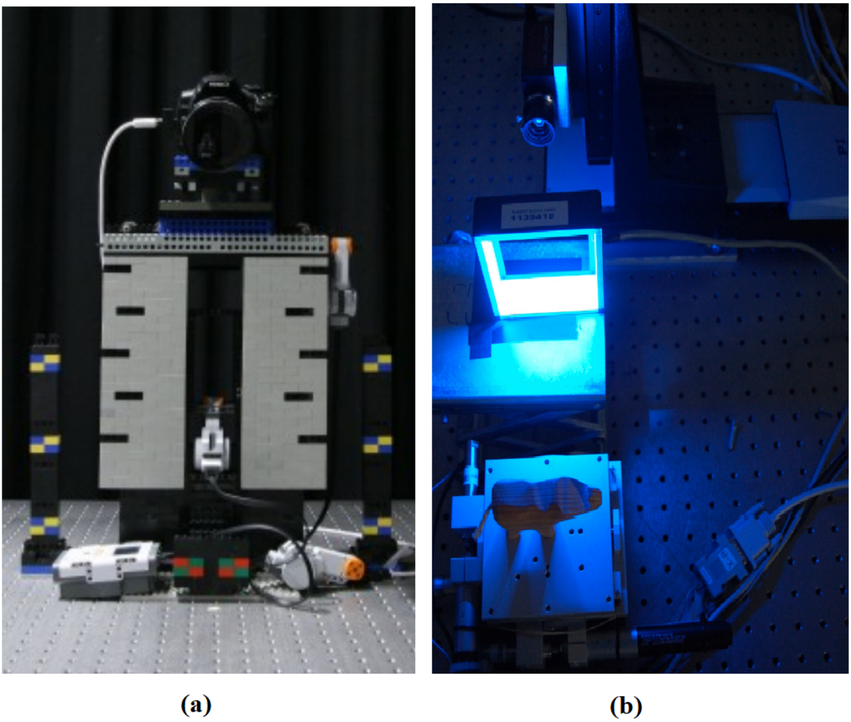


Figure 1.10: Single-camera setup for LF capturing (a) Stanford LEGO Gantry [111]. (b) HCI light field Gantry [112].

setup has been launched by Stanford University and University of Heidelberg [111–113]. These two setups have been shown in Fig. 1.10. Figure 1.10(a) shows the setup which is very similar to Fig. 1.10(b) of University of Heidelberg. In contrast to multi-camera system, the single-camera setup is much cheaper and can record the LF more densely. Nevertheless, the LF acquistion by single camera takes considerably longer than multi-camera which practically makes it impossible to record dynamic scenes. In 2017, a microlens array (MLA) was used to increase the field of view of the captured LF. The optical setup was mounted on a mechanical arm that could rotate around a fixed axis improving the FOV [114].

A major drawback of single-camera LF acquisition is the slow imaging. The aforementioned methods capture images from different viewpoints by moving the image sensor. Regarding the longer time required for high precision movement of the camera, the frame rate of the single camera capturring is much lower than that of the multi-camera setup. Fortunately, some fast time-sequential capture approaches have also been introduced. Therefore, some researchers attempted speeding up the LF acquisition of single-camera setup [115]. Another innovative approach to speed up the single-camera LF imaging was proposed by Taguchi et al. using a mirror ball and a single camera. As the LF was captured by moving the camera along the mirror's axis of rotation they called it *axial LF*.

1.7.3 Plenoptic Camera

Typically, the multi-camera and single-camera are not appropriate for portable devices. Hence some efforts have been made to integrate multiple cameras in a small surface. Figure 1.11 shows some of plenoptic cameras [116]. The ProFusion 25 released a matrix of 5×5 VGA cameras placed in a small box to capture LF video at 25FPS [117]. In 2013, Pelican camera array was introduced. This camera array is as small as a coin and delivers $4 \times 4 \times 1000 \times 750$ pixels. Later in 2014 Lytro Illum was released which turned to be one of the most popular plenoptic cameras. Raytrix is another remarkable player in the market. Figure 1.12 shows some sample LF images captured by Raytrix camera. Some other manufucturers have produced their own plenoptic camera as Adobe, Toshiba , CAFADIS and KILens [118].

1.8 LF applications

LF is a very recent topic and has rarely been addressed either by academic or industrial society. However, both communities are showing great interest in the enormous potential of the LF. The application of the LF in neurosurgery has been addressed in [120]. Besides depth estimation, post-capture aperture size adjustment, shape control, and 3D modeling LF promises

Figure 1.11: Commercial Plenoptic cameras (a) Pelican (b) Lytro (c) Raytrix (d) Adobe (e) Toshiba

post-capture focusing [121]. Once the focus plane of a conventional camera is set, it cannot be changed any longer. In contrary, the focus plane can be modified in LF which makes it very attractive for neurosurgery or teaching anatomy. The LF images have also shown promising benefits in face recognition, iris recognition, 3D Skincare, Otoscopy (measuring the 3D shape of the tympanic membrane in the ear), 3D plant image, observation of Biological cells through Biological Microscope, 3D X-ray reconstruction, and 3D endoscopy [122–125]. These are only some of the plentiful applications of the LF. There are many other non-medical applications such as fluid flow measurement, Metal surface scanning, PCB manufacturing, etc. It is very well-known in the literature that LF facilitates observation of *occluded* objects as opposed to any other imaging modality. For sure, incorporating the LF in your research will add a lot of values to your team and admits your group to the competitive club of LF pioneers. One example of the scientific and commercial successes of LF microscopy is Doitplenoptic S.L. which was launched by the 3D Imaging & Display Laboratory at the University of Valencia.

1.9 LF watermarking

The common 2D watermarking methods basically do not have any mechanism to preserve the tremendous angular information carried by LF images. Even if some 2D watermarking method is directly applicable to LF, there is a substantial risk of ruining the angular information of the LF. To the best of our knowledge, despite the numerous research papers addressing 2D image watermarking methods, the LF watermarking is seldom addressed [22,24,33,97,126,127]. Some few other works are done e.g. in 3D object watermarking [128] and free-view video watermarking [129, 130] which are not the same as LF watermarking. Any method proposed for the LF watermarking should take the tremendous spatial and angular information of the LF into account. The loss of LF angular information will have significant adverse effect on 3D reconstruction of the LF and may hinder the 3D perception of the observer. The importance of the watermarking method which is customized for LF will be discussed in more details in chapters 2 and 3.

Figure 1.12: Samples of LF images captured by Raytrix camera [119]

Chapter 2

DCT-based Watermarking

2.1 Overview

Over the last years, the LF has been increasingly used in various applications in medical imaging, manufacturing control, metrology, or even entertainment business. The high dimensionality and the tremendous amount of the data incorporated within the LF, demands some customized method for LF watermarking to fully protect the spatial and angular information simultaneously. To achieve this goal, a new method is proposed in this chapter which employs DCT and SVD to extract a robust watermarking feature. The usage of DCT and SVD is such that causes as less distortion as possible. The mathematical argument and experimental results both indicate the fidelity of the watermarked LF. This chapter elaborates the first watermarking method proposed specifically for LF. First, the definition, characteristics and mathematical formulation of DCT is presented. A section is dedicated to review the SVD. After covering DCT and SVD, the first proposed method for LF watermarking is discussed in details. The philosophy behind the proposed method and the mathematical methodology will be also discussed. The chapter then continues with extraction procedure. A separate section is dedicated to analyze the experimental results. This section begins by outlining the employed LF acquisition platform and necessary pre-processing steps. Afterwards, the experimental results of the proposed method are assessed both subjectively and objectively. Finally the robustness of the proposed method is examined against some common image processing attacks as Gaussian noise, JPEG compression, and median filtering.

2.2 Discrete Cosine Transform

The Discrete Cosine Transform (DCT) facilitates the conversion of a signal into frequency domain. The DCT basically converts the input signal into a mixture of weighted *basis functions*. The input and output of the DCT have identical dimension. Even though the decorrelation of DCT is generally inferior to Karhunen–Loève transform, its computational cost is significantly lower [131]. This feature, makes DCT very appealing for image/video compression standards and compression applications [132–136]. The DCT of an $M \times N$ matrix A is defined as

$$B_{ij} = DCT(A)_{ij} = \alpha_i \alpha_j \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos\left(\frac{\pi(2m+1)i}{2M}\right) \cos\left(\frac{\pi(2n+1)i}{2N}\right), \tag{2.1}$$

in which $\alpha_0 = 1/\sqrt{2}$ and $\alpha_k = 1$ for $k \neq 0$. Note also that DCT is invertible, in such a way that knowing the above DCT coefficients it is possible to recover the matrix A by means of the relationship

$$A_{mn} = DCT^{-1}(B)_{mn} = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \alpha_i \alpha_j B_{ij} \cos\left(\frac{\pi(2i+1)m}{2M}\right) \cos\left(\frac{\pi(2j+1)n}{2N}\right). \tag{2.2}$$

From a computational point of view, it is more convenient to model the DCT as a *pre-defined* combination of basis functions. The basis functions can be computed and stored for a given dimension. For any possible permutation of m, n, the terms $\cos\left(\frac{\pi(2m+1)i}{2M}\right) \cos\left(\frac{\pi(2n+1)i}{2N}\right)$ can be calculated once and stored. As these terms are not calculated each time, the storage of basis functions saves considerable amount of time. Figure 2.1 shows the

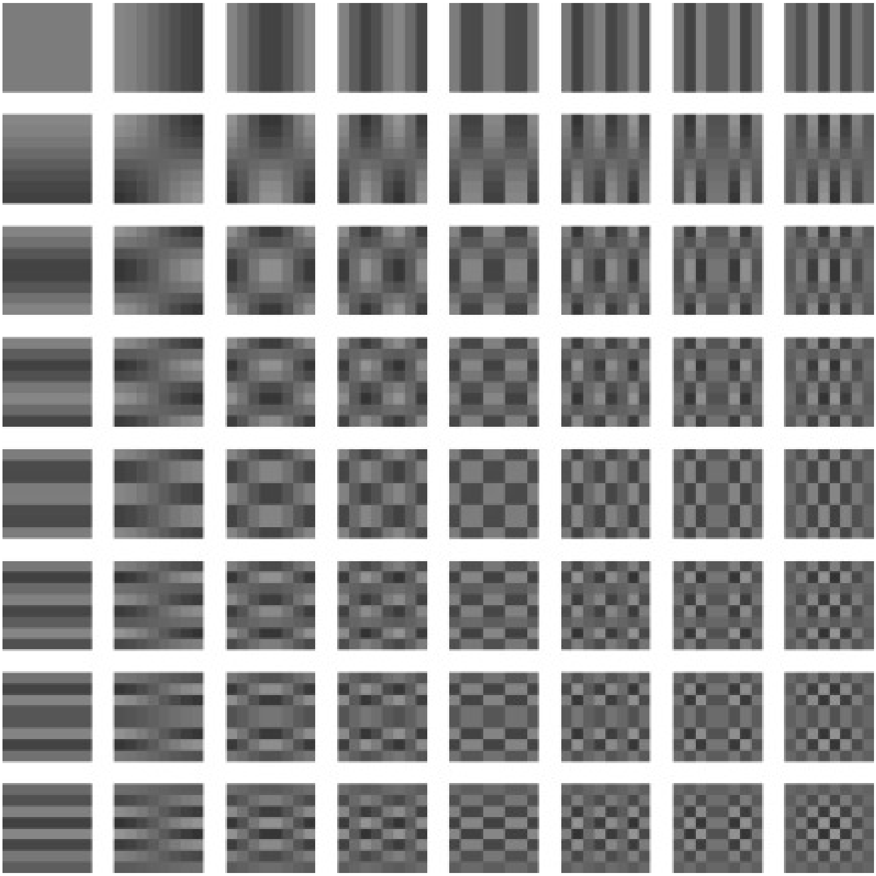


Figure 2.1: DCT basis functions for 8×8 *matrices. [137]*

DCT basis functions of 8×8 matrices. It is noticeable that the basis functions of Fig. 2.1 show increased variation as we go from the top-left matrix, corresponding to $m, n = 0$, to the bottom-right matrix, corresponding to $m, n = 8$. The top-left basis function has neither horizontal nor vertical variation and represents the DC or zero spatial frequency. Along the top row, the basis functions increase in horizontal spatial frequency content.

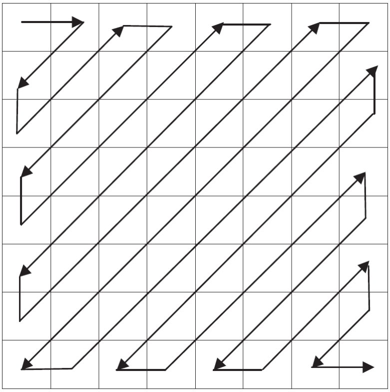


Figure 2.2: The zigzag order of energy distribution among DCT coefficients

Down the left column, the basis functions increase in vertical spatial frequency, with an increase in both horizontal and vertical frequencies along the diagonals.

It is common knowledge in the literature that DCT compacts most of the signal energy in a few coefficients [22, 132–134, 138]. Most of the energy of the images captured from natural scenes is usually concentrated in low frequency bands and distributed in zigzag order [132]. This concept has been visualized in Fig. 2.2. The coefficients lying in the top-left corner comprise the low-frequency content (e.g. the background) while the coefficients lying in the bottom-right corner form the high-frequency content (e.g. the lines and edges). The coefficients close to diagonal entries contribute to medium-frequency content. It is well-known in the literature that the low-frequency coefficients have by far much higher energy level than other ones [26, 138]. Such enormous compaction of coefficients energy provides huge robustness against potential attacks happening to the watermarked content [22]. On the other hand, the decorrelated DCT coefficients facilitate image reconstruction using much fewer coefficients than the number of image pixels. As a vast majority of the signal energy has been concentrated in low-frequency coefficients, using fewer transform coefficients for image reconstruction stands to reason and it usually causes minor invisible distortions [139].

Regarding the contribution of DCT coefficients to reconstruction, it is worth carrying out a simple test to illustrate the magnitude of DCT coefficients. Table2.1 shows an 8×8 block while Table 2.2 shows its DCT coefficients. The block has been reconstructed by first 3, 6, 10 and 64 coefficients and the results have been shown in Tables 2.3 to 2.6. As a metric for comparison of the original I and reconstructed R images, the Peak Signal to Noise Ratio ($PSNR$) is used, defined as

$$PSNR = 10\log_{10}\left(\frac{MAX^2}{MSE(I,R)}\right), \tag{2.3}$$

where MAX is maximum possible pixel value of the image and the Mean Square Error (MSE) is defined as

$$MSE(I,R) = \left(\frac{1}{M \times N \times n\_ch}\right)\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\sum_{p=0}^{n\_ch-1}\Big(I(m,n,p) - R(m,n,p)\Big)^2, \tag{2.4}$$

$M \times N \times n\_ch$ being the dimensions of both I and R.

As anticipated, if all the coefficients are employed in reconstruction, the reconstructed block will be identical to the original one and results in infinite $PSNR$. However, using less coefficients introduces negligible distortion and the maximum absolute difference of corresponding pixels between the reconstructed block and the initial one never exceeds 4.86. The average error of the corresponding pixels neither exceeds 1.61. Regarding the high dynamic range of the pixels (144), the average and maximum figures are 1.61% and 3.02% of maximum magnitude of the block. Hence, one may truely deduce that the difference caused by modifying the number of DCT coefficients is infinitesimal. In other words, discarding a high percentage of DCT coefficients not only introduces no significant distortion but also it widely contributes to boost the robustness of the watermarking platform against potential attacks. We will shortly address this effect in detail.

Table 2.1: Original block used as an example.

Original block							
142	142	141	140	142	136	143	140
142	142	141	140	142	136	143	140
142	141	141	140	142	136	143	140
142	142	141	140	142	136	143	140
142	141	141	140	142	136	143	140
144	144	137	135	140	140	138	140
140	140	143	138	139	143	139	137
138	138	135	136	138	139	136	138

Table 2.2: DCT coefficients of the block in Table2.1

DCT coefficients							
1120.375	4.388	4.231	1.076	0.375	-0.702	-5.135	6.156
6.605	0.625	0.46	-3.953	2.17	3.197	-5.66	4.648
-4.274	-1.017	-1.39	0.975	-0.975	-0.871	2.313	-2.274
2.562	1.694	1.77	1.25	-0.87	-1.377	0.406	-0.138
-1.625	-1.286	-0.466	-1.417	1.875	0.981	-2.106	0.658
1.924	0.633	-1.527	-0.248	-2.297	0.749	1.814	0.103
-2.153	0.081	2.813	1.815	1.892	-2.33	-0.859	-1.104
1.133	-0.722	-2.401	-1.606	-0.662	2.551	0.653	1.374

Table 2.3: Reconstructed block from the first 3 coefficients in Table2.2 (PSNR= 36.4874 dB)

Reconstructed block							
141.953	141.837	141.623	141.343	141.04	140.761	140.547	140.431
141.778	141.662	141.448	141.169	140.866	140.586	140.372	140.256
141.456	141.34	141.126	140.846	140.544	140.264	140.05	139.934
141.035	140.919	140.705	140.426	140.123	139.843	139.629	139.513
140.579	140.464	140.25	139.97	139.667	139.388	139.174	139.058
140.159	140.043	139.829	139.549	139.246	138.967	138.753	138.637
139.836	139.721	139.506	139.227	138.924	138.644	138.43	138.315
139.662	139.546	139.332	139.052	138.75	138.47	138.256	138.14

Table 2.4: Reconstructed block from the first 6 coefficients in Table2.2 (PSNR= 37.056 dB).)

Reconstructed block							
142.096	141.552	140.723	139.984	139.621	139.691	140.007	140.273
142.308	141.768	140.945	140.214	139.86	139.939	140.261	140.531
142.522	141.988	141.177	140.461	140.125	140.219	140.553	140.829
142.454	141.929	141.134	140.438	140.124	140.238	140.588	140.873
141.939	141.423	140.644	139.971	139.68	139.816	140.183	140.477
141.054	140.546	139.783	139.13	138.861	139.018	139.4	139.702
140.111	139.61	138.859	138.221	137.969	138.141	138.536	138.844
139.505	139.007	138.263	137.633	137.39	137.571	137.972	138.284

Table 2.5: Reconstructed block from the first 10 coefficients in Table2.2 (PSNR= 37.208 dB).

Reconstructed block							
142.505	141.74	140.74	140.104	140.045	140.342	140.66	140.827
142.371	141.598	140.579	139.912	139.808	140.054	140.328	140.468
142.39	141.612	140.576	139.871	139.708	139.883	140.09	140.19
142.612	141.844	140.818	140.106	139.911	140.034	140.187	140.253
142.558	141.824	140.849	140.183	140.012	140.133	140.268	140.319
141.692	141.009	140.119	139.547	139.451	139.619	139.776	139.834
140.173	139.543	138.743	138.273	138.271	138.507	138.705	138.781
138.951	138.354	137.612	137.209	137.27	137.555	137.785	137.875

Table 2.6: Reconstructed block from all 64 coefficients in Table2.2 (PSNR= ∞)

Reconstructed block							
142	142	141	140	142	136	143	140
142	142	141	140	142	136	143	140
142	141	141	140	142	136	143	140
142	142	141	140	142	136	143	140
142	141	141	140	142	136	143	140
144	144	137	135	140	140	138	140
140	140	143	138	139	143	139	137
138	138	135	136	138	139	136	138

2.3 Singular Value Decomposition (SVD)

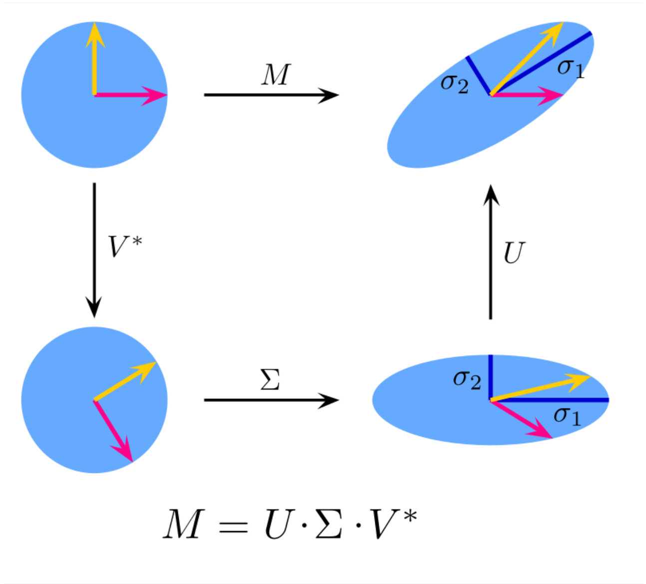


Figure 2.3: The geometrical interpretation of SVD. [140]

In linear algebra, SVD is defined as factorizing of a matrix to three ones. In contrary to eigen value decomposition, SVD is applicable to both square and non-square matrices. In fact, SVD finds other orthonormal basis vectors and represents any given matrix in a new space. In other words, SVD finds some basis vectors (orthogonal to each other) such that if the new basis vectors are multiplied by the given matrix, the yielded matrices remain prependicular. Figure 2.3 shows two prependicular vectors (0,1) and (1,0) on the top left. Then these two vectors are rotated through V^T. The rotated matrix is shown on bottom left. Afterwards, the rotated vectors are scaled unequally. The amount of scaling each vector is dictated by the singular values. Finally, the scaled vectors are rotated by calculating production of the scaled vectors and matrix U. This example is confined to 2D space which is easier to visualize. The same argument can be extended to higher dimensionalities. Each $m \times n$ matrix can be factorized as

$$M_{m\times n} = \underbrace{\begin{bmatrix}\vec{u}_{1(m\times 1)} & \vec{u}_{2(m\times 1)} & \dots & \vec{u}_{m(m\times 1)}\end{bmatrix}}_{U_{m\times m}} \underbrace{\begin{bmatrix}\sigma_1 & 0 & \dots & 0\\ 0 & \sigma_2 & \ddots & 0\\ \vdots & & & \\ 0 & 0 & \dots & \sigma_n\\ \vdots & & & \\ 0 & 0 & \dots & 0\end{bmatrix}}_{\Sigma_{m\times n}} \underbrace{\begin{bmatrix}\vec{v}_{1(1\times m)}\\ \vec{v}_{2(1\times m)}\\ \vdots\\ \vec{v}_{n(1\times m)}\end{bmatrix}}_{V_{n\times n}} . \tag{2.5}$$

Both U and V are unity matrices, i.e.

$$\begin{aligned} UU^T &= I\,, \\ V^TV &= I\,. \end{aligned} \tag{2.6}$$

Figure 2.4 visualizes the matrix factorization by SVD. It can be easily shown that for each matrix A, the singular values are in fact the eigen values of AA^T [141]. It is noticeable that the entries of Σ are arranged in descending order and the matrix M can be reconstructed with high accuracy even if only few largest singular values are exploited [142].

2.4 Embedding Procedure

As mentioned in section1.3 the watermarking platform should meet triple requirements, namely, transparency, robustness and capacity. The purpose of this section is to develop a method capable of LF watermarking to address such concerns. Besides the aforementioned requirements, LF watermarking specifically has to preserve the angular information. This regarding, even if a watermarking method of conventional images is applicable to LF, there is absolutely no guarantee that the angular information of the LF will be preserved.

Figure 2.5 shows the block diagram of the proposed method of watermark insertion. The proposal includes the following steps:

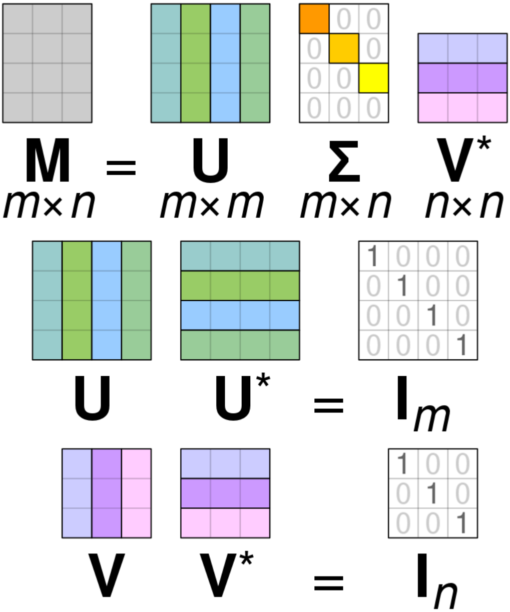


Figure 2.4: Visualization of matrix factorization by SVD. [140]

- The first step is to arrange a set of blocks from the registered integral image for carrying the watermark. For each pixel of the watermark, some pixels from specific μIs are chosen. The exact location of the selected pixels depends on the secret key. It may seem plausible to embed the watermark bits in individual μIs. We have implemented this approach in [24]. As shown in Fig. 2.6, another possible permutation of pixels forming the blocks may be selecting the block entries from different μIs. This approach has been used in [22]. For now, suppose the arranged block to carry the ith row and jth column pixel of the watermark is denoted as $img\_blk\_sel_{ij}$.

- Next, the DCT of each of these blocks is performed, following the definitions in section2.2. The corresponding coefficients are arranged

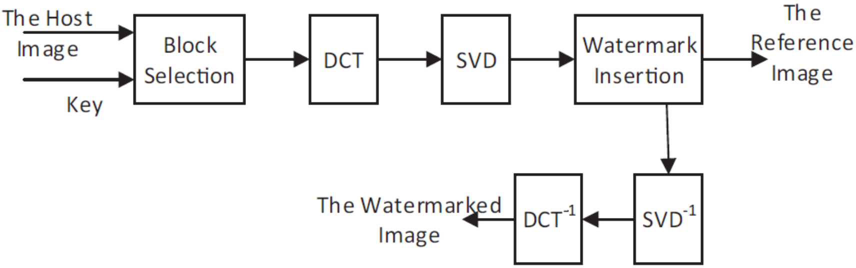


Figure 2.5: The block diagram of DCT-based watermarking method.

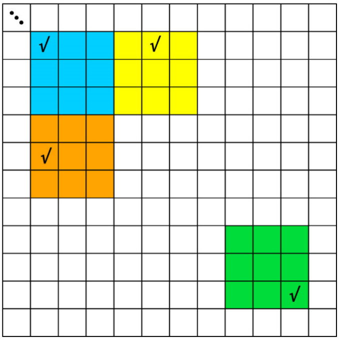


Figure 2.6: A possible selection of pixels of µIs to make watermarking blocks. Each color square represents a µI.

then as a column vector, namely

$$dct\_img\_blk\_sel_{ij} = zigzag[DCT(img\_blk\_sel_{ij})] \tag{2.7}$$

where the *zigzag* function of an $M \times N$ matrix A is defined as

$$zigzag[A] = [a_{00}, a_{01}, a_{10}, a_{20}, a_{11}, a_{12}, \ldots, a_{M-1,N-1}]^T\,. \tag{2.8}$$

Note that eq. 2.8 is the mathematical representation of ordering procedure in Fig. 2.2.

- Then, only the first n_dct coefficients are selected from the previous vector, and this truncated version of $dct\_img\_blk\_sel_{ij}$ is factorized by SVD, as stated in section 2.3. It is straightforward to show that only one singular value σ_{ij} is obtained in this case, being equal to the magnitude of the above truncated vector.

- Now, to obtain the watermarked block, this singular value is altered as
$$\sigma_{wm\_ij} = \begin{cases} \sigma_{ij} + gf\ , & wm\_bit_{ij} = 1 \\ \sigma_{ij} - gf\ , & wm\_bit_{ij} = 0\ . \end{cases} \tag{2.9}$$
where gf is the watermark strength. As will be discussed soon, this singular values will be required to extract the embedded watermark. Consequently, σ_{ij} will be recorded as the ith row and jth column of the *reference image* (ref_img_{ij}).

- Next, the inverse SVD of this watermarked coefficients vector is performed. Its components are used to substitute the first n_dct ones in eq. 2.8. After applying an inverse *zigzag* transform, we obtain the DCT coefficients of the watermarked block $wm\_img\_blk\_dct_{ij}$.

- Afterwards, the watermarked block is then obtained as
$$wm\_img\_blk_{ij} = DCT^{-1}(wm\_img\_blk\_dct_{ij})\ . \tag{2.10}$$

- Finally, the watermarked blocks will be mapped back to the chosen locations specified by the secret key.

2.5 Extraction Procedure

In order to extract the embedded watermark, the same secret key used in the embedding procedure is required. If the same secret key is not available, then the location of the blocks carrying the watermark cannot be specified and hence the embedded watermark cannot be extracted. This measure, will improve the security. Even if the third party somehow finds out the precise embedding and extraction procedure, he still will be unable to extract the embedded watermark without the secret key. As shown in Fig. 2.7,

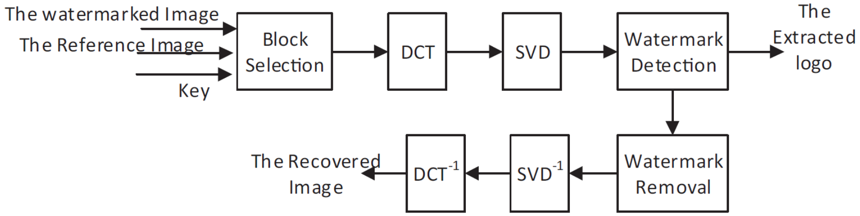


Figure 2.7: The extraction procedure of DCT-based watermarking

the watermarked and the reference images both are the other inputs of the extraction system along with the secret key.

The block selection and DCT are performed exactly in the same fashion detailed in section 2.4 . To extract the watermark bit of ith row and jth column, wm_ext_{ij}, the singular value of the first n_dct coefficients of the corresponding block (σ_ext_{ij}) is compared with ref_img_{ij} by using the following decision function

$$wm\_ext_{ij} = \begin{cases} 1 & \sigma\_ext_{ij} > ref\_img_{ij} \ , \\ 0 & \sigma\_ext_{ij} < ref\_img_{ij} \ . \end{cases} \tag{2.11}$$

The host LF may then be recovered by removing the extracted watermark. To remove the watermark, we apply the following rule

$$\sigma\_rec_{ij} = \begin{cases} \sigma\_ext_{ij} - gf & wm\_ext_{ij} = 1 \ , \\ \sigma\_ext_{ij} + gf & wm\_ext_{ij} = 0 \ . \end{cases} \tag{2.12}$$

Afterwards, the inverse SVD and DCT are applied and the recovered LF will be obtained. It is worth mentioning that U and V^T of the host LF are different with those of the watermarked LF. That is to say, after modification of largest singular value of the corresponding block of the host LF (eq. 2.9), if the same block is factorized by SVD, the singular values (Σ) and the left/right singular vectors may slightly be different. That is why the recovered host LF is not exactly identical to the host LF and the *PSNR* of the recovered LF is not infinite. The bigger the gf, the difference between the SVD components will be bigger as well.

2.6 Experimental Results

2.6.1 Capturing Setup

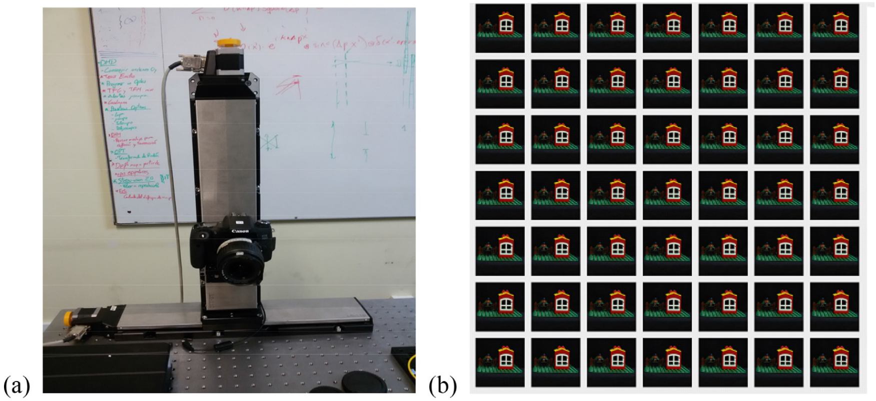


Figure 2.8: (a) The experimental setup to capture LF (b) Central 7×7 EIs of the 3D scene.

In order to assess the proposed method subjectively and objectively, some experiments are designed. The experimental setup to capture the LF has been shown in Fig. 2.8(a). In this setup, a digital camera (Canon 450D) is mounted on a rail and a computer controls the camera lateral position precisely. Sliding the camera horizontally and vertically, the EIs are captured. The scene is placed at an approximate axial distance of 73 cm from the camera. The vertical and horizontal displacement steps of the camera has set equal to 5 mm in both directions. 16×16 EIs have been captured each having 300×300 pixels. Consequently, this integral image has 4800×4800 pixels. The central 7×7 array of this setup has been shown in Fig. 2.8(b). Afterwards, the acquired array of EIs has been transposed following the method in [143] and 300×300 μIs have been obtained each having 16×16 pixels.

2.6.2 Assessment Criteria

To assess the performance of the proposed method objectively, some objective metrics have been utilized. The first one is *PSNR* as introduced in eq. 2.3, but properly generalized to assess the captured 5D LF (4D plus RGB channel number), defined as [22]

$$PSNR = 10\log_{10}\left(\frac{MAX^2}{MSE(I,I_W)}\right), \tag{2.13}$$

where *MAX* is the maximum possible value of the pixel intensity and *MSE* is

$$MSE(I,I_W) = \left(\frac{1}{n_{EI,h}\, n_{EI,v}\, n_{\mu I,h}\, n_{\mu I,v}\, n_{ch}}\right)\times \sum_{s=0}^{n_{EI,h}-1}\sum_{t=0}^{n_{EI,v}-1}\sum_{u=0}^{n_{\mu I,h}-1}\sum_{v=0}^{n_{\mu I,v}-1}\sum_{ch=0}^{n_{ch}-1}\Big(I(s,t,u,v,ch)-I_W(s,t,u,v,ch)\Big)^2 \tag{2.14}$$

in which $n_{EI,h}$ and $n_{EI,v}$ are the number of rows and columns in each μI, $n_{\mu I,h}$ and $n_{\mu I,v}$ are the number of μIs in horizontal and vertical directions, and n_ch is the number of the chromatic channels respectively. Equation (2.14) is a generalization of the definition in eq. 2.4 and $I(:)$ and $I_W(:)$ are the host and watermarked LF. The "W" subscript is used to refer to the watermarked LF while letter with no subscript represents the host LF.

The fidelity of the extracted watermark to the embedded one is measured by the Bit Error Rate (BER), defined as

$$BER = \frac{\sum_{i=0}^{N_b-1}\sum_{j=0}^{N_b-1}(wm\_bit_{ij}\oplus wm\_ext_{ij})}{N_b^2}, \tag{2.15}$$

where the wm_bit and wm_ext stand for the embedded and extracted watermark bits, respectively, and $\oplus$ is the exclusive "OR" operator (XOR). Even though there is no compulsion to assume the number of the rows and columns of the watermark are the same as N_b, making such assumption is only for the sake of notation convenience and obviously, the watermark may have arbitrary number of rows and columns.

If all the watermark bits are extracted correctly, the *BER* will be zero which signifies perfect watermark extraction. Conversely, if all the watermark bits are extracted falsely, the *BER* will equal 1. Nevertheless, provided that $BER = 1$, the extracted watermark can turn to the embedded one by simply flipping every single bit. The hardware realization of this task is just as simple as passing the extracted watermark through a logical *"NOT"* gate.

Despite the popularity of *PSNR* for measuring the transparency of the watermarking schemes, it comes with its own imperfection. The relationship between actual *perception* by the HVS and *PSNR* has been investigated earlier indicating that the *PSNR* of a severely degraded image may be high while the visual distortion seems very dominant to HVS. On the other hand, some other images with minor degradation may have a misleading low *PSNR*. To deal with this challenge, Weng. et. al. introduced the Mean Structural SIMilarity (*MSSIM*) [144]. *MSSIM* addresses the HVS mechanism of image perception and provides an objective metric much closer to HVS. The basic premise of *MSSIM* is that HVS perception of the image quality is vastly affected by structural similarity of the image content rather than the absolute values of the pixels. It also incorporates the average intensity and contrast of the image which both play a key role in HVS perception of the image quality. Table 2.7 summarizes the parameters used to derive *MSSIM*.

Table 2.7: Required parameters to derive MSSIM

μ_I	The average intensity of the host LF
μ_W	The average intensity of the watermarked LF
σ_I	Standard deviation of the host LF
σ_W	Standard deviation of the watermarked LF
σ_{I,I_w}	Covariance of the host image and the watermarked one
C_1, C_2 *and* C_3	Stabilizing constants

MSSIM models the luminance similarity as

$$l(I, I_W) = \frac{2\mu_I \mu_{I_W} + C_1}{\mu_I^2 + \mu_{I_W}^2 + C_1} . \tag{2.16}$$

Another component of *MSSIM* is contrast similarity, that is assessed by means of the function

$$C(I,I_W)=\frac{2\sigma_I\sigma_{I_W}+C_2}{\sigma_I^2+\sigma_{I_W}^2+C_2}\ . \tag{2.17}$$

Additionally, the comparison of structural similarity is achieved by

$$S(I,I_W)=\frac{\sigma_{I,I_w}+C_3}{\sigma_I\sigma_{I_W}+C_3}\ . \tag{2.18}$$

Regarding the impact of luminance, contrast and structural similarity, the quality metric is stated as

$$Similarity(I,I_W)=[l(I,I_W)]^{\alpha}[C(I,I_W)]^{\beta}[S(I,I_W)]^{\gamma}\ , \tag{2.19}$$

where $\alpha,\beta,\gamma>0$ adjust the contribution of luminance, contrast and similarity, respectively. By setting $\alpha=\beta=\gamma=1$ and $C_3=C_2/2$, *MSSIM* is defined as in [144]

$$MSSIM(I,I_W)=\frac{(2\mu_I\mu_{I_W}+C_1)(2\sigma_{I,I_W}+C_2)}{(\mu_I^2+\mu_{I_W}^2+C_1)(\sigma_I^2+\sigma_{I_W}^2+C_2)}. \tag{2.20}$$

Letting C_1 and C_2 zero will turn *MSSIM* to Universal Quality Index (UQI) [145].

2.6.3 Objective and Subjective Assessment

The embedded watermark that we use has 8 rows and 8 columns and each watermark bit is embedded in a 16 × 16 block of pixels of the host LF. The embedded watermark is generated by random distribution of the watermark bits. The randomness of the watermark bits guarantees that the proposed method is not biased toward any specific watermark and will work for any given watermark. Figure 2.9(a) shows the used watermark. As mentioned in section 2.6.2, the assumption of the equal number of rows and columns of the watermark is only for more convenience and the watermark may have different number of rows and columns. The proposed method does

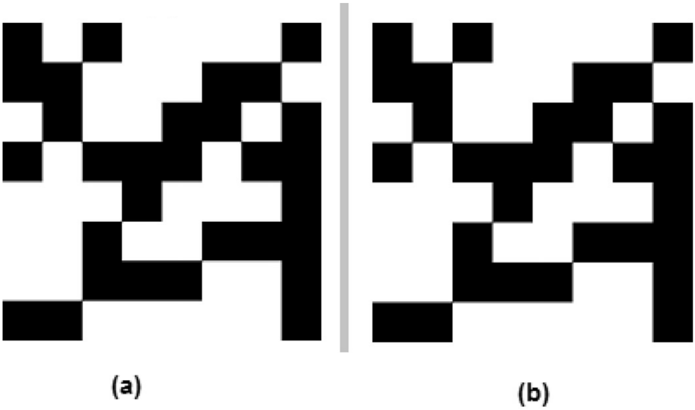


Figure 2.9: (a) The embedded 8×8 *watermark. (b) The extracted watermark*

not impose any limitation on the number of rows and columns of the blocks either. For watermark insertion, we use $gf = 90$ and 3 DCT coefficients. Figure 2.10 shows the central *EI* of both the host and watermarked LF. As is obvious from Fig. 2.10, the host and watermarked LF seem indistinguishable. To ensure the fidelity of the proposed method, the same *EI* of both host and watermarked LFs are zoomed in exactly at the same region. Apparently, the magnified regions can not be discriminated. Throughout this chapter, we will always visualize the same *EI* unless otherwise stated. Figure 2.9 shows the identity of the embedded and extracted watermark. As opposed to many watermarking methods proposed in the literature, our approach promises error-free watermark extraction regardless of the watermark strength.

Figure 2.11 shows *PSNR*, *BER* and *MSSIM* of the watermarked LF. To investigate the impact of the number of DCT coefficients, the simulations have been conducted using 1, 3, 6 and 10 coefficients. A very interesting question that may rise, is the contribution of DCT in the embedding and extraction mechanism (Fig. 2.6 and Fig. 2.7). To address this question, the simulations have been carried out excluding the DCT and the results have been identified as ”SVD method” in all the graphs. Accordingly, hereafter we will refer to this approach as SVD method. In other words, the SVD is

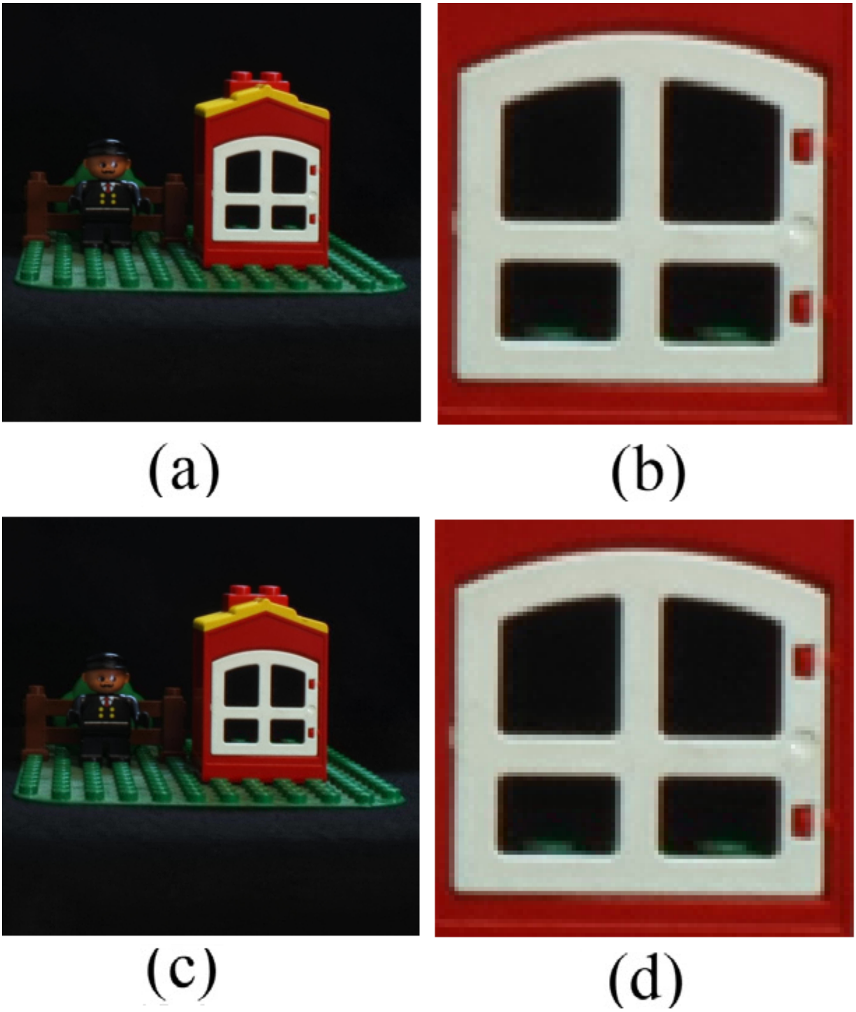


Figure 2.10: (a) The central elemental view of host LF (b) The watermarked LF (c) and (d) The zoomed in region of (a) and (b)

directly applied to the block carrying watermark bit. The other curves illustrate the proposed method with different number of the DCT coefficients.

As illustrated in Fig. 2.11(a), even for $gf = 240$ the $PSNR$ of the watermarked LF will remain higher than 60 dB in both methods. For $gf = 80$, we will have $PSNR = 70.295$ dB for the proposed method, which is way too far to be distinguishable from the host LF.

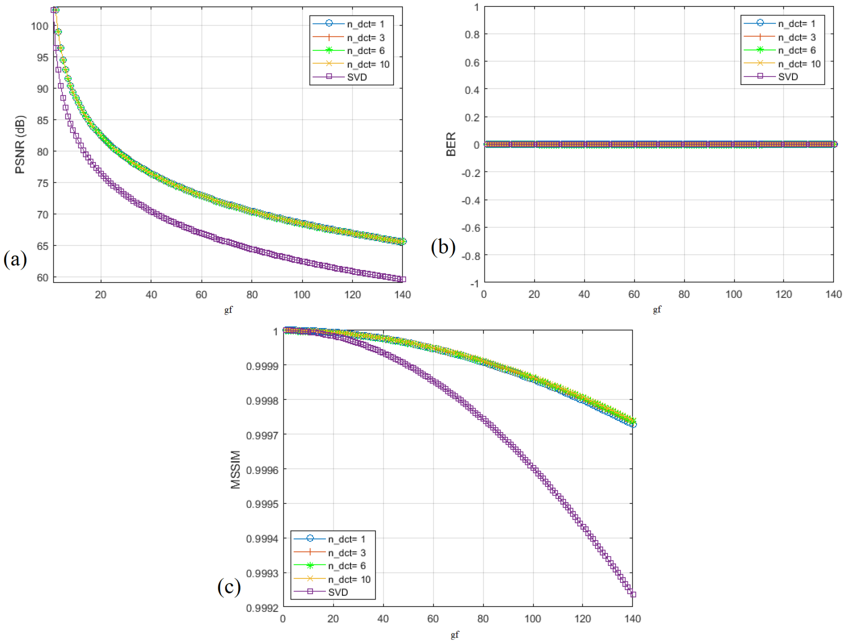


Figure 2.11: (a) PSNR, (b) BER, and (c) MSSIM of the watermarked LF.

Unlike plenty of watermarking methods proposed in the literature, it can convincingly be seen from Fig. 2.11(b) that no matter how many DCT coefficients are used to embed the watermark, the BER will always remain zero. It means that if the watermarked LF, is not attacked, the embedded watermark can be perfectly extracted even for the lowest figures of gf.

The $MSSIM$ of the watermarked LF has been shown in Fig. 2.11(c). As is noticeable from this figure, the $MSSIM$ is slightly lower for $n\_dct = 1$. However, the ordinate of Fig. 2.11(c) has been extremely exaggerated and emphasized to reflect the most minuscule differences among the $MSSIM$

figures and all the curves indicate values higher than 0.999. The absolute difference between *MSSIM* figures of any method (or the *MSSIM* figures of the same method with different gf values) never exceeds 0.001. Considering the ideal value of +1, the achieved values for *MSSIM* are very encouraging.

2.6.4 Robustness Analysis

Gaussian Noise

As any other signal, the watermarked images are also subject to noise attack. Delivering acceptable robustness against common image processing attacks is of paramount importance for the watermarking platform. One of the most common attacks imposed to the watermarked images is additive Gaussian noise. Figure 2.12 shows the effect of Gaussian noise on the watermarked images. Gaussian noise is a very common model for noise and is based on assumption that the statistical distribution of the noise is Gaussian with zero mean and the variance of σ_n^2. The images correspond to the central $EI(8,8)$ from the LF. As it is obvious from Fig. 2.12, noise powers of 625 and 1225 are very aggressive and can heavily degrade the visual quality of the LF. The latter case is so destructive that some color dots overwhelm the picture.

The *BER* of the noisy watermarked LF has been calculated for different cases of $n\_dct \in \{1,\ 3,\ 6\ ,10\}$ and SVD method. Figure 2.13 shows *BER* of watermark extraction of the noisy LF. Even though the host LF may dominantly be degraded by noise, the watermark can still be extracted. As it is obvious from Fig. 2.13, if $\sigma_n^2 = 100$, then the number of DCT coefficients seems having no noticeable impact on *BER*. Additionally, even though the performance of the SVD method is slightly inferior to the proposed method, the difference may rationally be tolerated and it is safe to say that the SVD method has more or less the same performance as the proposed method.

Once the noise power rises to $\sigma_n^2 = 225$, then the number of DCT coefficients start contributing to the *BER* of the noisy watermarked LF for small values of gf. However, even with the noise power of $\sigma_n^2 = 225$ the superiority of the proposed method over SVD method is evident. Anyway,

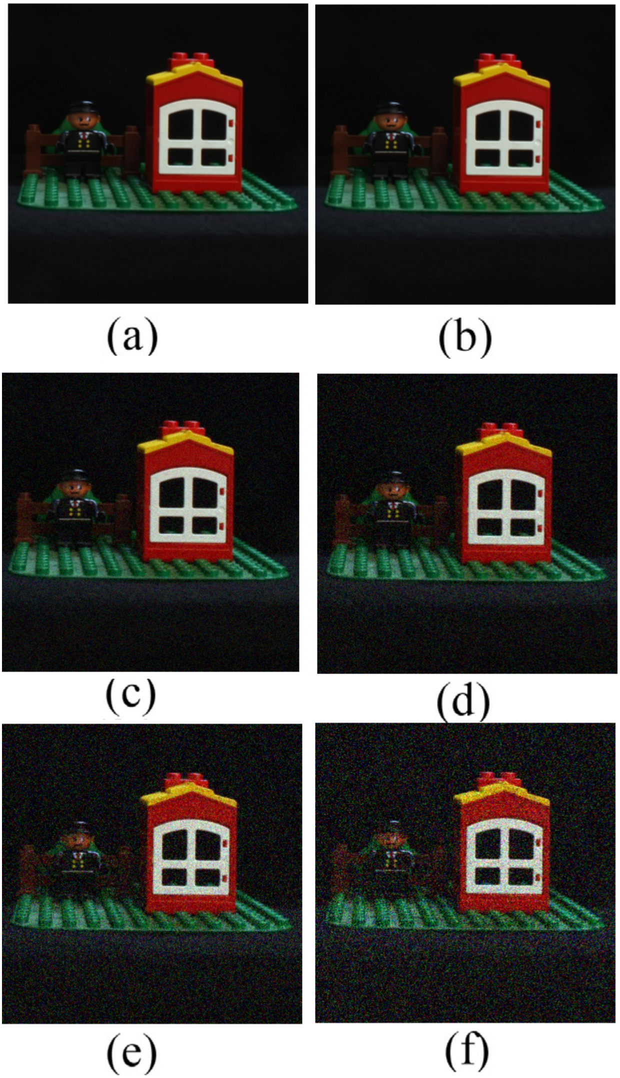
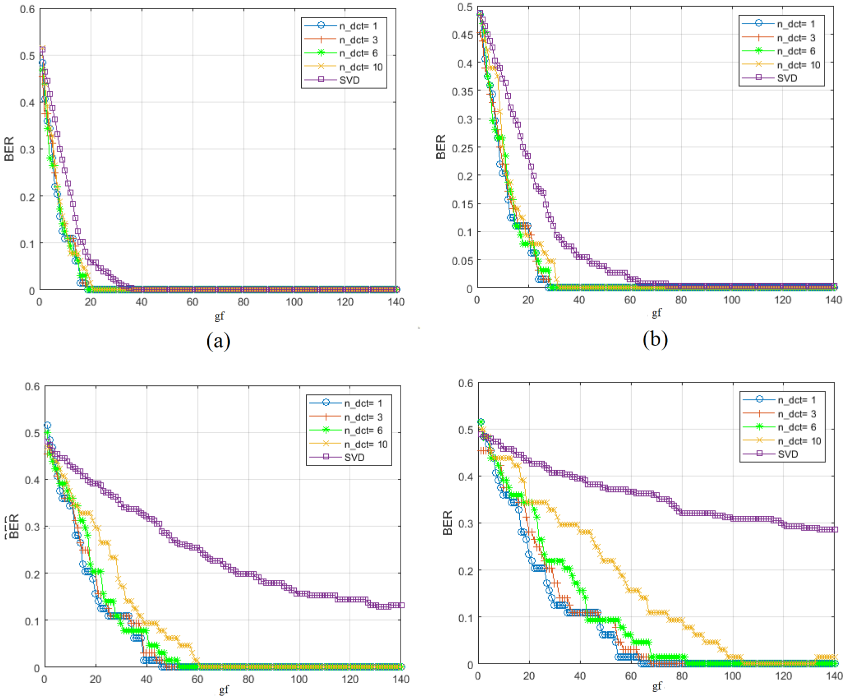


Figure 2.12: *Noisy watermarked LF. (a) The host LF; (b) The watermarked image (gf = 80 and n_dct = 3); (c) The watermarked image exposed to Gaussian noise of* $\sigma_n^2 = 100$ *; (d)* $\sigma_n^2 = 225$*; (e)* $\sigma_n^2 = 625$ *; and (f)* $\sigma_n^2 = 1225$.

the *BER* of the SVD method may still converge to zero.



(a) (b) (c) (d)

Figure 2.13: The BER of watermarked LF exposed to Gaussian noise of (a) $\sigma^2 = 100$ *; (b)* $\sigma^2 = 225$*; (c)* $\sigma^2 = 625$ *; and (d)* $\sigma^2 = 1225$.

In case of more intense noise attacks, namely $\sigma_n^2 = 625$ *and* 1225, not only the importance of the number of DCT coefficients is stressed, but also the absolute supremacy of the proposed method over SVD becomes much more sensible. Even with extreme figure of $gf = 140$, if the noise power increases to $\sigma_n^2 = 625$, the SVD method can never reach any *BER* less than 13.28%. On the other hand, using 10 DCT coefficients heavily deteriorates the robustness of the proposed method and causes the *BER* converging to zero for $gf = 61$. These arguments also concretely hold for $\sigma_n^2 = 1225$. If an extreme attack of Gaussian noise occurs, even for $gf = 140$, the *BER* of SVD method never falls below 28.51%. In contrast, the *BER* of the proposed method falls much faster. Letting $n\_dct = 3$, for $gf > 67$ the *BER* will be less than 1% and for $gf > 87$ we have $BER \approx 0$. Conversely, for $n\_dct = 10$, the *BER* is generally higher than that for $n\_dct = 3$ and the *BER* converges to zero only if $gf > 103$.

These results corroborate the hypothyesis of discarding noise-prone DCT coefficients. As most of the signal energy is concentrated in low frequency coefficients of DCT, they provide more robustness against Gaussian noise [138]. Hence, embedding the watermark in such coefficients will substantially increase the robustness of the proposed method against Gaussian noise. As mentioned earlier in section 2.4, such low frequency coefficients mainly lie on the top left part of DCT block (Fig. 2.2 and Table 2.2). Compared to other DCT coefficients (specially the ones lying on the bottom right region carrying high frequency content), the low frequency coefficients have overwhelmingly larger amplitude and are the best candidate to obtain robust watermarking feature. The more DCT coefficients used for watermark insertion, the more noise-prone coefficients are involved causing degradation of the robustness of the embedded watermark against Gaussian noise. As anticipated, the SVD method exhibits a poor performance against Gaussian noise. This can be fully justified by the fact that SVD method literally uses all 256 coefficients to acquire the watermarking feature and on top of that, using high frequency components will make it excessively vulnerable against Gaussian noise. Facing with more agressive noise attacks, the adverse effect of high frequency coefficients will be more noticeble in robustness of the proposed method.

JPEG Compression

Another common attack to the watermarked images that may occur very frequently is JPEG compression. In fact, JPEG compression is very common and it is widely used in smart phones, TVs, computers and tablets. Hence, the robustness of the watermarking method against JPEG compression is critical to recover the embedded watermark from various platforms using this kind of compression. Figure 2.14 shows the impact of JPEG compression on watermarked LF. Despite some rare lossless distributions

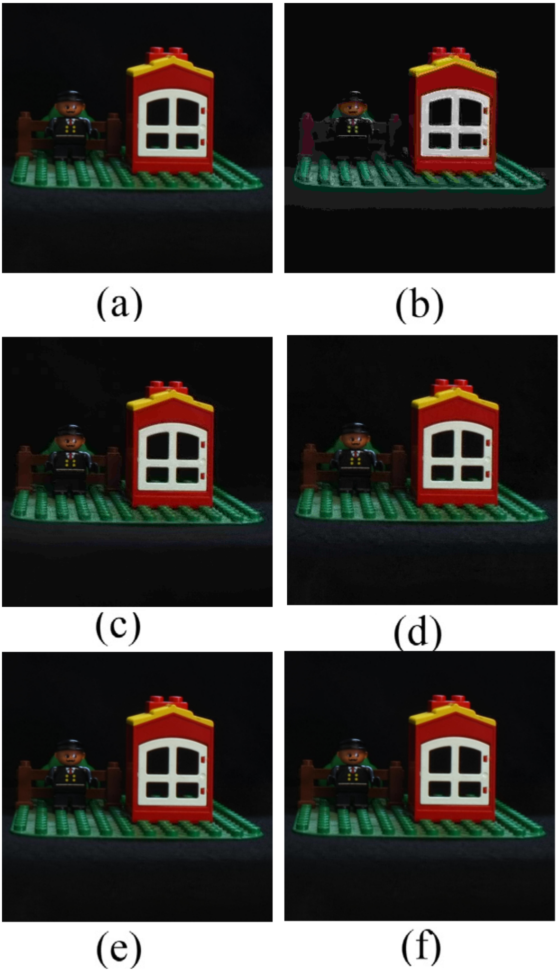
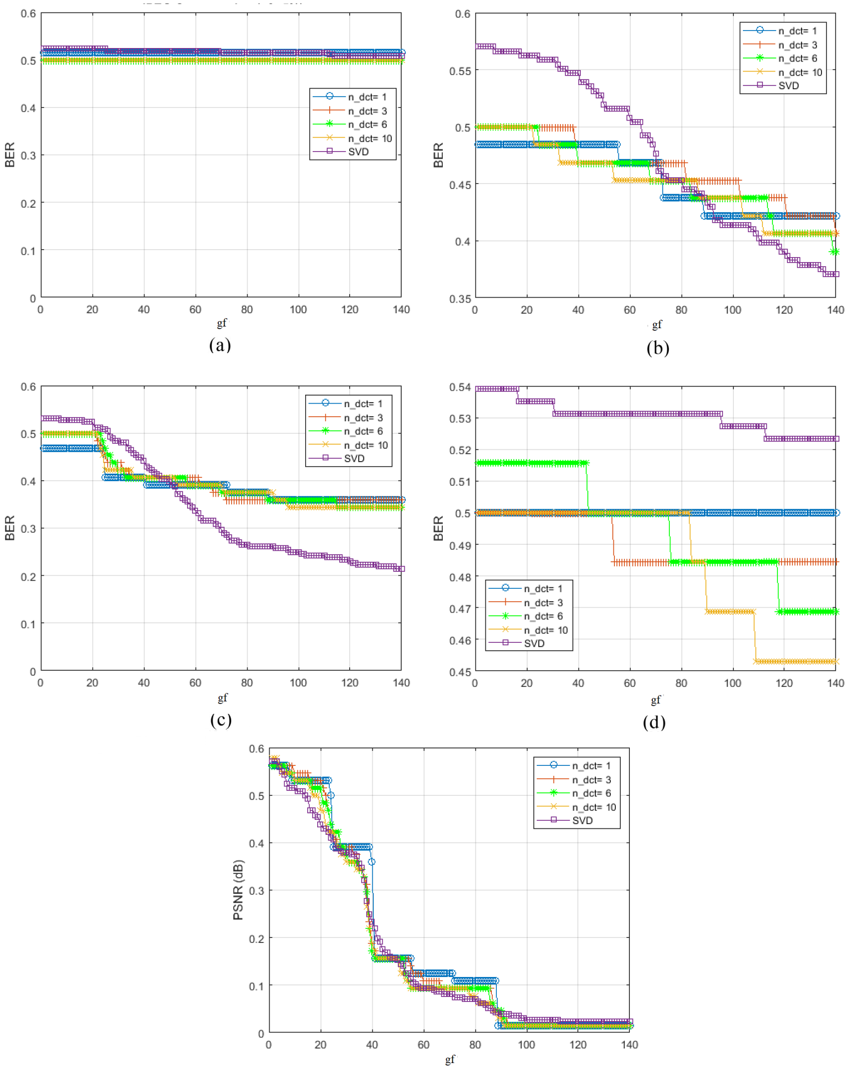


Figure 2.14: The watermarked LF with JPEG compression. (a) The original watermarked LF; after compresion with: (b) $qf = 5\%$; (c) $qf = 25\%$; (d) $qf = 50\%$; (e) $qf = 75\%$; and (f) $qf = 100\%$.



(a) (b) (c) (d) (e)

Figure 2.15: BER of watermarked LF compressed with JPEG: (a) $qf = 5\%$;(b) $qf = 25\%$;(c) $qf = 50\%$;(d) $qf = 75\%$;(e) $qf = 100\%$.

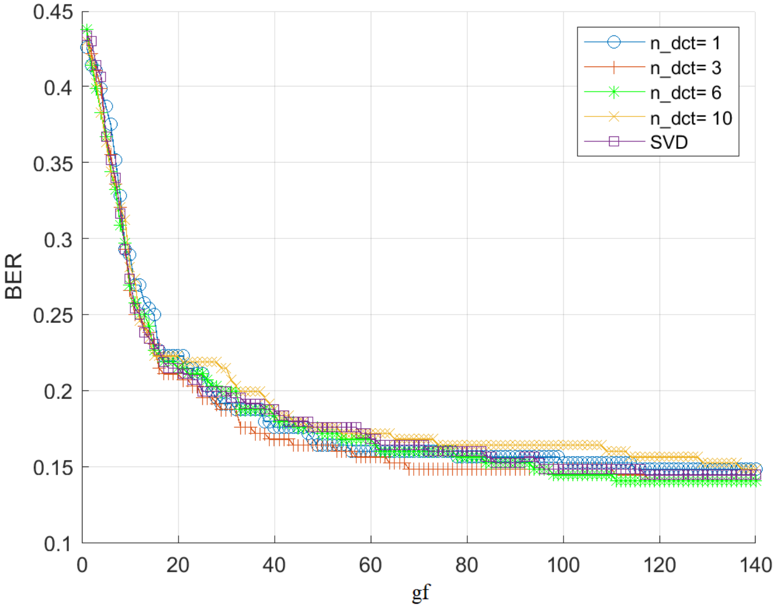


Figure 2.16: The BER of the watermarked LF passed through median filter.

of the JPEG, it is often known as a lossy compression scheme. There is a compromise between the quality and size of the compressed image. The quality factor (qf) is used to control the amount of compression and adjust the balance between the size and quality. The quality factor lies in the range $1\% - 100\%$. Unlike the lossless image compression methods as PNG, BMP, and PPM, JPEG always introduces some loss in the quality of compressed image. It is worth mentioning that even if the highest possible qf is selected, there is still some loss of quality and some reduction of the file size. The qf is reversely proportional to the quality of the compressed image. Hence, the higher the quality factor the less artifacts will appear. For example, using $qf = 100\%$ usually does not yield visible artifacts. [132, 146].

As it is evident from Fig. 2.14, the JPEG compression with $qf = 5\%$

causes an excessively severe degradation such that it is clearly noticeable. For example, the vertical columns of the fence or the regions near the man have been completely deformed. Practically, JPEG compression is very unlikely to be used with $qf = 5\%$ as it heavily ruins the image visual appearance. As can easily be seen from Fig. 2.15(b and c) the $qf = 25\%$ and 50% impose much less distortion and the main skeleton of the LF is considerably preserved. However, the sharp viwer may find some noticeable distortion. In many cases the $qf = 75\%$ may deliver such a good quality that even the sharp viewer will be unable to distinguish any (possible) difference. Finally, as expected the JPEG compression with $qf = 100\%$ provides the highest possible quality which may be achieved by JPEG. It is noticeable that even for $qf = 100\%$, the file size will be decreased and there will be some loss of data (even though less than lower qf figures). Figure 2.15 shows BER graphs for watermarked LF compressed by JPEG. Unsurprisingly, for the extreme case of $qf = 5\%$ the BER mostly lies higher than 50% which basically means the extracted watermark has a greatly random nature. For less agressive qf figures (namely $25\%, 50\%, 75\%$ and 100%), no significant difference was observed in terms of number of DCT coefficients. It is safe to say that both the proposed method and SVD approaches follow the same trajectory in terms of robustness against JPEG compression. This is fully consistent with the fact that $qf = 100\%$ implies minor compression and highly preserves the quality of the watermarked LF. As discussed earlier in section 2.6.3, if no attack occurs, the proposed method and SVD method both deliver error-free extraction of the embedded watermark. The same trend is observed for when the JPEG compression is done using a high qf. As anticipated, the minor compression loss caused by a high qf yields lower BER in comparison with lower qf figures. Nevertheless, JPEG compression (even with the highest qualities) alters the intensity of the pixels and challenges the extraction of the embedded watermark. However, given that the watermark strength is high enough, the embedded watermark can simply survive JPEG high quality compression.

Median Filtering

Another common attack which is quite likely to happen, is median filtering. A 3×3 window was used for median filtering. Figure 2.16 shows the BER of watermarked extraction under median filtering effect. Again, the proposed method slightly outperforms the SVD method. Unlike the Gaussian noise, it seems that the more DCT coefficients used, the higher robustness will be obtained against median filtering. Even though using three DCT coefficients does not cause significant difference in robustness of the proposed method against median filtering, but it makes sense to end up this result. Some areas of the LF (e.g. the edges) may carry high frequency information and such features are better protected if six DCT coefficients are used. Nevertheless, using ten DCT coefficients seems to deteriorate the robustness of the proposed method against median filtering. It can be deduced that using too few (one) or too many (ten) DCT coefficients lowers the robustness of the proposed method against median filtering. Using three DCT coefficients yields better results. If six coefficients are used, the BER will decrease even further by 1.57%.

Chapter 3

Joint DCT and Wavelet Watermarking

3.1 DCT vs DWT

The DCT is widely utilized in image watermarking [42, 44, 147], image compression [132, 133, 135] and video coding [136, 148, 149]. As the main scope of this dissertation concerns LF watermarking, the details of such compression schemes are off topic. The main reason of addressing DCT and DWT in compression application is to express the motivation behind using wavelet for LF watermarking. Meticulous details of compression standards may be found on [132, 136, 148, 150, 151]. It is utterly true that the low computational cost of DCT has made it very popular. However it is no secret that the DCT may introduce some artifacts [139, 152]. The basic premise of DCT-based compression methods is to divide the whole image into non-overlapping blocks and compress each block independently and this is exactly what causes blocking artifact in JPEG [146].

In complete agreement with section 2.4 and Table 2.1, it is very well grounded in the literature that the nearing pixels are substantially correlated even if they do not lie in the same block. Figure 3.1 shows blockwise division of the image pixels. The encircled pixels laying in different blocks (identified with different colors) will be encoded independently and the

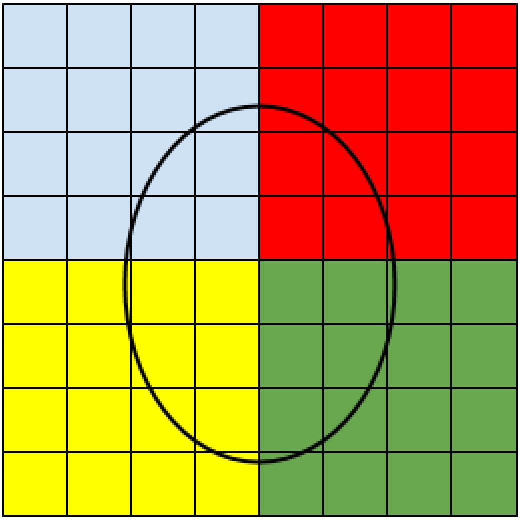


Figure 3.1: Illustration of the high correlation of the pixels of adjacent blocks.

tremendous correlation among them will not be efficiently employed unless they are located in the same block. This phenomenon is often known as JPEG blocking artifact [146]. Figure 3.2 shows the impact of JPEG blocking artict on visual perception. As it can be clearly seen, some artifacts are introduced due to the blockwise compression. To highlight the artifacts, some regions have been magnified and the artifacts are more perceptible. The principal reason of such artifact may be contributed to the blockwise coding and underlying assumption of independent uncorrelated adjacent blocks. In contrast, wavelet transform is applied into the whole image or subtiles of larger images. Hence, the blocking artifact will be negligible in comparison to DCT-based method [153].

In chapter 2, the DCT-based watermarking was addressed. As was mentioned earlier, the fundamental idea of DCT-based compression is independence of adjacent blocks. However, such hypothesisis is much more likely to hold in transform domain. That is to say, even though the coefficients may not be perfectly decorrelated in transform domain, they are still by far much more decorrelated than the heavily-correlated pixels in spatial domain. Hence, the idea of using DCT and wavelet sounds absolutely plausible. Inspired by this fact, we managed to employ DCT and wavelet jointly and gained the benefits of both transforms.

Despite the extensive usage of channel decorrelation in compression standards [132, 136, 148, 153], it has also been used in watermarking of common 2D images and video sequences [83–86, 154]. In this context, decorrelation basically refers to reduce the auto-correlation of a sequence. As the term implies, when the signal samples are correlated (depending on the amount of the correlation) the value of one sample may be estimated based on a priori knowledge of the value of other known samples [155]. The pixels of a a color image are typically represented using triple color channels, namely, R, G and B. These three elements can be considered as *pixel vectors*. For example, a color image with M rows and N columns has $M \times N$ pixel vectors. In order to boost the robustness of the proposed method, we investigated watermark insertion into luminance domain. The color conversion is regarded as a great tool of decorrelating the triple channels of a pixel vector. Even though the pixel vectors (R, G and B components) may also be decorrelated by common decorrelating transformations as DCT, wavelet, etc, the short length of the pixel vectors makes the conversion of color space a highly demanded approach instead. In other words, as the length of the pixel vectors is extremely short, it is not common to decorrelate such vectors with three elements by DCT or wavelet transformation. As a rule of thumb, in signal processing the signal is usually (much) longer than the the filter kernel which is going to be convolved with. Hence, the color conversion standards typically do not use a kernel but a 3×3 conversion matrix [156].

3.2 Discrete Wavelet Transform

The continuous wavelet transform (CWT) of a 1D function $f(t)$ is defined as [157]

$$W_f^{\psi}(a,\tau) = \int_{-\infty}^{+\infty} \psi_{a,\tau}^{*}(t) f(t) dt \ , \tag{3.1}$$

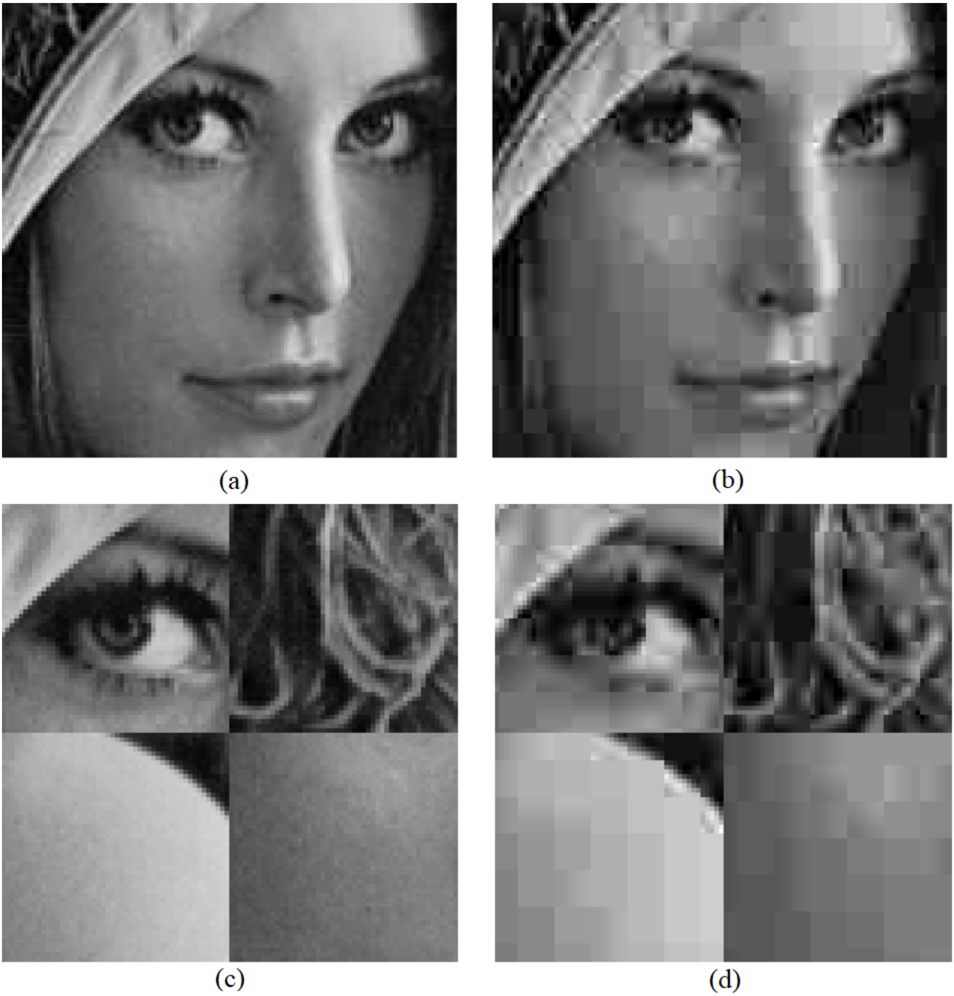


Figure 3.2: Illustration of the effect of JPEG blocking artifact: (a) the original image; and (b) the compressed image by JPEG at 33% compression ratio. Enlarged regions from: (c) the original image; and (d) JPEG-compressed image [139].

where * stands for complex conjugate and $\psi_{a,\tau}(t)$ is a shifted and scaled version (*child*) of a *mother* wavelet function $\psi(t)$, namely

$$\psi_{a,\tau}(t) = \frac{1}{\sqrt{a}}\,\psi\left(\frac{t-\tau}{a}\right), \tag{3.2}$$

in which a and τ are scale and shift, respectively. In fact, eq. 3.1 represents the convolution $\circledast$ of the rescaled version of ψ all over the domain of the function, that is

$$W_f^{\psi}(a,\tau) = \frac{1}{\sqrt{a}}\,\psi\left(\frac{\tau}{a}\right) \circledast f(\tau)\,. \tag{3.3}$$

Note also that this convolution becomes a *filtering* in the Fourier domain with a transfer function given by the Fourier transform of $\psi(t/a)$, $\Psi(a\omega)$.

If one wants the above transform to be invertible, i.e., that a complete recovery of the original function $f(t)$ from $W_f^{\psi}(a,\tau)$ is possible, some mathematical conditions are needed to be imposed to the wavelet function [158, 159]. On the one hand, the wavelet has to fulfil an integrability condition known as *admissibility* property, namely

$$C_\psi = \int_0^{+\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < +\infty\,. \tag{3.4}$$

As a consequence,

$$\Psi(0) = \int_{-\infty}^{+\infty} \psi(t)dt = 0\,, \tag{3.5}$$

which means that the average value of the wavelet in the time domain must be zero. This condition also leads to the fact that $\psi(t)$ must have a band-pass like spectrum.

On the other hand, *regularity conditions* state that the wavelet function should have some smoothness and concentration in both time and frequency domains. In mathematical terms, these conditions apply to the moments of $\psi(t)$

$$M_p = \int_{-\infty}^{+\infty} t^p \psi(t)dt\,, \tag{3.6}$$

that have to be zero (or small enough) for all $p < L$, L being the *approximation order* which, for a proper reconstruction, depends on the behaviour and smoothness of $f(t)$.

These conditions lead to one of the most interesting properties of the wavelet transform eq. 3.1, namely, the fact that changing the scale a implies a filtering with a simultaneous change in the central frequency ω_0 and the bandwidth BW. Specifically, the value a controls synchronously the bandwidth and the central frequency of the filtering in such a way that $BW, \omega_0 \propto 1/a$.

As stated previously, when *admissibility* and *regularity* conditions are fulfilled, an *inverse* CWT can be applied to recover the original function, namely

$$f(t) = \frac{1}{C_\psi} \int_{-\infty}^{+\infty} \int_{0}^{+\infty} W_f^\psi(a;\tau)\,\psi_{a;\tau}(t) \frac{da\, d\tau}{a^2} \,. \tag{3.7}$$

Note that from the *filtering* interpretation we presented above, it is clear that a very high redundancy is present in this form for the inverse CWT, since the bands of the filtering process are highly overlapping. It is, indeed, possible to use a discrete set of the whole family $\psi_{a;\tau}(t)$, whose members are named as *discrete* wavelets. Typically, for obtaining that set the mother wavelet is shifted and scaled by powers of two, i.e., considering scales of the form $a_j = \frac{1}{2^j}$ and displacements of $\tau_{j,k} = \frac{k}{2^j}$ $(j,k \in \mathbb{Z})$. These *dyadic* discrete wavelets are defined as

$$\bar{\psi}_{j,k}(t) = \sqrt{2^j}\,\psi\left(2^j t - k\right) \,, \tag{3.8}$$

where j and k represent the scale and shift parameters, respectively. Note that, despite their denomination, discrete wavelets are continuous functions of t, but their scaling and shifting operations from the original mother wavelet are discretized. One very interesting property of this family of functions is that by selecting properly $\psi(t)$ the whole set can be made orthonormal in the t domain, that is

$$\int_{-\infty}^{+\infty} \bar{\psi}_{j,k}(t)\,\bar{\psi}^*_{l,m}(t)\, dt = \delta_{j,l}\delta_{k,m} \,, \tag{3.9}$$

being $\delta_{j,l}$ the Kronecker delta. In this case, a complete reconstruction of any function $f(t)$ can be done by using the inversion formula

$$f(t) = \sum_{k=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} \bar{W}_f^\psi(j,k)\,\bar{\psi}_{j,k}(t) \,, \tag{3.10}$$

where

$$\bar{W}_f^{\psi}(j,k) = \int_{-\infty}^{+\infty} \bar{\psi}_{j,k}^{*}(t) f(t) dt \; . \tag{3.11}$$

It is interesting to note that when these discrete wavelets are used, the result of the transform is a discrete series of *wavelet coefficients* $\bar{W}_f^{\psi}(j,k)$, and eq. 3.10 represents the *wavelet series decomposition*.

Note also that for a given value of j (scale) the set of coefficients $\{\bar{W}_f^{\psi}(j,k)\}_{k\in\mathbb{Z}}$ represents a sampled version of the convolution of the signal $f(t)$ with the scaled mother wavelet $\psi(2^j t)$. This convolution is equivalent to, as stated previously, a band-pass filtering with central frequency and bandwidth proportional to 2^j. Thus, the higher the value of j (smaller scale factors), the higher the frequencies considered in the corresponding coefficients. On the other hand, the smaller the j values, the closer to zero will be the central frequency of the corresponding subband. However, as mentioned above, the bandpass nature of the mother wavelet makes that only including the infinite number of subbands a proper reconstruction of the low frequencies of the signal will be achieved. Following this reasoning, the typical approach to wavelet decomposition introduces a new function, the *father* wavelet $\phi(t)$ (a.k.a. *scaling function*), that *fills* the low-frequency region not covered by the mother wavelet $\psi(t)$ [160]. Thus, considering the bandpass nature of the mother wavelet, the father wavelet has to act as a low pass filter on the CWT of any function $f(t)$. Again, its dyadic discrete version is defined as

$$\bar{\phi}_{j,k}(t) = \sqrt{2^j}\, \phi\left(2^j t - k\right) \; . \tag{3.12}$$

The scaling function is selected conventionally in such a way that the following orthogonality properties hold

$$\begin{aligned} \int_{-\infty}^{+\infty} \bar{\phi}_{j,k}(t)\bar{\phi}_{j,m}^{*}(t)\, dt &= \delta_{k,m} \; , \\ \int_{-\infty}^{+\infty} \bar{\psi}_{j,k}(t)\bar{\phi}_{j,m}^{*}(t)\, dt &= 0 \; . \end{aligned} \tag{3.13}$$

Let us consider from now on that the original signal $f(t)$ has a finite frequency (Fourier) band, which is a common assumption in image processing. Then it is possible to choose the initial scale of the father (and mother) wavelet to just cover, for a scale given by $j = 0$, the whole frequency interval with the band of the father wavelet. In this way, the signal can be expressed as

$$f(t) = \sum_{k=-\infty}^{\infty} s_f^{\phi}(0,k)\bar{\phi}_{0,k}(t) , \tag{3.14}$$

where

$$s_f^{\phi}(j,k) = \int_{-\infty}^{+\infty} \bar{\phi}_{j,k}^{*}(t) f(t) dt . \tag{3.15}$$

On the other hand, we can use the scaling and wavelet functions corresponding to $j = -1$ (upper scale factor) to decompose the signal, as

$$\begin{aligned} f(t) = \sum_{k=-\infty}^{\infty} \left[s_f^{\phi}(-1,k)\bar{\phi}_{-1,k}(t) + \bar{W}_f^{\psi}(-1,k)\bar{\psi}_{-1,k}(t) \right] = \\ = f^{(-1)}(t) + \sum_{k=-\infty}^{\infty} \bar{W}_f^{\psi}(-1,k)\bar{\psi}_{-1,k}(t) , \end{aligned} \tag{3.16}$$

Note that now the coefficients have to be computed from a new set of projections of the signal $f(t)$ onto $\bar{\phi}_{-1,k}(t)$ and $\bar{\psi}_{-1,k}(t)$. However, these functions can be expressed in terms of the previous scale father wavelet function by using the *two-scale relations* [159]

$$\begin{aligned} \bar{\phi}_{j-1,k}(t) &= \sum_{k'=-\infty}^{\infty} p(k')\bar{\phi}_{j,k'+2k}(t) , \\ \bar{\psi}_{j-1,k}(t) &= \sum_{k'=-\infty}^{\infty} q(k')\bar{\phi}_{j,k'+2k}(t) , \end{aligned} \tag{3.17}$$

where $p(k')$ and $q(k')$ represent the *interscale* coefficients, that can be obtained by the projection operations

$$\begin{aligned} \int_{-\infty}^{+\infty} \bar{\phi}_{j-1,k}(t)\bar{\phi}_{j,m}^{*}(t)\, dt &= p(m-2k) , \\ \int_{-\infty}^{+\infty} \bar{\phi}_{j-1,k}(t)\bar{\psi}_{j,m}^{*}(t)\, dt &= q(m-2k) . \end{aligned} \tag{3.18}$$

By using these relationships, it is straightforward to find that

$$
\begin{aligned}
s_f^{\phi}(j-1,k) &= \sum_{k'=-\infty}^{\infty} p(k')s_f^{\phi}(j,k'+2k)\;, \\
\bar{W}_f^{\psi}(j-1,k) &= \sum_{k'=-\infty}^{\infty} q(k')s_f^{\phi}(j,k'+2k)\;,
\end{aligned}
\tag{3.19}
$$

that is, the corresponding coefficients for the low and high frequency bands now can be obtained from the previous ones by a linear superposition. This process can be extended recursively to any further step in the subband division, and thus all the needed coefficients for any scale level decomposition can be expressed through combinations of the first coefficients $s_f^{\phi}(0,k)$ by means of the interscale factors $p(k)$ and $q(k)$. Note also from eqs. 3.19 that for any step in the subdivision process, the new coeffcients are affected only by the contribution of a subset of the previous ones. In fact, eqs. 3.19 represent the correlations between the projections at subband j, $s_f^{\phi}(j,k)$, and $p(k)$ and $q(k)$, respectively. This correlation is then downsampled by a factor of two (see the double-shift of p and q in eqs. 3.19). The set of interscale coefficients $p(k)$ and $q(k)$ are called the discrete low-pass (LPF) and high-pass filters (HPF), respectively.

Summarizing, wavelet series decomposition can be completed in an iterative way. First, we compute the upper scale vector of coefficients $\{s_f^{\phi}(0,k)\}$ by projecting the original signal onto the father wavelet $\bar{\phi}(0,k)$. Next, we correlate this vector with the interscale coefficients vectors $\{p(k)\}$ and $\{q(k)\}$ and apply a downsampling of factor 2 to obtain the coefficients for scale level $j=-1$. This new set of coefficients is again correlated and downsampled to obtain the coefficients for scale level $j=-2$, and so on. If we stop the process at a scale level $j=-J$, the final series decomposition will be expressed as

$$
f(t) = \sum_{k=-\infty}^{\infty} \left[s_f^{\phi}(-J,k)\bar{\phi}_{-J,k}(t) + \sum_{j=1}^{J} \bar{W}_f^{\psi}(-j,k)\bar{\psi}_{-j,k}(t) \right] \;. \tag{3.20}
$$

The set of coefficients in this decomposition represents the DWT of the signal $f(t)$ up to scale level $j=-J$, while last equation serves as a definition for the inverse DWT from this set.

The above reasoning is in fact based on the so-called multiresolution (MR) approach, proposed in the eighties by Mallat et al. [157, 161]. Mallat showed that a particular type of filters can be used to decompose a discrete signal into low frequency and high frequency components each with half number of samples of the original signal. In fact, his major contribution was to propose calculating eq. 3.1 with *Quadrature Mirror Filters* (QMF) [160, 162]. Mallat's MR approach facilitates the signal decomposition into multiple levels. The *coarse* resolution is associated with low frequency components and represents the main features of the signal shape. Conversely, the fine resolution is associated with high frequency components. The QMF (either high-pass or low-pass) used for decomposition of the signal, are typically referred to as *decomposition* or *analysis* filters. The output of the decomposition filter is then downsampled by a factor of two. The downsampled frequency components may then be utilized for complete retrival of the original signal. The filters used for reconstruction of the signal, are typically referred to as *reconstruction* or *synthesize* filters. Suppose the low-pass and high-pass decomposition filters are LPF_D and HPF_D respectively. Similarly, suppose the low-pass and high-pass filters for reconstruction are LPF_R and HPF_R. Decompostion and reconstruction filters are related as follows [163]. Figure 3.3 shows the decomposition and reconstruction of a 1D signal by a single step in DWT. The $2\downarrow$ and $2\uparrow$ symbols stand for dyadic downsampling and upsampling, respectively. Technically, this upsampling in reconstruction is performed by interleaving null values in the corresponding initial coefficient vector.

Finally, it is interesting to say that this DWT can be applied straighforwardly to sampled finite-time signals, instead of continuous ones, by using sampled versions of the mother and father wavelets. For a number of samples equal to $M = 2^N$, the maximum scale factor $j = -J$ is set then by choosing $J = N$. Hence, for a such a discrete $\{f_i\}_{0\leq i\leq M-1}$ 1D signal, we have

$$f_i = \sum_{k=-\infty}^{\infty}\left[s_f^{\phi}(-N,k)\bar{\phi}_{-J,k}(i) + \sum_{j=1}^{N}\bar{W}_f^{\psi}(-j,k)\bar{\psi}_{-j,k}(i)\right] . \tag{3.21}$$

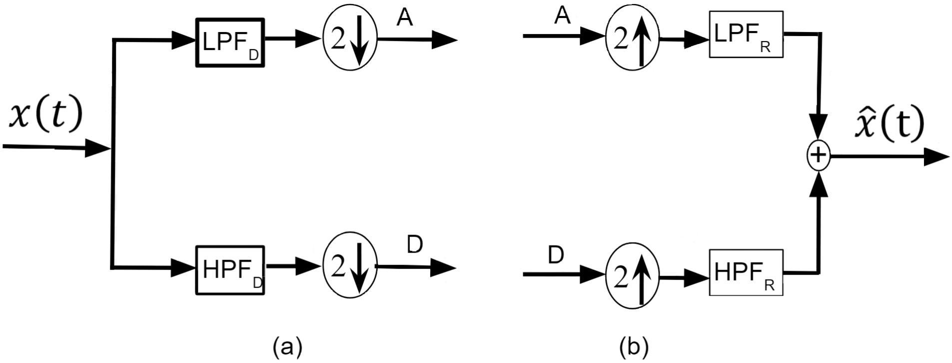


Figure 3.3: (a) Decomposition and (b) reconstruction of a 1D signal by DWT with a single-step scale change.

3.3 4D Discrete Wavelet Transform of a Lightfield

Unsurprisingly, there may be a great amount of visual similarity between the neighboring *EI*s horizontally, vertically and diagonally in a given LF. In other words, every single *EI* has far higher correlation with neighboring *EI*s than further ones. For example, the correlation between every pair of *EI*s chosen from $\{(1,1),(1,2),(2,1),(2,2)\}$ will probably be higher than that of $\{(1,1),(16,16)\}$. In this dissertation, we refer to the correlation of the pixels of the same *EI* as *intracorrelation* while the correlation among *EI*s will be referred to as *intercorrelation*. The enormous intercorrelation of *EI*s brings the idea of gaining advantage of both intracorrelation and intracorrelation of the LF. That is why we proposed the *4D wavelet* for LF watermarking purpose. The underlying reason of employing the 4D wavelet for LF watermarking is exactly the same as using 3D decorrelation of video sequences (decorrelating the rows and columns of each frame and nearby frames). The usage of the 4D wavelet for LF watermarking ensures gaining the maximal benefit of the huge intercorrelation of the LF pixels. Even if the numerous methods of 2D image watermarking are directly applicable to the LF, there is always a major concern that such methods may ruin the angular information carried by the LF. In contrast, 4D watermarking is customized for LF promising full protection of spatial and angular information incorporated with the LF. Even though it is possible to apply 2D wavelet to a plenoptic image, it will be detrimental to both spatial and angular information of the LF. The LF is not a 2D image and every single dimension should be processed properly. Sliding the filter bank over the rows and columns of the plenoptic image never means to protect neither the spatial nor the angular information of the LF.

As stated before, the LF is by nature 4D and it sounds plausible to use 4D wavelet for decorrelation task. Having that said, eq. 3.1 may be extended as

$$W_f^{\psi}(a,s_0,t_0,u_0,v_0) = \iiiint_{-\infty}^{+\infty} \psi_{a,s_0,t_0,u_0,v_0}(s,t,u,v) LF(s,t,u,v) ds\,dt\,du\,dv \,, \tag{3.22}$$

where

$$\psi_{a,s_0,t_0,u_0,v_0}(s,t,u,v) = \frac{1}{\sqrt{a^4}}\,\psi\left(\frac{s-s_0}{a},\frac{t-t_0}{a},\frac{u-u_0}{a},\frac{v-v_0}{a}\right), \tag{3.23}$$

s_0, t_0, u_0 and v_0 being the continuous shifts along the 4D LF space. Assuming separability of multidimensional wavelet transform [164], we have

$$\begin{aligned}\psi_{a,s_0,t_0,u_0,v_0}(s,t,u,v) &= \frac{1}{a^2}\,\psi\left(\frac{s-s_0}{a}\right)\psi\left(\frac{t-t_0}{a}\right)\psi\left(\frac{u-u_0}{a}\right)\psi\left(\frac{v-v_0}{a}\right)\\ &= \frac{1}{a^2}\,\psi_{a,s_0}(s)\,\psi_{a,t_0}(t)\,\psi_{a,u_0}(u)\,\psi_{a,v_0}(v)\,,\end{aligned} \tag{3.24}$$

where we consider the same mother function ψ for all 4 dimensions. Following the same reasoning as in previous section, a 4D scaling function (father wavelet) $\phi(s,t,u,v)$ is also defined. Discrete dyadic 4D wavelets will be defined now as

$$\begin{aligned}\bar{\psi}_{j,m,n,p,q}(s,t,u,v) &= 2^{2j}\psi(2^j s - m, 2^j t - n, 2^j u - p, 2^j v - q)\,,\\ \bar{\phi}_{j,m,n,p,q}(s,t,u,v) &= 2^{2j}\phi(2^j s - m, 2^j t - n, 2^j u - p, 2^j v - q)\,,\end{aligned} \tag{3.25}$$

where m, n, p and q indicate shiftings along the 4D LF space. For 4D LF, a DWT up to scale level $j = -J$ as in eq. 3.20 will be written as

$$\begin{gathered} LF(s,t,u,v) = \frac{1}{\sqrt{n_{EI,h}\, n_{EI,v}\, n_{\mu I,h}\, n_{\mu I,v}}} \times \\ \left(\sum_{s_0=0}^{n_{EI,h}-1} \sum_{t_0=0}^{n_{EI,v}-1} \sum_{u_0=0}^{n_{\mu I,h}-1} \sum_{v_0=0}^{n_{\mu I,v}-1} LLLL(-J, s_0, t_0, u_0, v_0)\, \bar{\phi}_{-J,s_0,t_0,u_0,v_0}(s,t,u,v) + \right. \\ \left. \sum_{ABCD} \sum_{j=1}^{J} \sum_{s_0=0}^{n_{EI,h}-1} \sum_{t_0=0}^{n_{EI,v}-1} \sum_{u_0=0}^{n_{\mu I,h}-1} \sum_{v_0=0}^{n_{\mu I,v}-1} ABCD(-j, s_0, t_0, u_0, v_0)\, \bar{\psi}^{ABCD}_{-j,s_0,t_0,u_0,v_0}(s,t,u,v) \right) , \end{gathered} \tag{3.26}$$

where

$$A,B,C,D \in \{L,H\},\ ABCD \neq LLLL\,, \tag{3.27}$$

L and H standing for highpass and lowpass, respectively. This kind of filtering is applied, in the same order as in the *ABCD* sequence, to the spatial-angular coordinates (s,t,u,v) that describe the LF. Similar to 1D wavelet, it makes perfect sense to realize 4D wavelet transform by filter banks. The low-frequency component *LLLL* represents the low-frequency section of the spectrum along all spatial and angular dimensions. *LLLL* is obtained by

$$\begin{gathered} LLLL(s_0,t_0,u_0,v_0) = \frac{1}{\sqrt{n_{EI,h}\, n_{EI,v}\, n_{\mu I,h}\, n_{\mu I,v}}} \times \\ \left(\sum_{s=0}^{n_{EI,h}-1} \sum_{t=0}^{n_{EI,v}-1} \sum_{u=0}^{n_{\mu I,h}-1} \sum_{v=0}^{n_{\mu I,v}-1} LF(s,t,u,v)\, \bar{\phi}^{*}_{-J,s_0,t_0,u_0,v_0}(s,t,u,v) \right) . \end{gathered} \tag{3.28}$$

The detail (high frequencies) coefficients are obtained by

$$\begin{gathered} ABCD(j,s_0,t_0,u_0,v_0) = \frac{1}{\sqrt{n_{EI,h}\, n_{EI,v}\, n_{\mu I,h}\, n_{\mu I,v}}} \times \\ \left(\sum_{s=0}^{n_{EI,h}-1} \sum_{t=0}^{n_{EI,v}-1} \sum_{u=0}^{n_{\mu I,h}-1} \sum_{v=0}^{n_{\mu I,v}-1} LF(s,t,u,v)\, \bar{\psi}^{ABCD}_{j,s_0,t_0,u_0,v_0}(s,t,u,v) \right) . \end{gathered} \tag{3.29}$$

Considering the implementation of 1D wavelet, it sounds absolutely plausible to achieve the 4D wavelet with filter banks as well. Figure 3.4 shows the block diagram of the 4D wavelet decomposition employed for watermarking purpose. Even though the order of applying 4D transform to the different dimensions does not matter, we applied the transform into the rows and columns of the *EI*s and then into the rows and columns of μIs, respectively.

Just like the 1D case, the output of the decomposed LF from every filter is also downsampled by a factor of two. Anyway, using the filter banks to calculate the wavelet coefficients will make no difference in the final results and the same coefficients may be directly calculated without employing any filter bank.

Figures 3.5 and 3.6 show the decomposed subbands of the 4D wavelet transform. We choose for this illustration the Haar wavelet set, that will be introduced later on in this chapter. If the input LF has $16 \times 16 \times 300 \times 300$ pixels, the decomposed subbands will have $8 \times 8 \times 150 \times 150$ coefficients. Nevertheless, we have only visualized 8×8 *EI*s of the LF and 4×4 blocks of the subbands. Additionally as the 4D wavelet is only applied into Y component, the LF *EI*s are converted into gray level images. It is noticeable that subsampling has not only halved down the number of rows and columns of the *EI*s (spatial subsampling) but also has halved down the dimentions of the μIs both horizontally and vertically (angular subsampling).

Figure 3.7 illustrates the block diagram of the 4D wavelet reconstruction. The decomposed subbands are *upsampled* both spatially and angularly. The upsampled coefficients are then passed through reconstruction filters.

3.4 Embedding Procedure

As mentioned in section 2.1, DCT and SVD are used for LF watermarking. However, while the hypothesis of independent blocks may not be necessarily true in spatial domain, it holds far more in the transform domain. Joint utilization of DCT and DWT will result in gaining the advantages of both transforms. Another enormous redundancy is the interchannel correlation

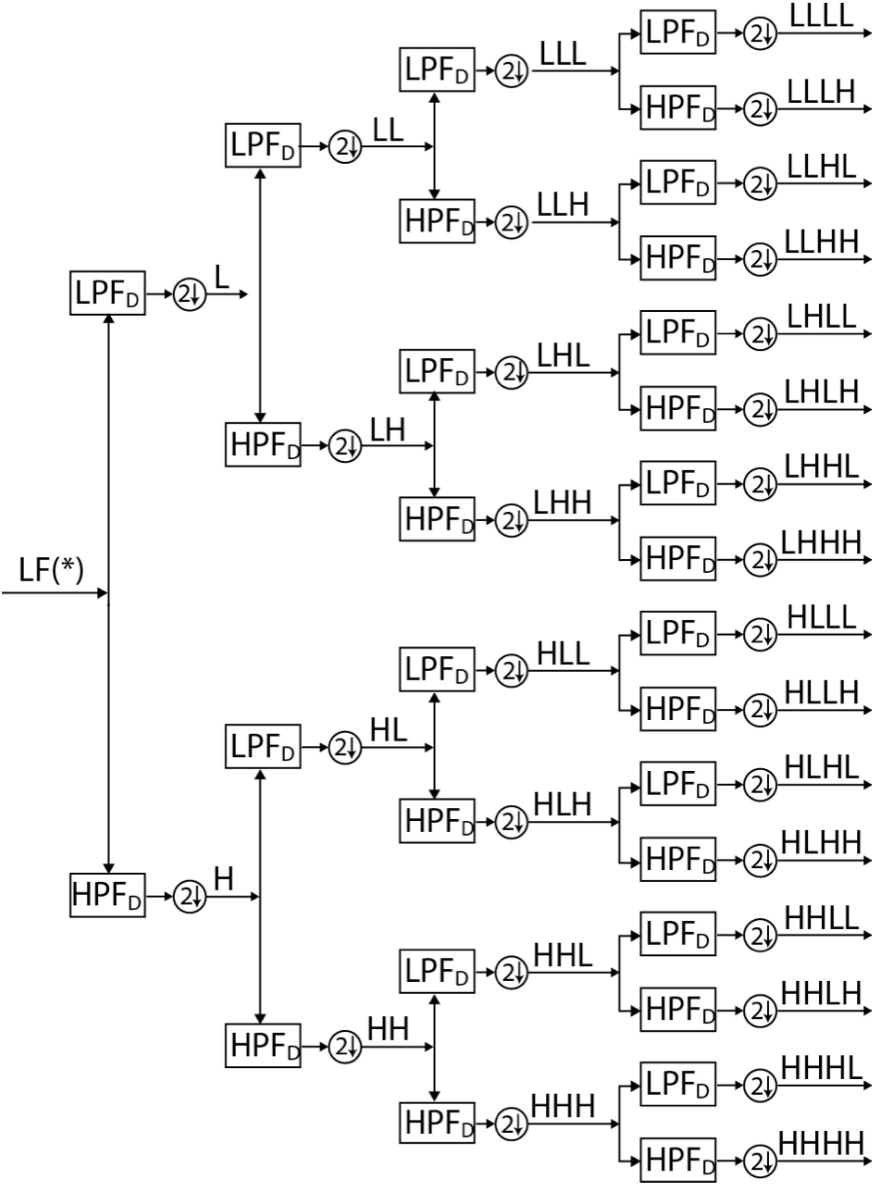


Figure 3.4: 4D wavelet decomposition of LF by filter banks

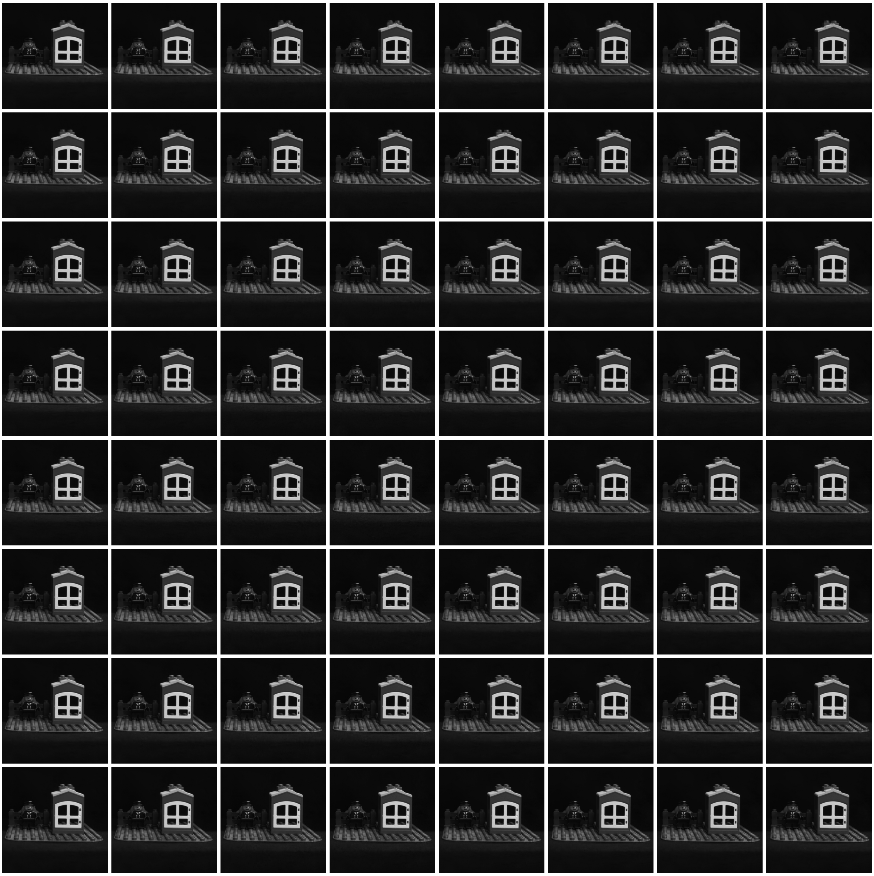


Figure 3.5: 4D wavelet decomposition of $16 \times 16 \times 300 \times 300$ *LF by filter banks (only* 8×8*EIs have been shown).*

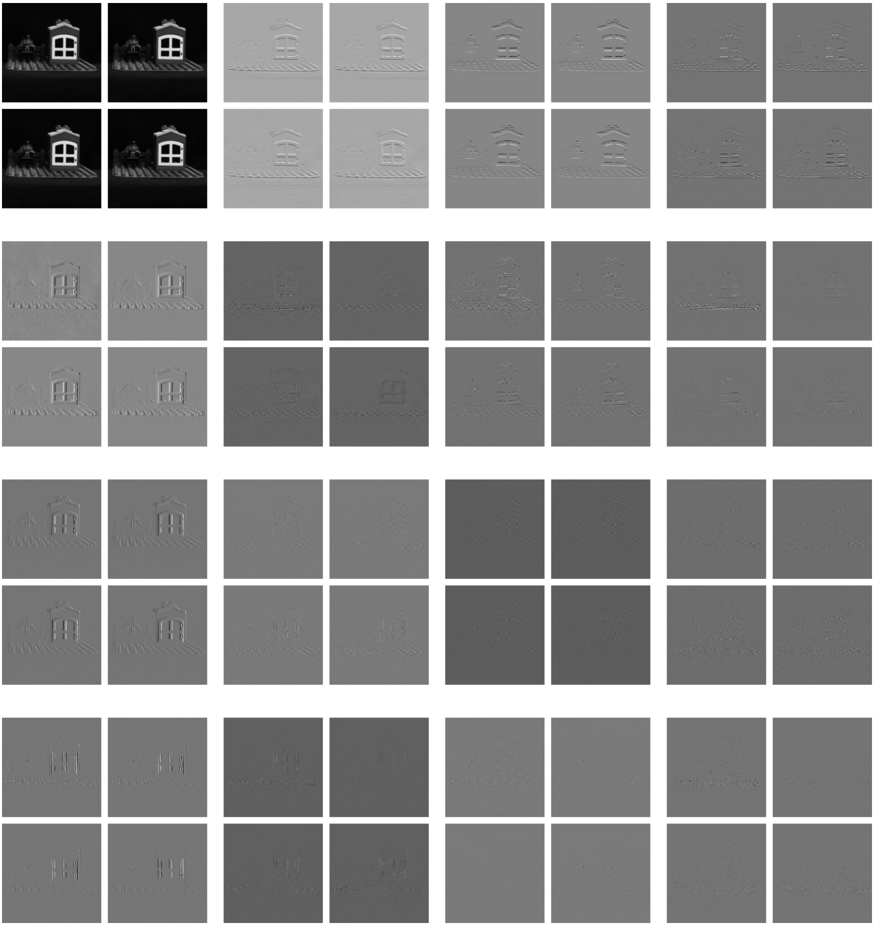


Figure 3.6: 4D wavelet decomposition of $16 \times 16 \times 300 \times 300$ *LF by filter banks (only* 4×4*EIs of each subband have been shown).*

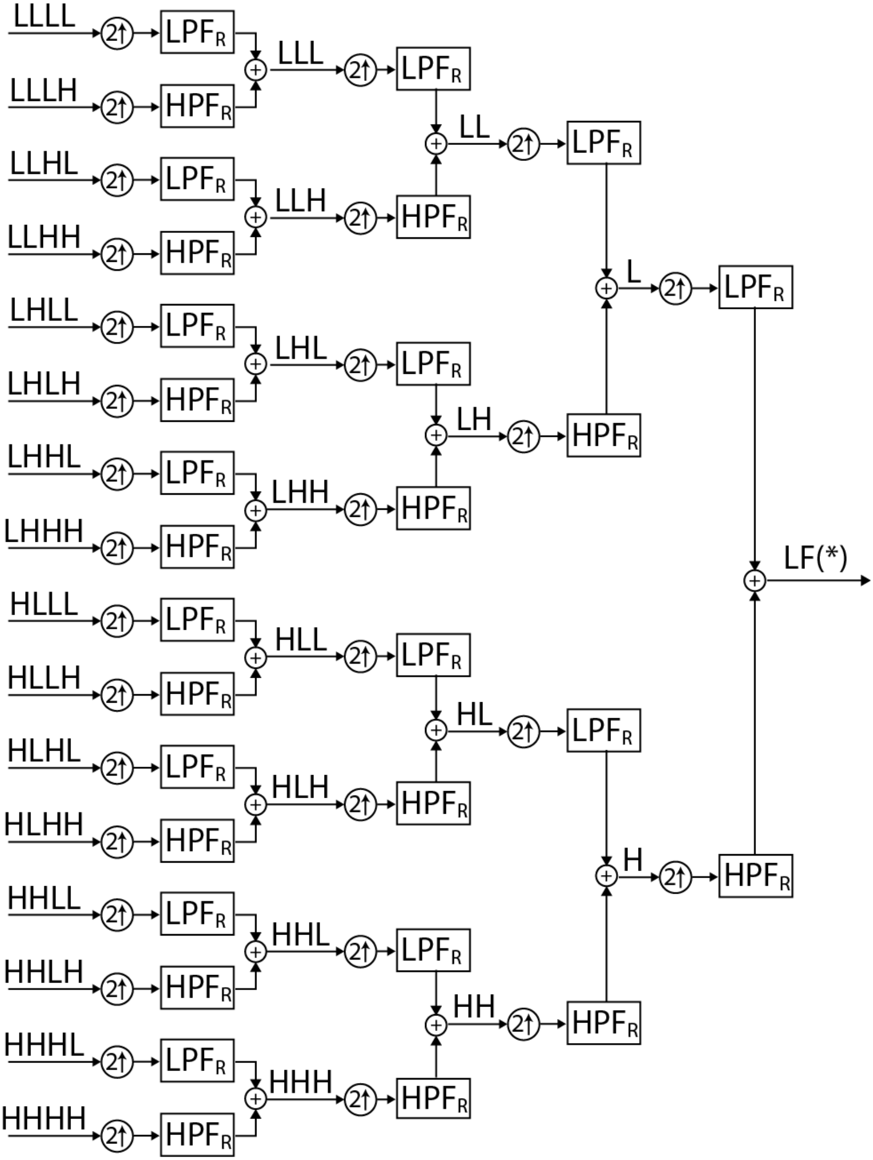


Figure 3.7: 4D Wavelet reconstruction by filter banks

of triple channels of the LF pixels. Theoretically, it is possible to reduce this correlation by DCT, DWT, Fourier or any other transformation. Nevertheless, given the short length of the pixel vectors in RGB domain, it does not seem wise using a filter longer than three. Instead, a very common approach for channel decorrelation is RGB to YUV and is widely used in most compression standards. Additionally, it is very common for most of the image/video coding standards to downsample the chrominance component as 4:4:2 or 4:2:2 [132, 150, 165–167]. However, as the proposed method does not concern the data compression, no downsampling will be carried out. Figure 3.8 highlights the block diagram of the embedding procedure of the second proposed method. As is evident from Fig. 3.8, first the pixels are transformed from color domain to chrominance-luminance domain. Then 4D wavelet transform is applied into luminance component. Specified by the secret key, the desired *LLLL* coefficients are chosen to embed the watermark. The chosen *LLLL* coefficients are grouped into different blocks. Each block carries one single bit of watermark. Suppose the *i*th row and

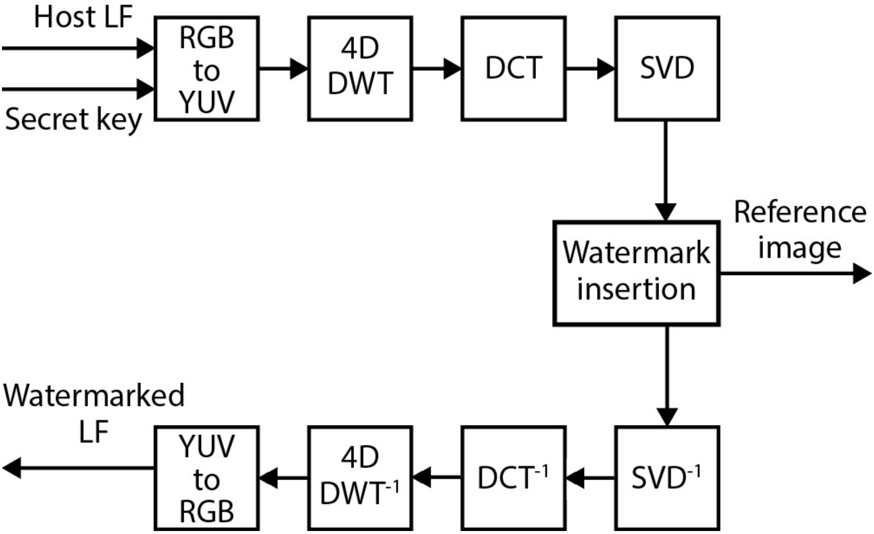


Figure 3.8: The block diagram of joint DCT and wavelet watermarking.

jth column of the watermark is embedded in the block $LLLL\_blk_{ij}$.

Even though the number of the rows and columns of the chosen blocks does not matter, without loss of generality, suppose the number of the rows and columns of the blocks equal $BlockSize$. Obviously, this assumption is only for notational convenience and the block may have arbitrary dimensions. The DCT coefficients of each selected block are arranged in zigzag order

$$blk\_LLLL\_zz_{ij} = zigzag(DCT(LLLL\_blk_{ij})) \tag{3.30}$$

where $zigzag$ is defined as eq. 2.8. Not all the ordered zigzag coefficients are used for watermark insertion. The first n_dct coefficients of $LLLL\_blk_{ij}$ are chosen

$$blk\_LLLL\_zz\_wm_{ij} = \begin{bmatrix} blk\_LLLL\_zz_0 \\ blk\_LLLL\_zz_1 \\ \vdots \\ blk\_LLLL\_zz_{n\_dct-1} \end{bmatrix} \tag{3.31}$$

The obtained vector is then factorized by SVD

$$blk\_LLLL\_zz\_wm_{ij} = U\Sigma V^T \tag{3.32}$$

The top left entry σ_{ij} will be used to embed the ith row and jth column of the watermark

$$\sigma_{wm\_ij} = \begin{cases} \sigma_{ij} + gf & wm\_bit_{ij} = 1 \\ \sigma_{ij} - gf & wm\_bit_{ij} = 0 \end{cases} \tag{3.33}$$

where σ_{wm_ij} stands for the largest singular value of the watermarked coefficient and σ_{ij} is that of coefficients from the host LF employed to carry the ith row and jth column of the watermark. σ_{ij} will later on be used in extraction procedure and is stored in the reference image as detailed in section 2.4. The first n_dct coefficients of the watermarked block are generated as

$$blk\_LLLL\_zz\_dct\_rec_{ij}\Big|_{n\_dct-1}^{0} = U\Sigma_{wm}V^T \tag{3.34}$$

in which $x\Big|_{N_2}^{N_1}$ refers to

$$x_{N_1}, x_{N_1+1}, x_{N_1+2}, \ldots, x_{N_2} \quad (N_1 < N_2)$$

As the other DCT coefficients of $LLLL\_blk_{ij}$ have not been modified through embedding procedure, we have

$$\begin{aligned} blk\_LLLL\_zz\_dct\_rec_{ij}\Big|_{(BlockSize*BlockSize)-1}^{n\_dct} \\ = blk\_LLLL\_zz_{ij}\Big|_{(BlockSize*BlockSize)-1}^{n\_dct} \end{aligned} \tag{3.35}$$

To retrieve the $LLLL$ coefficients of the watermarked LF in the wavelet domain, inverse DCT is applied into $blk\_LLLL\_zz\_dct\_rec$

$$blk\_LLLL\_wm_{ij} = DCT^{-1}(zigzag^{-1}(blk\_LLLL\_zz\_dct\_rec_{ij})) \tag{3.36}$$

where $blk\_LLLL\_wm_{ij}$ is the block carrying watermarked $LLLL$ coefficients. As the watermark is embedded only into $LLLL$, the other subbands will directly be fed into inverse 4D wavelet transform to acquire the Y component of the watermarked LF

$$\begin{gathered} Y_{wm} = IDWT4D(LLLL_{wm}, ABCD) \\ A,B,C,D \in \{L,H\},\ ABCD \neq LLLL \end{gathered} \tag{3.37}$$

In which $IDWT4D(:)$ is the inverse 4D wavelet transform and can be achieved by filter banks and downsamplers (Fig. 3.7) or equivalently by calculating eq. 3.26. Finally, the last step will be to attain watermarked LF in RGB domain

$$\begin{bmatrix} R_{wm} \\ G_{wm} \\ B_{wm} \end{bmatrix} = \begin{bmatrix} T_{00} & T_{01} & T_{02} \\ T_{10} & T_{11} & T_{12} \\ T_{20} & T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} Y_{wm} \\ U \\ V \end{bmatrix}, \tag{3.38}$$

where $\{T_{ij}\}_{0\leq i,j\leq 2}$ is the transfer YUV-to-RGB matrix. As the U,V components of the host LF are not modified, the same components will be employed to generate the watermarked LF.

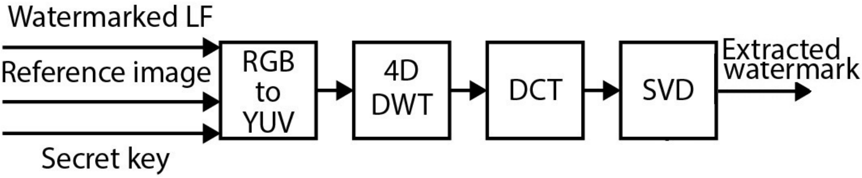


Figure 3.9: The extraction procedure of joint DCT and wavelet watermarking method.

3.5 Extraction Procedure

The mechanism of the watermark extraction is very similar to that of the watermark insertion. The conversion from RGB domain to chrominance-luminance, 4D-DWT, DCT and SVD are performed the same way as stated in section 3.4. Figure 3.9 shows the extraction procedure. In order to extract the watermark bit, the singular value of the corresponding block and reference image will be used, with the following selection rule

$$wm\_ext_{ij} = \begin{cases} 1 & \sigma_{ij} > ref\_img_{ij}, \\ 0 & \sigma_{ij} < ref\_img_{ij}. \end{cases} \tag{3.39}$$

3.6 Experimental Results

3.6.1 Implementation Remarks

As mentioned earlier in section 2.2, the DCT basis functions are fixed. Conversely, the basis functions of the wavelet depend on the chosen filter type. In some specific applications like electroencephalography signal processing, analysis of pavement roughness,... specific filter banks may be preferred [168–172]. As it is apparent from section 3.2, if the coefficients of the filter meet some specific requirements, it can be used as wavelet filter. In the wavelet transform, the acquisition of information localized within the signal is dependent on the selection of the filter banks. There are a lot of different filters that can be used to decompose and reconstruct the signals by wavelet transform [163, 173]. In this work, we have used Haar filters to implement the 4D wavelet transform, whose mother wavelet function is defined as

$$\psi(t) = \begin{cases} 1 & 0 \leq t < \frac{1}{2}, \\ -1 & \frac{1}{2} \leq t < 1, \\ 0 & \text{elsewhere.} \end{cases} \tag{3.40}$$

Its scaling function (father wavelet) can be described as

$$\phi(t) = \begin{cases} 1 & 0 \leq t < 1, \\ 0 & \text{elsewhere.} \end{cases} \tag{3.41}$$

Figure 3.10 shows a representation of this Haar set.

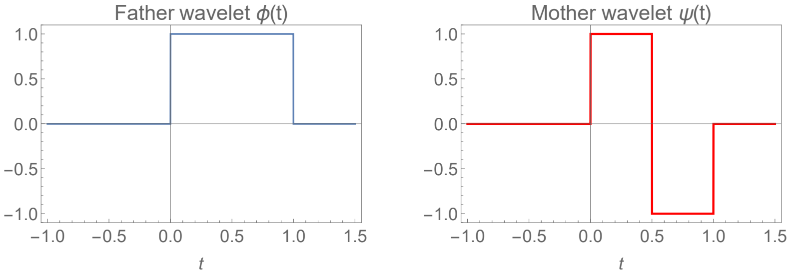


Figure 3.10: Haar wavelet functions representation.

From a computational point of view, the 4D wavelet was implemented to watermark LF using OpenCV library. To improve the speed, the CPU cores were parallelized by multithreading techniques.

3.6.2 Objective and Subjective Performance

As mentioned in chapter 1, the LF watermarking has rarely been addressed in the literature. In order to make a fair comparison, the method proposed in this chapter (method II from now on) is compared with that of chapter 2 (method I). To highlight the importance of the individual blocks of embedding (Fig. 3.8) and extraction (Fig. 3.9) procedures, both proposed mthods are also compared with SVD method as defined in section 2.6.3.

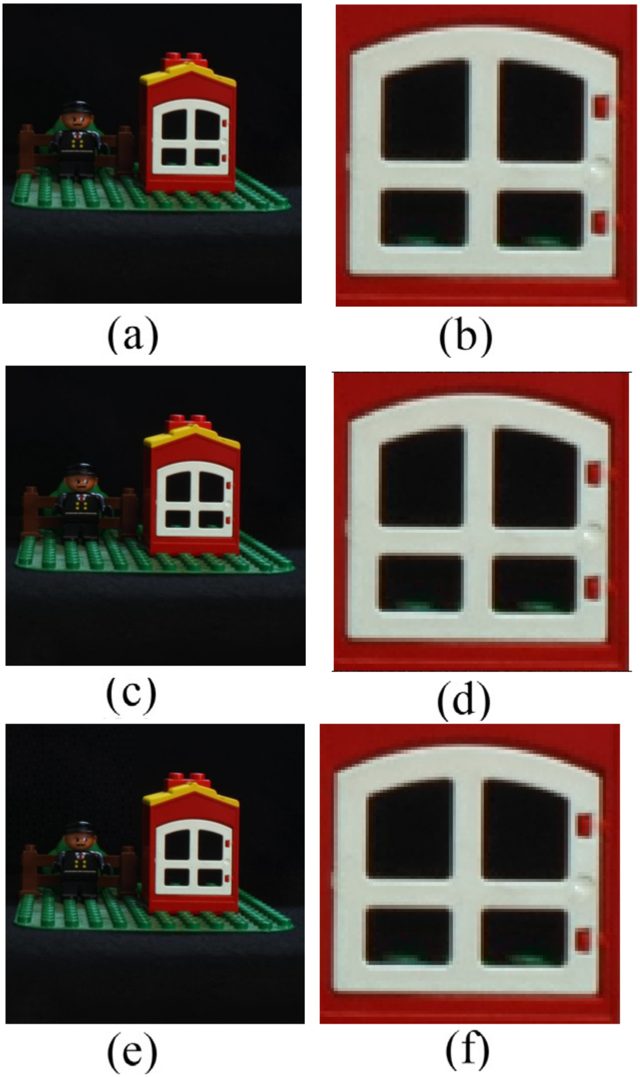


Figure 3.11: Verification of the proposed method on man LF: (a, b) the host image; (c, d) the watermarked image with proposed method I; (e, f) the watermarked image with proposed method II.

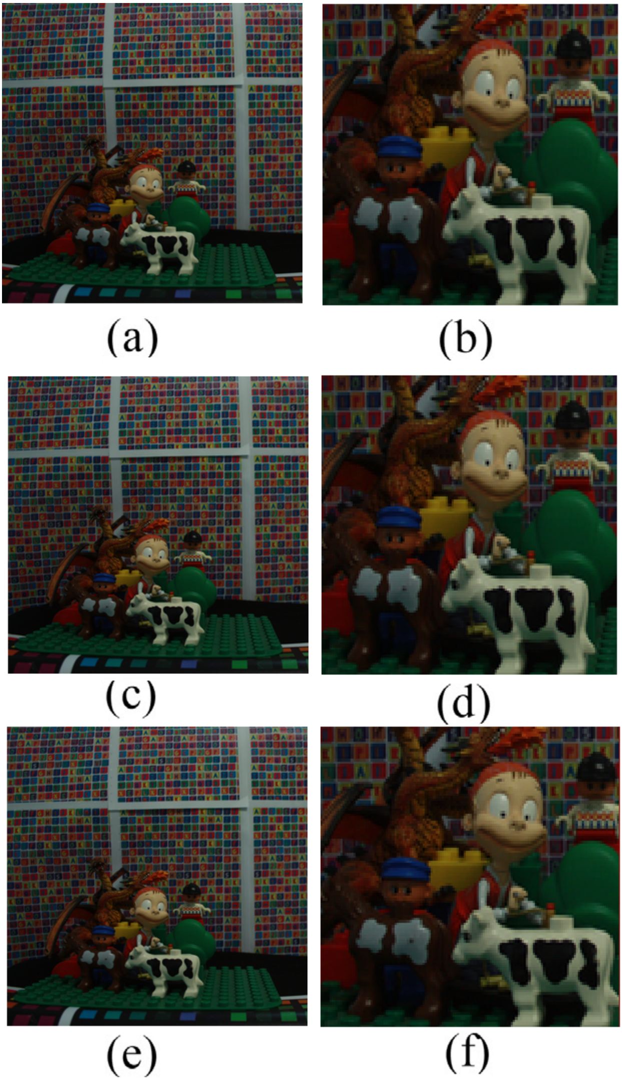


Figure 3.12: Verification of the proposed method on toys LF: (a, b) the host image; (c, d) the watermarked image with proposed method I; (e, f) the watermarked image with proposed method II.

Figure 3.13: (a) The embedded watermark; (b) the extracted watermark (man LF).

Figure 3.14: (a) The embedded watermark; (b) the extracted watermark (toys LF).

In other words, the SVD method excludes all the other transformations as color space conversion, 4D DWT and DCT. The simulations have been conducted with two different LF datasets. The first one is the same as previously used in section 2.6 which is referred to as *man* hereafter. The other LF is a new one which is referred to as *toys*. Similar to our visualization approach on chapter 2, the central view of the man LF is picked up from eighth row and eighth column while the central view of toys LF has been chosen from ninth row and ninth column. This difference is because the number of the μIs of man and toys are 16×16 and 17×17, respectively. Throughout this chapter, we will use the same central *EI*s for man and toys unless otherwise stated. The central *EI* of the host and watermarked LF for both man and toys are shown in Fig. 3.11 and Fig. 3.12. For $gf = 90$, $BlockSize = 4$ and $n\_dct = 6$ the achieved *PSNR* is 53.7434 dB for the man LF and 54.3434 dB for the toys LF. Unlike chapter 2 where 8×8 watermark had been used, we managed to use a watermark sixteen times larger. The embedded watermark has dimensions of 32×32. As is evident from Fig. 3.11 and Fig. 3.12, the host and watermarked LF seem identical so they can really not be distinguished. For both LFs, even for $gf = 170$, the *PSNR* remains higher than 48 dB which is far imperceptible for HVS.

Figure 3.13 shows the embedded and the extracted watermark into and from the man LF. As can be seen from Fig. 3.13, the embedded and extracted watermark are absolutely identical. Figure 3.14 also shows the embedded and the extracted watermarked into and from the toys LF. Similar to the man LF, no difference is observed between the embedded and the extracted watermark.

Figure 3.15(a) and Fig. 3.16(a) show the *PSNR* of the watermarked LF. As can be seen from Fig. 3.15(a) and Fig. 3.16(a), the *PSNR* of the proposed method may be slightly lower than that of chapter 2. However, it has no noticeable impact on HVS perception for gf values offerring good robustness against different attacks.

Contrary to most watermarking methods in the literature, the proposed method offers an almost error-free watermark extraction for operational range of gf. This statement also holds for the lowest values of gf.

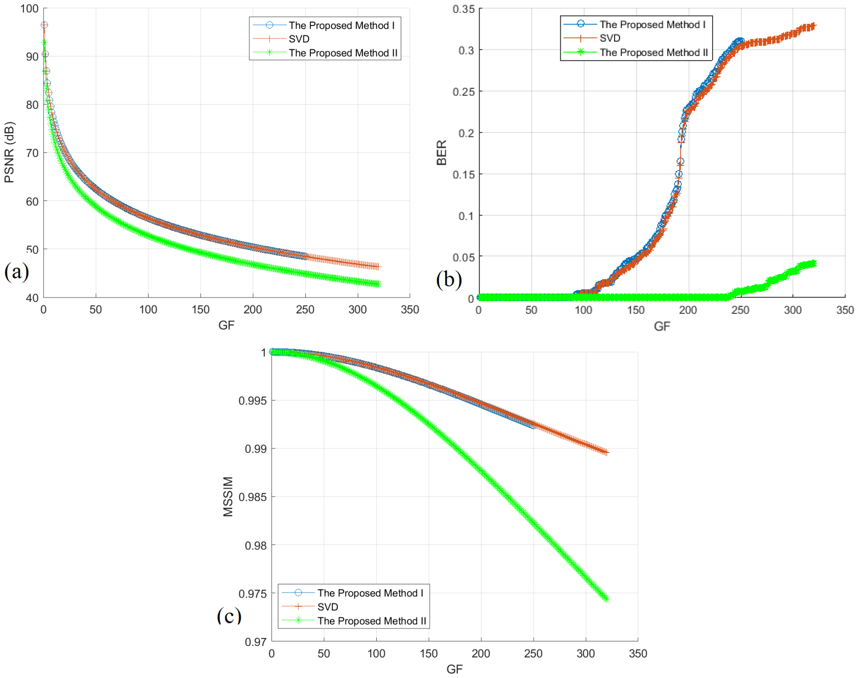


Figure 3.15: (a) PSNR (b) BER (c) MSSIM of the watermarked LF (man)

However, U and V are not exactly the same for the host and watermarked LF and hence, the hypothesis of error-free extraction holds for a range of gf values smaller than a given threshold. As it is evident from Fig. 3.15(b) and Fig. 3.16(b), for gf values less than a given threshold, the BER remains zero. As mentioned earlier, the vectors U and V of the host and watermarked LF are not identical. The bigger figures of gf causes such a big mismatching between U and V vectors of the host and watermarked LF that the BER never reaches zero. Nevertheless, letting $gf < 235$ for the man LF and $gf < 146$ for the toys LF, will cause the BER of the proposed method II remaining zero. Any gf beyond these ranges is not practical and does not make any sense to use such extreme gf figures. In other words, if the watermarked LF provides a good robustness against intended attacks, it

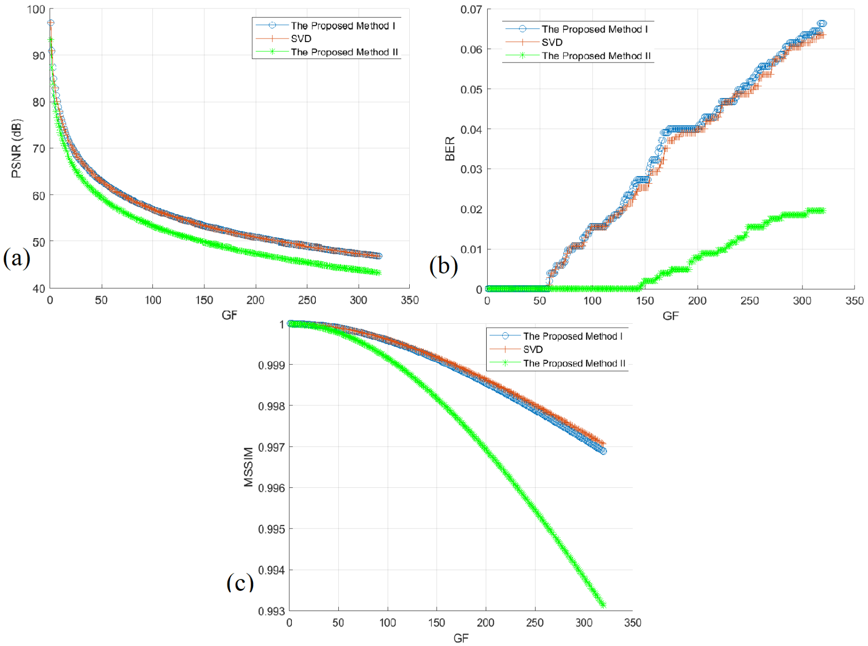


Figure 3.16: (a) PSNR (b) BER (c) MSSIM of the watermarked LF (toys)

does not make any sense to increase the gf and degrade the image quality further. Additionally, the maximum figures of BER for man LF and toys LF are 4.10% and 1.90%, respectively. Even though such extreme gf values are beyond practical range, the fact of quite low BER at very high values of gf indicates the accuracy of the extracted watermark regardless of the gain factor.

Figure 3.15(c) and Fig. 3.16(c) also show the $MSSIM$ of watermarked man and toys LFs. As is obvious from Fig. 3.15(c) and Fig. 3.16(c), the $MSSIM$ of the watermarked LF always remains higher than 0.974 and 0.993 for the man and toys LF respectively. Such high figures of $MSSIM$ signify an enormous fidelity between the structural similarity of host and watermarked LF.

Another prominent feature of the LF is to carry substantial angular

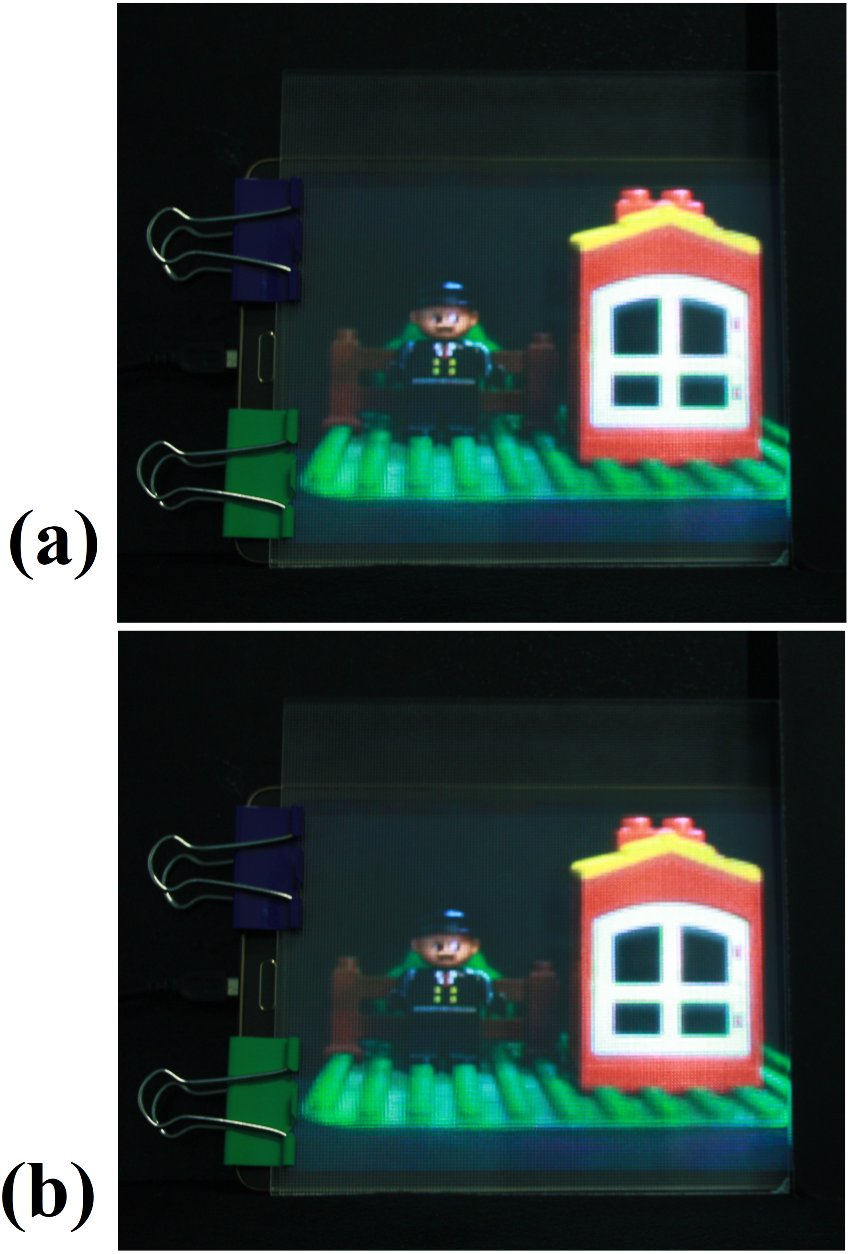


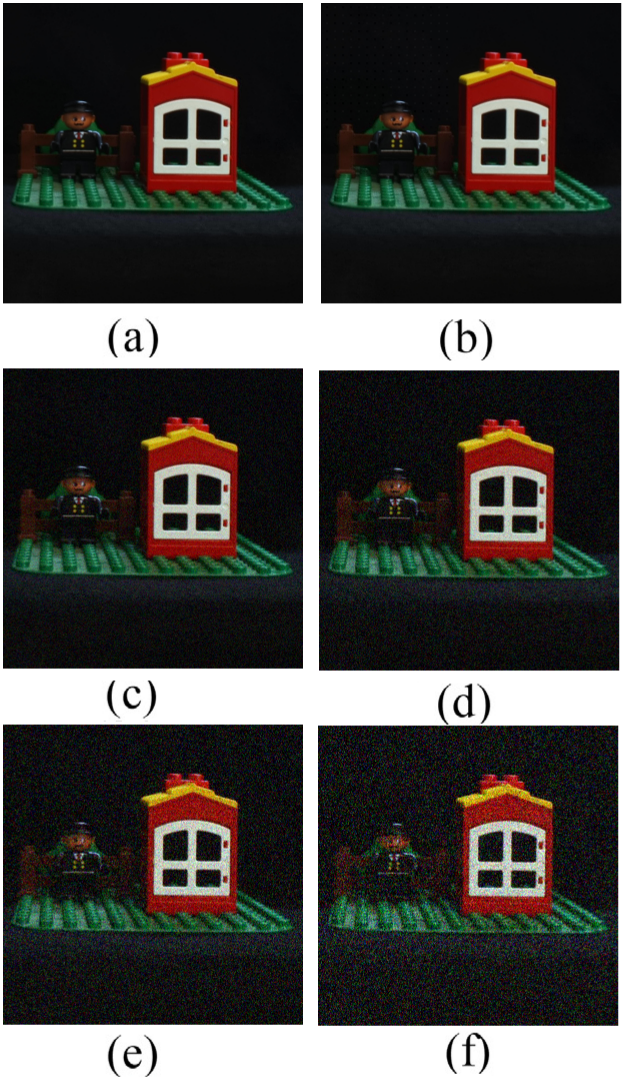
Figure 3.17: Projection of man LF into IIM. (a) The host LF (b) The watermarked LF.

information. As stated earlier, any watermarking scheme should promise full protection of angular information incorporated in the LF. As the *PSNR* and *MSSIM* are noticeably high, it makes perfect sense to assume that placing a micro-lens array (MLA) on the integral imaging monitor (IIM) will also yield a 3D presentation similar to original scene. In order to investigate the visual impact of the proposed watermarking method on 3D perception of the reconstructed LF, both host and watermarked LF were projected into an IIM. Figure 3.17 shows the projection of the host and watermarked LF into IIM. As expected, no difference (not even any minor one) was found between the two videos. Specially, the parallax of the host and watermarked LF seemed absolutely identical. This experiment was carried out only on the man LF. The video for the host and watermarked LF are available online.

3.6.3 Robustness Analysis

Gaussian Noise

As mentioned in section 2.6.4, the Gaussian noise is a common attack to the watermarking platforms and is very important in designing most transmission systems. Hence, the robustness of the proposed method against Gaussian noise has been verified. The extreme noise attacks can be very detrimental to the quality of the image. Figure 3.18 and Fig. 3.19 visualize the degradation of the LF images exposed to the Gaussian noise attack from moderate ($\sigma_n^2 = 100$) to extreme ($\sigma_n^2 = 1250$). Fig.3.20 shows the robustness of the watermarked LF (man) against Gaussian noise. As can be seen from Fig. 3.20, if the watermarked man LF is exposed to the moderate Gaussian noise of $\sigma_n^2 = 100$, the BER falls down to zero quite fast and for $gf > 35$, we have $BER = 0$. If the noise power increases to $\sigma_n^2 = 225$, for $gf > 44$ we have $BER < 0.09\%$ which is absolutely a negligible error. Even so, for $gf > 63$ the BER falls to zero. SVD and method I also follow a similar trajectory. If the noise power rises 625, for $gf > 60$, the BER of the method II falls below 1%. Provided that $\sigma_n^2 = 625$, the BER of the SVD method never falls below 30%. In case of excessive noise attack of $\sigma_n^2 = 1225$, the BER of the method II will be less than 1% for $gf > 88$



(a) (b) (c) (d) (e) (f)

Figure 3.18: The noisy watermarked LF(man) (a) The host image (b) The watermarked image ($gf = 90$ and $n\_dct = 6$) (c) $\sigma_n^2 = 100$; (d) $\sigma_n^2 = 225$; (e) $\sigma_n^2 = 625$; and (f) $\sigma_n^2 = 1225$.

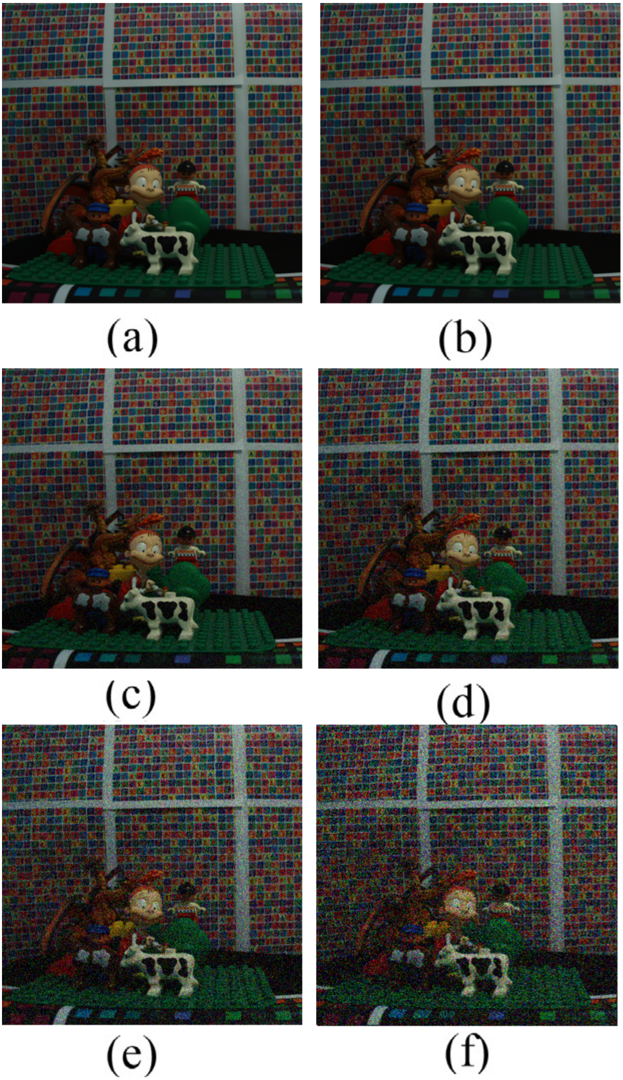


Figure 3.19: The noisy watermarked LF (toys) (a) The host image (b) The watermarked image ($gf = 90$ and $n\_dct = 6$) (c) $\sigma_n^2 = 100$; (d) $\sigma_n^2 = 225$; (e) $\sigma_n^2 = 625$; and (f) $\sigma_n^2 = 1225$.

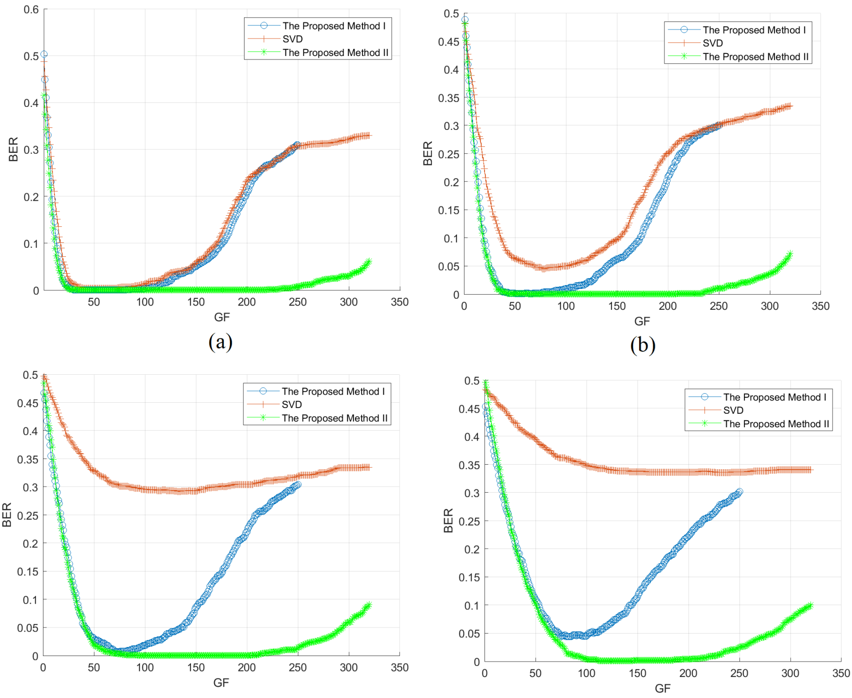
whereas the *BER* of the SVD method never falls below 33%. The method I never attains any *BER* less than 4.49%. Generally speaking, for the moderate noise attack (noise power of 100, 225 and even 625) the proposed method II showed no noticeable advantage over the proposed method I. However, in case of very aggressive noise attack with noise power of 1225, the robustness of the proposed method is slightly higher. As anticipated, SVD method (as described on section2.6) delivers very poor results not comparable with any of the proposed methods. That is to say, the exploitation of color conversion, 4D-DWT and DCT contribute to the robustness of the watermarked man LF against Gaussian noise.

Figure 3.20 shows the robustness of the watermarked toys LF against Gauusian noise. In case of the toys LF, the difference between the proposed method I and II is visible as well. Again, for more aggressive noise power of 1225, the method II outperforms the method I noticeably. One can deduce that the general trajectory of the *BER* of the toys LF exposed to the Gaussian noise, is approximately the same as man LF, but the gap between the performances is quite less. However, when more intense noise attack occurs, the superiority of the method II is more pronounced.

Even though the performance of SVD method is inferior to the proposed methods, the difference is quite smaller in comparison with man LF. This may be justified by the plentiful high frequency components of the toys LF. Both of the proposed methods use a watermarking feature which is profoundly associated with low frequency components. It is very well established in the literature that a vast majority of contents of natural scenes lies into the low frequency section of the spectrum. As the toys LF has a good deal of high frequency content, SVD will capture such information and can approach the proposed methods with a lower difference than the man LF.

JPEG Compression

JPEG compression is another common attack which widely occurs in most platforms. Hence, the robustness of the proposed watermarking method against JPEG compression is absolutely essential. Having that said, we investigated the impact of JPEG compression on the accuracy of water-

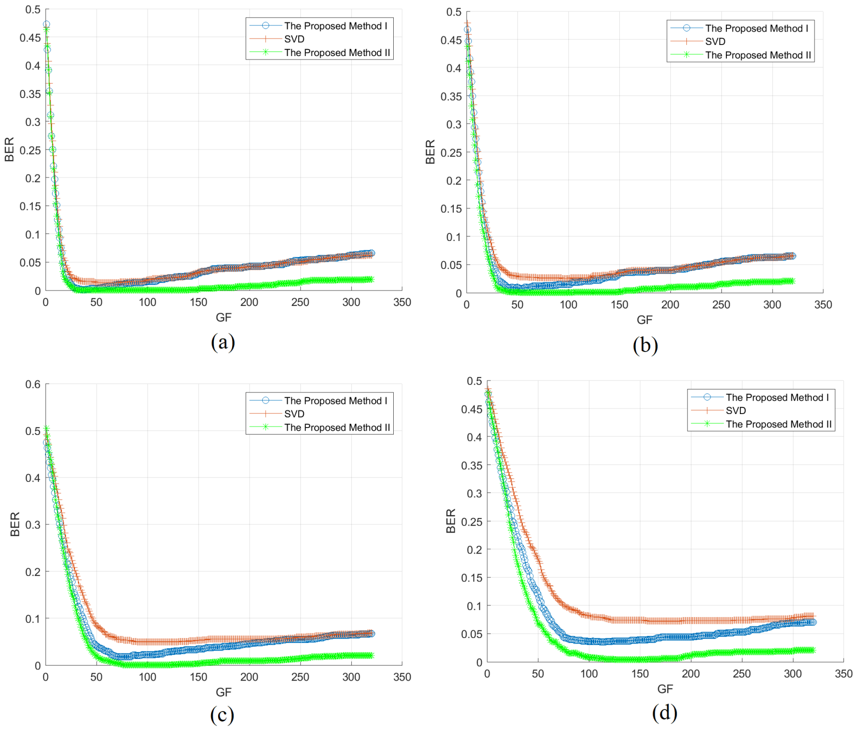


(a) (b) (c) (d)

Figure 3.20: The BER of watermarked LF (man) exposed to Gaussian noise (watermark dimensions 32×32*) (a)* $\sigma_n^2 = 100$ *; (b)* $\sigma_n^2 = 225$*; (c)* $\sigma_n^2 = 625$ *; and (d)* $\sigma_n^2 = 1225$.

mark extraction. A wide range of quality factors have been used for JPEG compression namely 5%, 25%, 50%, 75% and 100%. As is evident from Fig. 3.22 and Fig. 3.23, compression with $qf = 5\%$ has such a devastating effect on watermarked LF that it will be severely degraded. As the watermarked LF is entirely degraded by an aggressive compression, the embedded watermark will also be affected as well as the entire visual content of the watermarked LF.

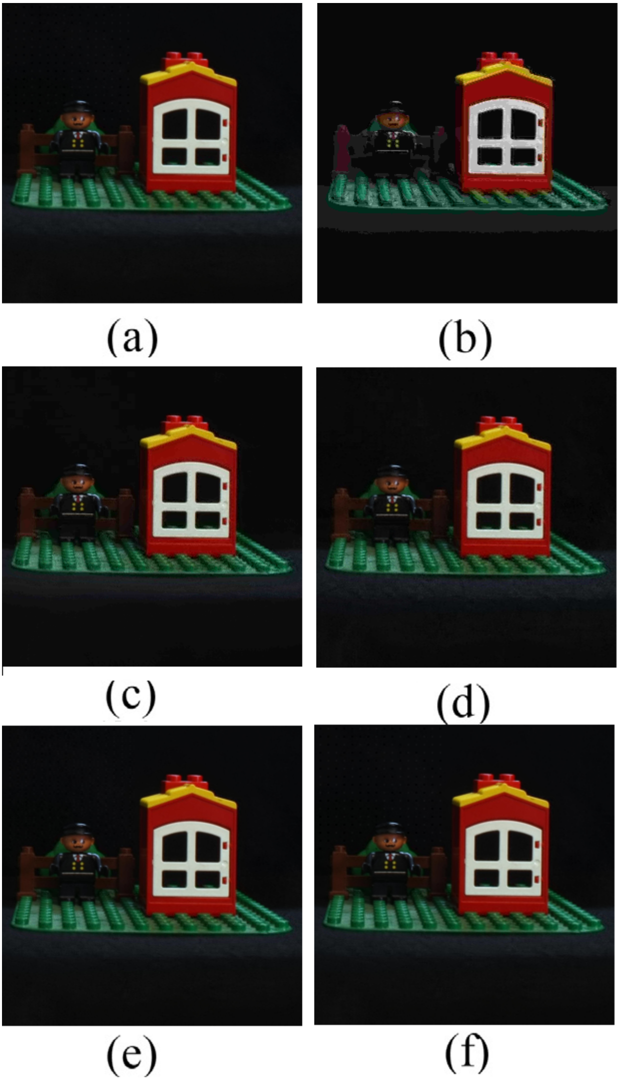
Unsurprisingly, for the man LF, the *BER* is high even though the proposed method II gives a *BER* slightly lower than the proposed method



(a) (b) (c) (d)

Figure 3.21: The BER of watermarked LF (toys) exposed to Gaussian noise(watermark dimensions 32×32*) (a)* $\sigma_n^2 = 100$ *; (b)* $\sigma_n^2 = 225$*; (c)* $\sigma_n^2 = 625$ *; and (d)* $\sigma_n^2 = 1225$*.)*

I and SVD. As can be seen from Fig. 3.24, once the qf rises to 25%, the supremacy of the proposed method II is pronounced more and achieves $BER < 3\%$. Letting $qf = 50\%$, the BER falls down very rapidly and for a wide range of $gf (44 \leq gf < 123)$ the BER remains less than 12% and finally converges to zero. In case of JPEG compression with $qf = 75\%$, the BER graph converges to zero more quickly. The proposed method I and SVD, reach a BER figure no better than 46% and 45% respectively. Such high BER implies a substantial uncertainty in extracted watermark bit almost as much as tossing a fair coin and measuring the probability of



(a) (b) (c) (d) (e) (f)

Figure 3.22: *The compression of watermarked LF (man) with JPEG compression (a) the watermarked image (b) qf = 5% (c) qf = 25% (d) qf = 50% (e) qf = 75% (f) qf = 100%.*

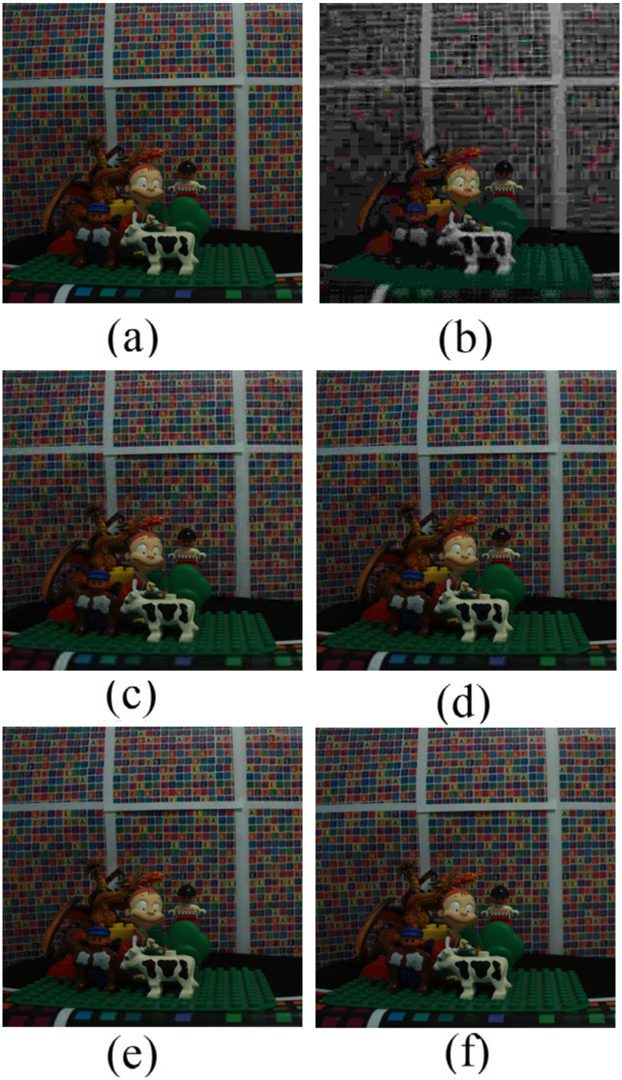
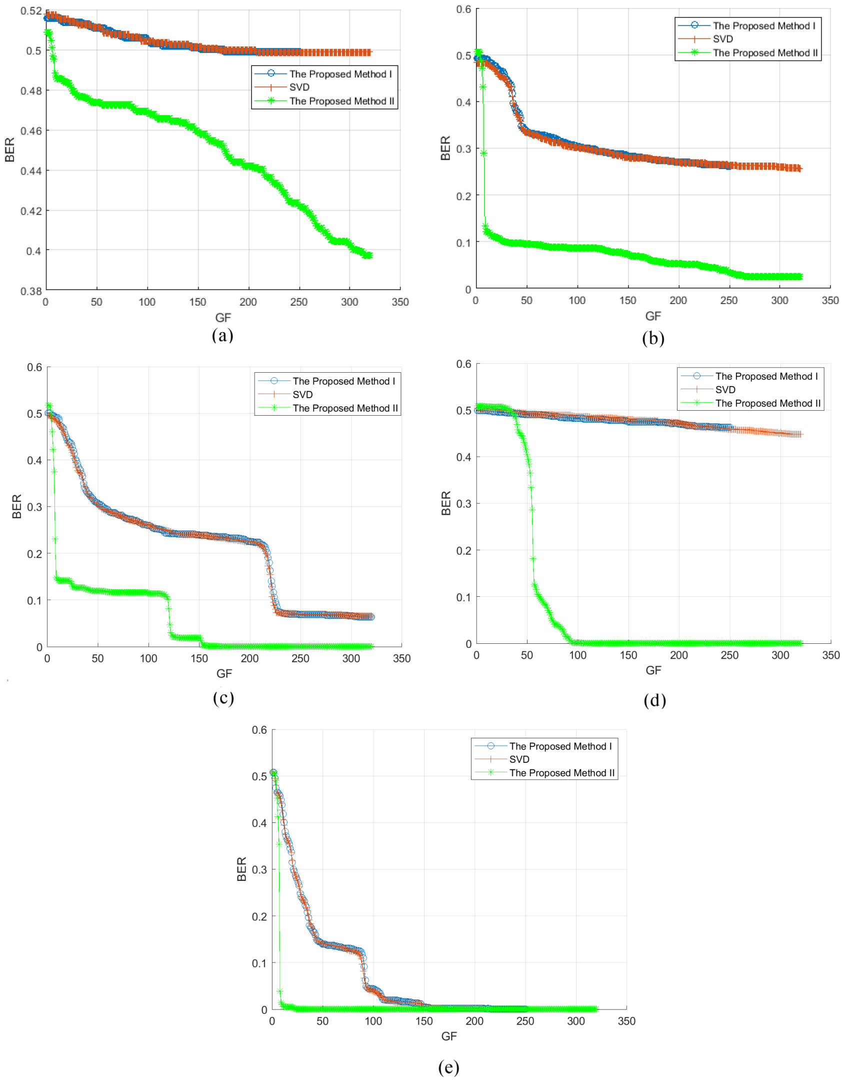
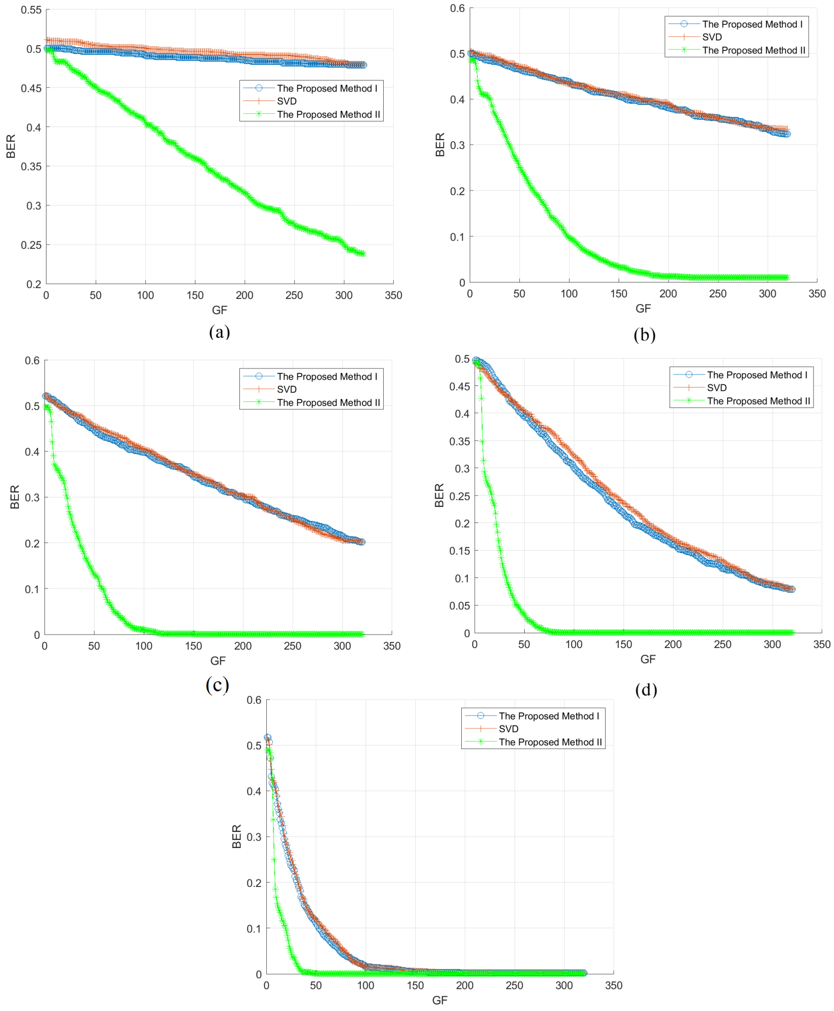


Figure 3.23: The compression of watermarked LF (toys) with JPEG compression (a) the watermarked image (b) qf = 5% (c) qf = 25% (d) qf = 50% (e) qf = 75% (f) qf = 100%.



(a) (b) (c) (d) (e)

Figure 3.24: The BER of watermarked LF (man) compressed with JPEG (a) $qf = 5\%$ (b) $qf = 25\%$ (c) $qf = 50\%$ (d) $qf = 75\%$ (e) $qf = 100\%$



(a)

(b)

(c)

(d)

(e)

Figure 3.25: The BER of watermarked LF (toys) compressed with JPEG (a) $qf = 5\%$ (b) $qf = 25\%$ (c) $qf = 50\%$ (d) $qf = 75\%$ (e) $qf = 100\%$

getting heads or tails. Conversely, the proposed method delivers a BER significantly lower approaching 1.37%. The last compression test was carried out using $qf = 100\%$. The qf of 100% is commonly misunderstood as lossless despite the fact that $qf = 100\%$ still imposes some loss. Letting $qf = 100\%$, the BER of the proposed method II swiftly (for $11 \leq gf$) falls down below 1%. Letting $24 \leq gf$, the BER falls down into zero which is by far better than the proposed method I and SVD which achieve $BER < 1\%$ under $148 \leq gf$.

As is apparent from Fig. 3.25, a similar trend is observed for BER of toys LF compressed with JPEG. It is noticeable that for lower qf figures of $25\%, 50\%$ and 75%, the proposed method II has clear advantage over the proposed method I and SVD. The proposed method II exhibits a promising performance. Letting $qf = 25\%, 50\%$ and 75%, the BER of the proposed method II is considerably lower and converges to zero much faster than the proposed method I and SVD. The toys LF has much more high frequency components and JPEG compression may cause a more severe loss, as is obvious from Fig. 3.23(b). In comparison to Fig. 3.22(b), the degradation is so enormous that the background is entirely scrambled.

Median Filtering

The median filtering is typically utilized to remove the salt and pepper noise. Figure 3.26 and Fig. 3.27 show the BER of watermark extraction from watermarked LF passed through median filters. The proposed method II delivers more robustness against median filtering than the proposed method I and SVD. Letting $gf = 90$, the BER of watermark extraction from the man and toys LF is 8.20% and 21.58% respectively. The higher BER of the toys LF may be contributed to the substantial high frequency components.

JPEG 2000 Compression

It is no secret that JPEG 2000 provides much higher quality than JPEG at the same compression ratio (CR). Even with higher CR, JPEG 2000 usually yields a higher quality [153]. While JPEG is heavily relied upon DCT,

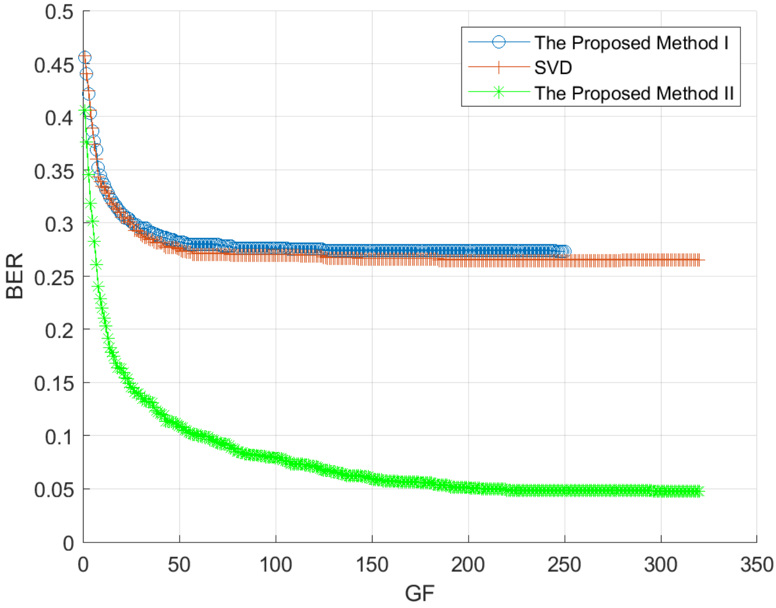


Figure 3.26: The BER of watermarked LF (man) passed through median filter.

JPEG 2000 employs wavelet to avoid blockwise distortion [139]. Having that said, along with exploitation of wavelet transform in our watermarking platform, we were intrigued to investigate the effect of JPEG 2000 compression on watermark extraction. Unlike the qf of JPEG compression, the CR is inversely proportional to the visual quality of compressed image. In this simulation, the CR was set 10, 20, 30 and 40.

Figure 3.28 and Fig. 3.29 show the watermarked LF of man and toys. As is evident from Fig. 3.28 and Fig. 3.29, JPEG 2000 has a higher quality than JPEG. Figure 3.30 shows BER of the watermarked man LF compressed by JPEG 2000 using different CR values. For low to moderate CR of 10 and 20, both of the proposed methods converge to zero quite fast. However, letting $CR = 20$, the proposed method converges to zero slightly faster than the proposed method I and SVD. Letting $CR = 30, 40$, the BER

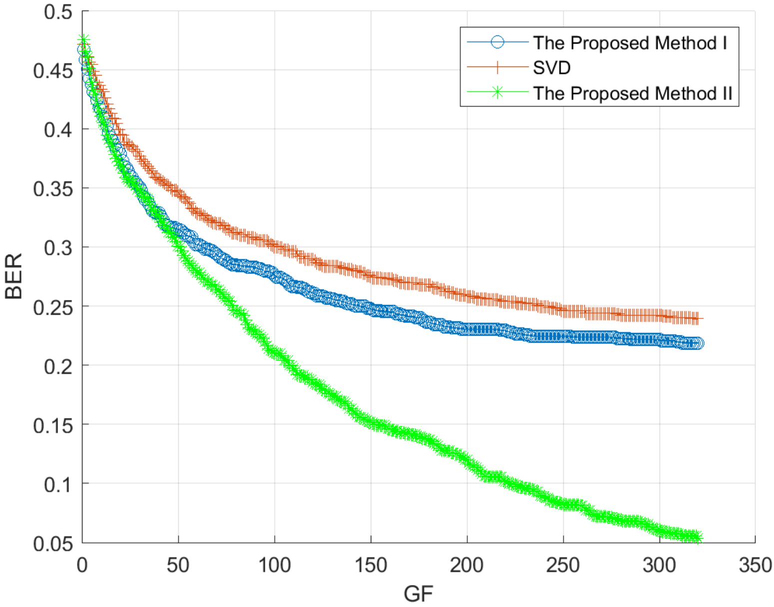


Figure 3.27: The BER of watermarked LF (toys) passed through median filter.

converges zero noticeably faster than the other two methods.

As is aparent from Figure 3.31, the same trend is also observed for the toys LF. However, the advantage of the proposed method II is more pronounced. The *BER* gap between the proposed method II and the two other methods is prominently larger than the man LF. In case of $CR = 30,\ 40$ the absolute dominance of the proposed method II is even more obvious. It is also worth mentioning that the proposed method I and SVD fail to achieve $BER = 0$ for $CR = 30, 40$.

3.6.4 Integrity analysis

As mentioned in section 3.4, the proposed method consists of multiple transforms namely color conversion, 4D wavelet, DCT, and SVD. The poor

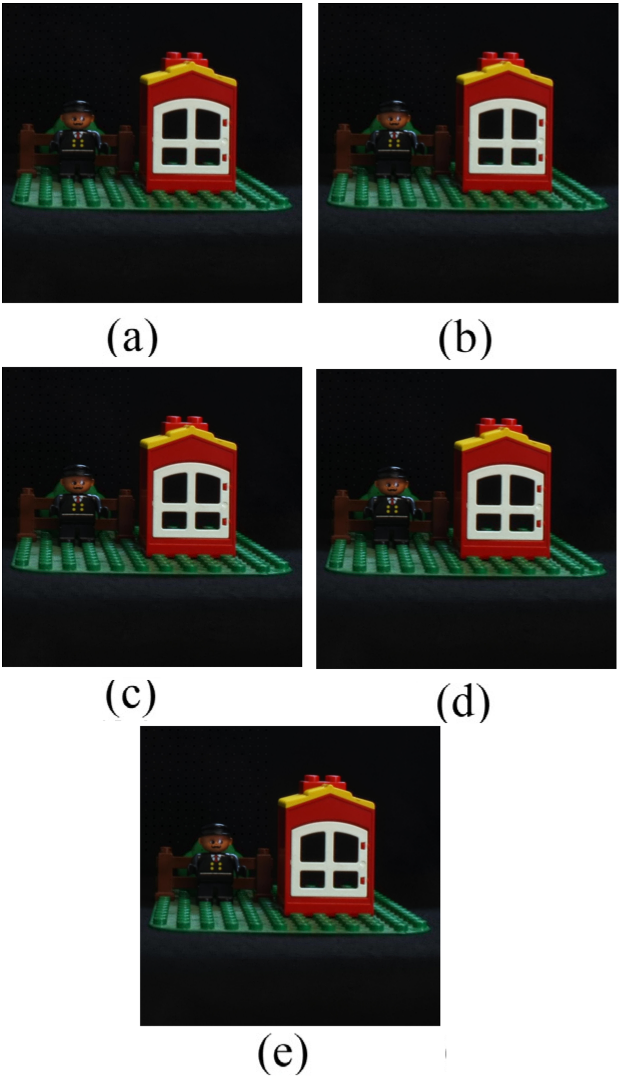


Figure 3.28: The compression of watermarked LF (man) with JPEG 2000 compression: (a) the watermarked image; (b) CR= 10; (c) CR= 20; (d) CR= 30; (e) CR= 40.

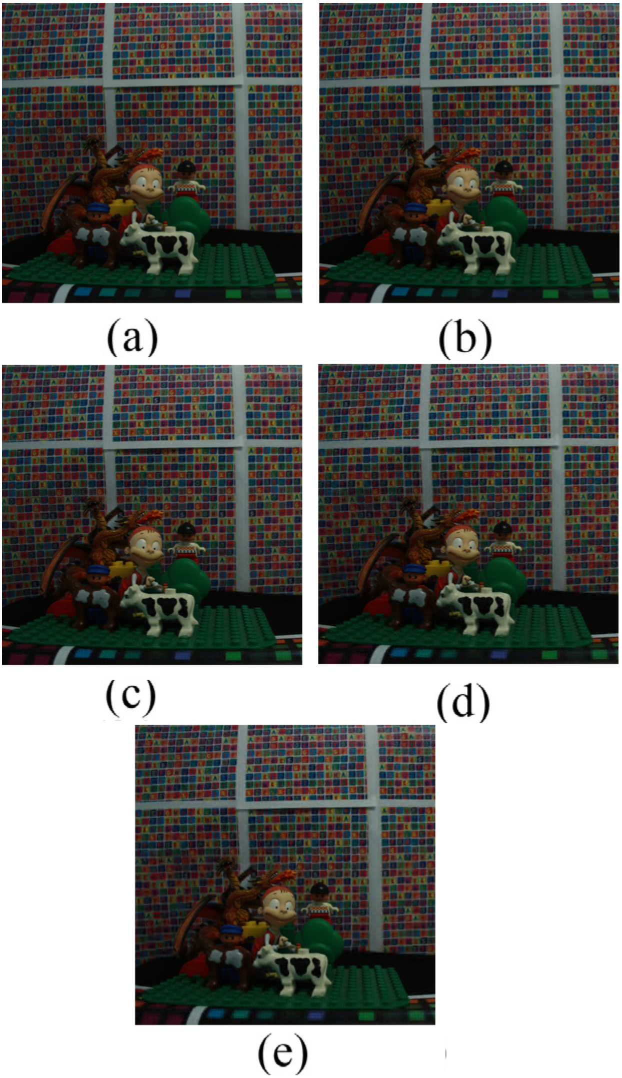
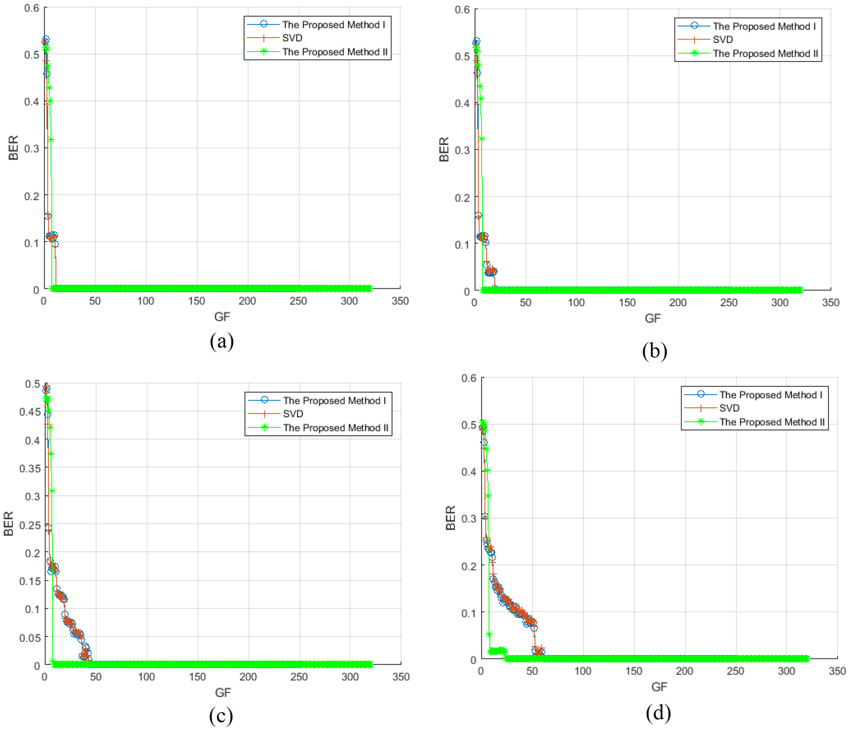


Figure 3.29: The compression of watermarked LF (toys) with JPEG 2000 compression: (a) the watermarked image; (b) CR= 10; (c) CR= 20; (d) CR= 30; (e) CR= 40.



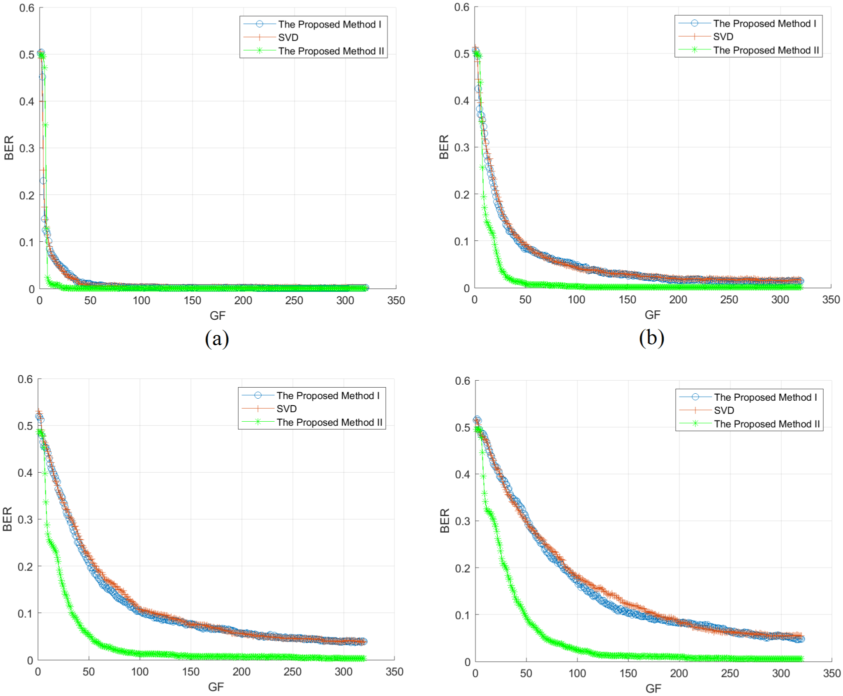
(a) (b) (c) (d)

Figure 3.30: The BER of watermarked LF (man) compressed with JPEG 2000: (a) CR= 10; (b) CR= 20; (c) CR= 30; (d) CR= 40.

results of the SVD method was demonstrated in section 3.6.2. Hence, the results of the SVD method are not discussed in this section. Besides, the inferior results of the SVD method makes other methods to seem following the same trajectory. The exclusion of the SVD method leads to the finer scale of the ordinate making the comparison of other methods more insightful.

This section is intended to answer the following question: is it really possible to attain the optimal performance despite the removal of one of the transforms?

As mentioned in section 1.4, the LF has significantly higher dimen-



(a) (b) (c) (d)

Figure 3.31: The BER of watermarked LF (toys) compressed with JPEG 2000: (a) CR= 10; (b) CR= 20; (c) CR= 30; (d) CR= 40.

sionality than the common 2D images. Each of the individual transforms serve to decrease the correlation along one of the dimensions which is not possible by other transforms.

- The color conversion from RGB to YUV is vital to minimize the tremendous interchannel correlation of RGB channels. Such enormous correlation cannot be decreased neither by applying a transform into individual *EI*s (4D wavelet or DCT) nor the sub-regions of the transformed *EI*s. Applying the transform to the μIs will not result in reduction of the interchannel correlation either.

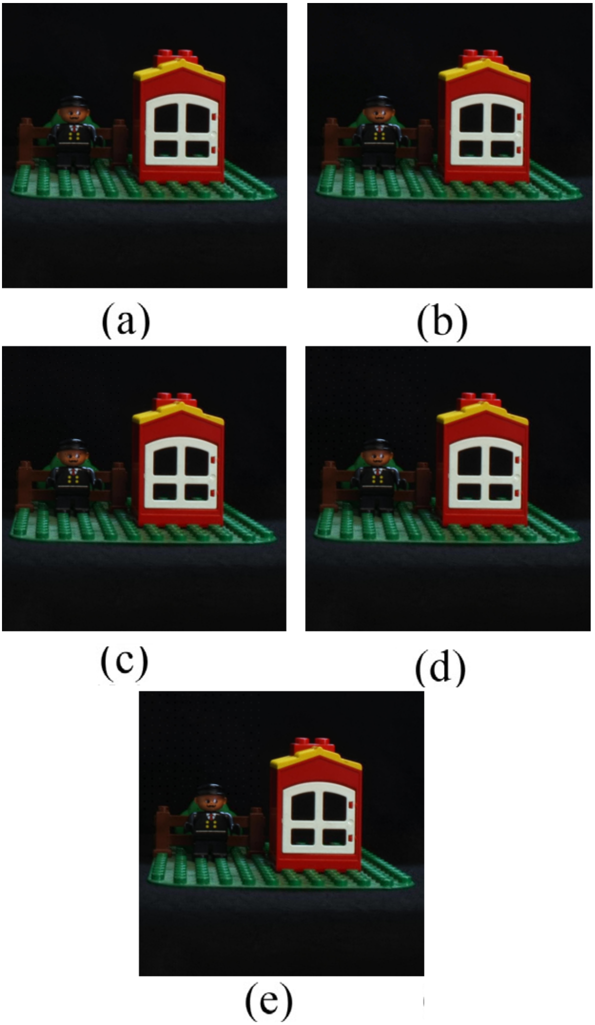


Figure 3.32: Illustration of the impact of different transforms on the visual appearance of the central view of watermarked man LF. (a) The host LF. Watermarked LF : (b) without color conversion; (c) without 4D wavelet; (d) without DCT; (e) the proposed method.

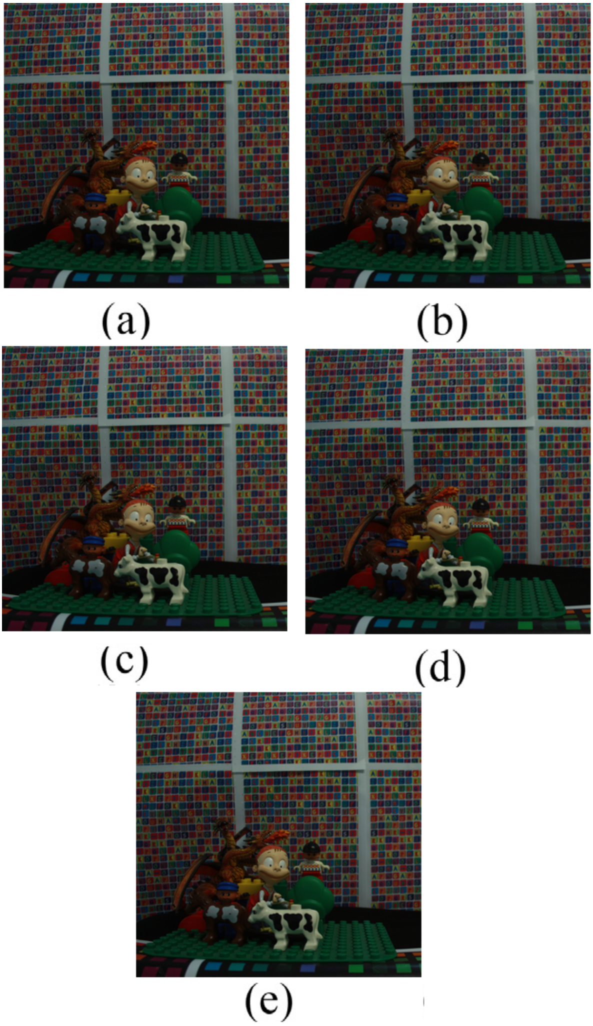


Figure 3.33: Illustration of the impact of different transforms on the visual appearance of the central view of watermarked toys LF. (a) The host LF. Watermarked LF : (b) without color conversion; (c) without 4D wavelet; (d) without DCT; (e) The proposed method.

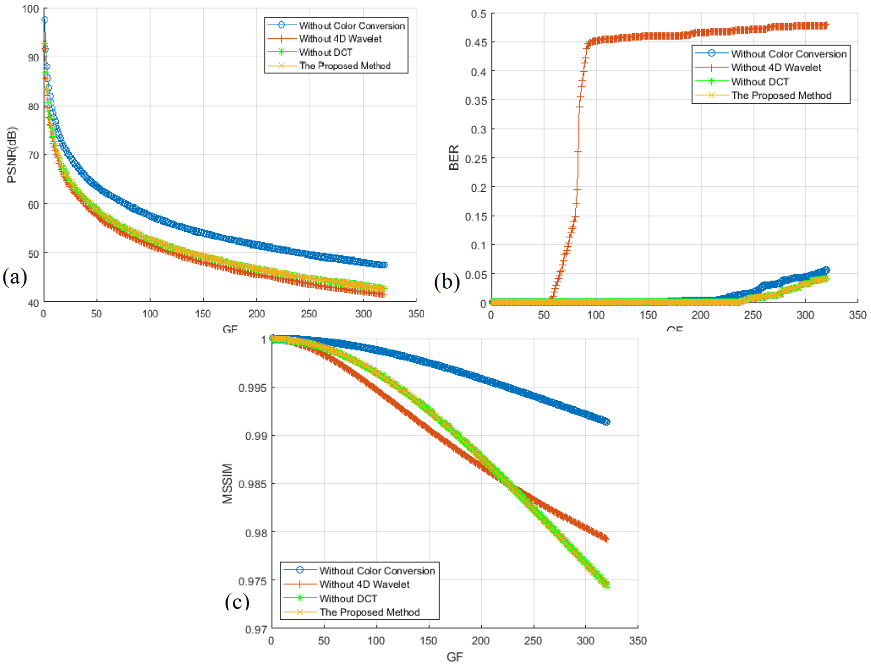


Figure 3.34: The impact of individual transforms on (a) PSNR, (b) BER, and (c) MSSIM of the watermarked LF (man).

- The 4D wavelet transform is used to reduce the huge intercorrelation among neighboring EIs and μIs. Additionally, the 4D wavelet also decreases *intracorrelation* of the EIs and μIs.

  The high *intercorrelation* among EIs occurs horizontally, vertically and diagonally. For example, the intercorrelation between $EI(i,j)$ and each of $EI(i,j\pm k)$, $EI(i\pm k,j)$, $EI(i\pm k,j\pm k)$ is inversely proportional to $k\in\mathbb{N}$. No matter if the transform is applied to an entire EI or some regions, the intercorrelation of the individual EIs cannot be reduced by applying any transform to each of the tripple channels of the EIs. The very same argument also holds about μIs. A 4D transform is required to fulfill the decrease of the horizontal, vertical and diagonal correlation of the μIs. A 4D transform is essential for

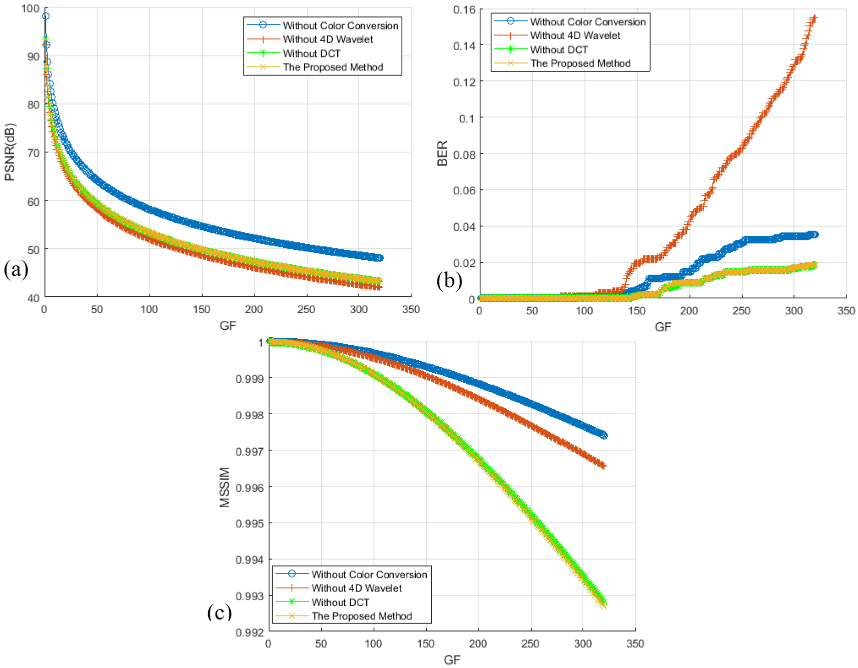
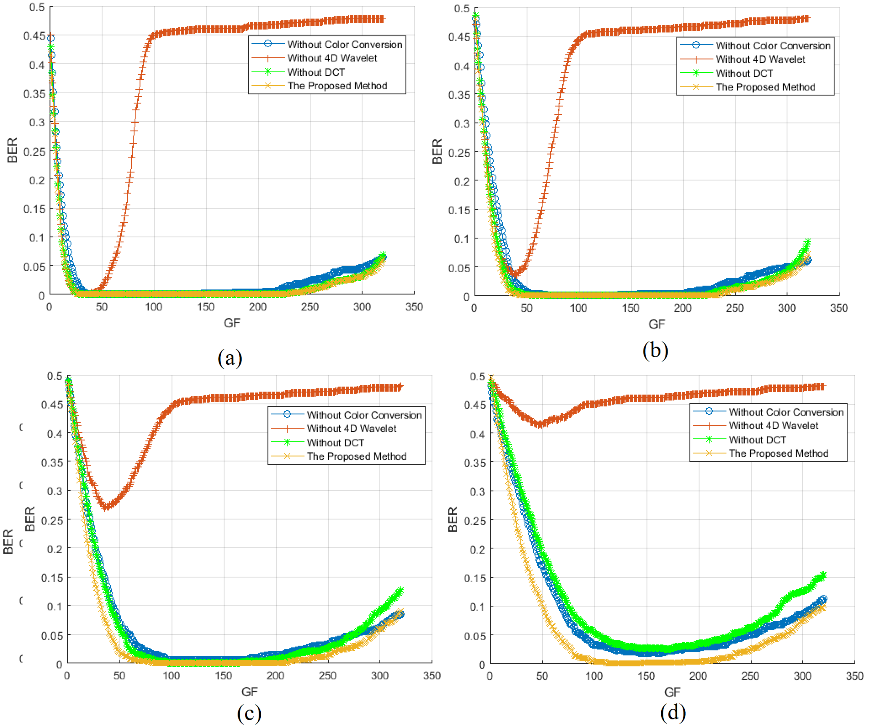


Figure 3.35: The impact of individual transforms on (a) PSNR, (b) BER, and (c) MSSIM of the watermarked LF (toys).

decorrelationg the μIs horizontally, vertically and diagonally which is impossible to attain by applying any kind of transform into entire or sub-regions of the μIs. This is why it is so important to utilize 4D wavelet for significant decorrelation of *spatial* and *angular* information carried by LF.

- DCT is employed to increase the robustness of the proposed method. The DCT has a property of compacting the signal energy in some few coefficients. Applying the DCT into wavelet coefficients makes the watermarking feature even robuster. A vast majority of the natural scenes have an enormous portion of low frequency information and the usage of the DCT to reinforce the watermarking feature makes



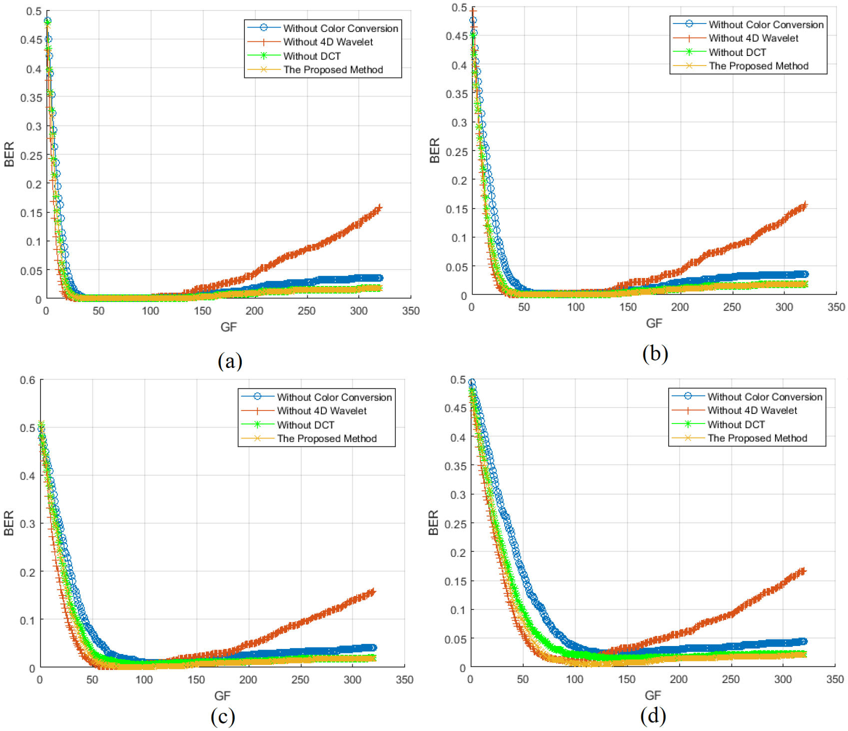
(a) (b) (c) (d)

Figure 3.36: The BER of watermarked LF (man) exposed to Gaussian noise: (a) $\sigma_n^2 = 100$; (b) $\sigma_n^2 = 225$; (c) $\sigma_n^2 = 625$; and (d) $\sigma_n^2 = 1225$.

perfect sense.

- SVD is essential to extract a robust watermarking feature. The usage of SVD is inevitable in both embedding and extraction procedures.

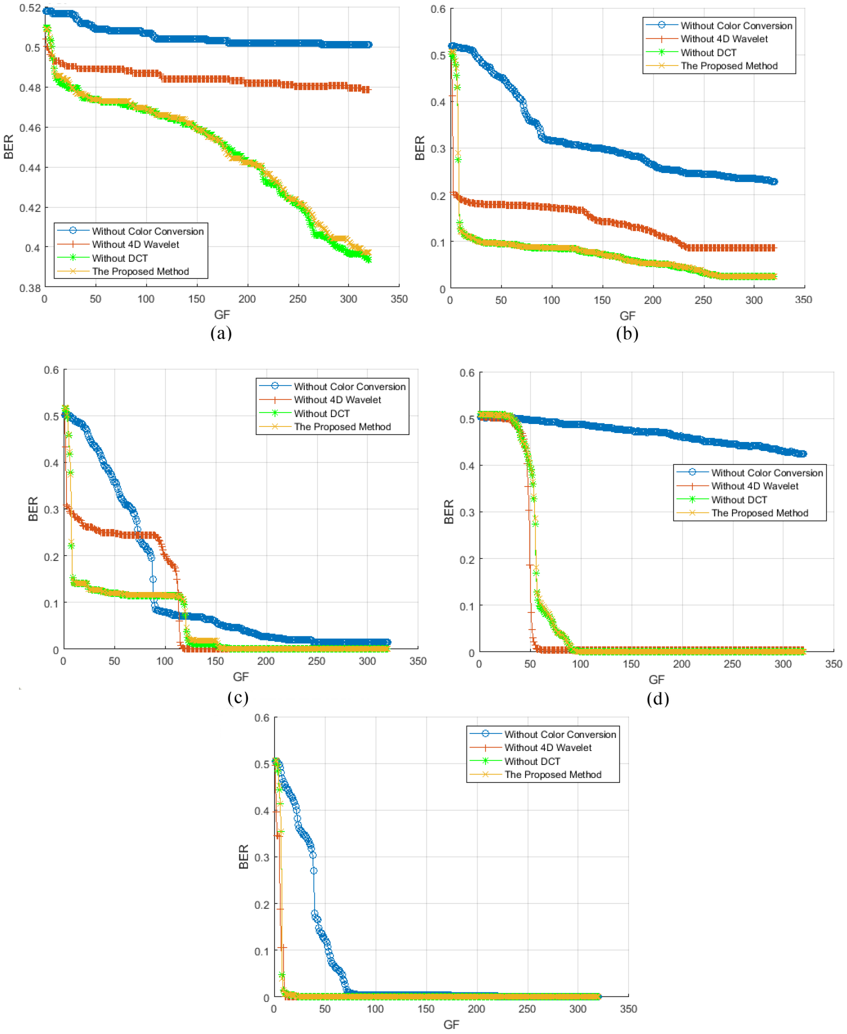
Apparently, each transform decorrelates the LF along a specific dimension. The removal of any transform (specifically the 4D wavelet) deteriorates the performance of the proposed method. In other words, the loss of the performance through the removal of any transform cannot be compensated by the other remaining transforms. The impact of every single transform is investigated by an experiment. The experiment was conducted to verify the



(a) (b) (c) (d)

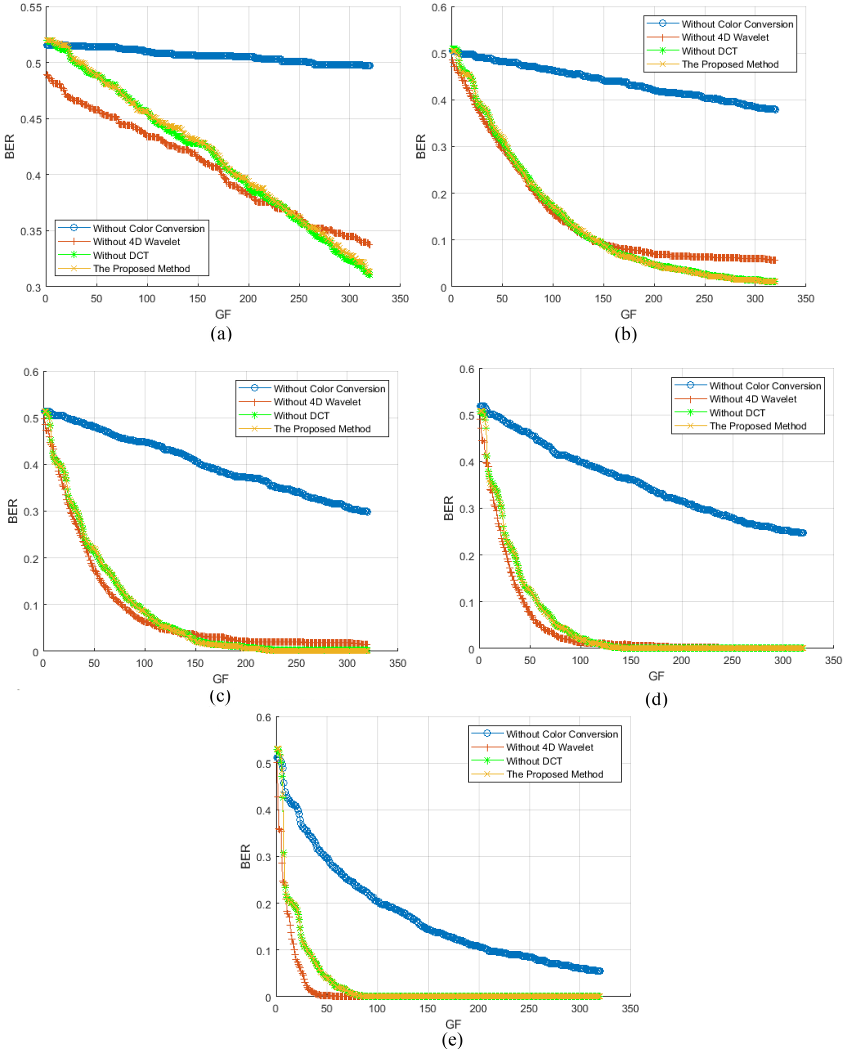
Figure 3.37: The BER of watermarked LF (toys) exposed to Gaussian noise: (a) $\sigma_n^2 = 100$ *; (b)* $\sigma_n^2 = 225$*; (c)* $\sigma_n^2 = 625$ *; and (d)* $\sigma_n^2 = 1225$.

contribution of each transform to the robustness of the watermark. The insertion and the extraction of the watermark was performed excluding each of the aforementioned transforms while retaining all the others. Therefore, the first experiment was carried out without color conversion. Likewise, the next experiment was performed without 4D wavelet and finally the last one was done without employing DCT. As is obvious from Fig. 3.32 and Fig. 3.33, the exclusion of individual transforms causes no perceptible visual difference. However, we will shortly see that the exclusion of the transforms has a considerable effect on the robustness of the proposed method against different attacks. Figure 3.34 and Fig. 3.35 show *PSNR*, *BER* and



(a) (b) (c) (d) (e)

Figure 3.38: The BER of the extracted watermark against JPEG compression (man): (a) qf = 5%; (b) qf = 25%; (c) qf = 50%; (d) qf = 75%; (e) qf = 100%.



(a) (b) (c) (d) (e)

Figure 3.39: The BER of the extracted watermark against JPEG compression (toys): (a) qf = 5%; (b) qf = 25%; (c) qf = 50%; (d) qf = 75%; (e) qf = 100%.

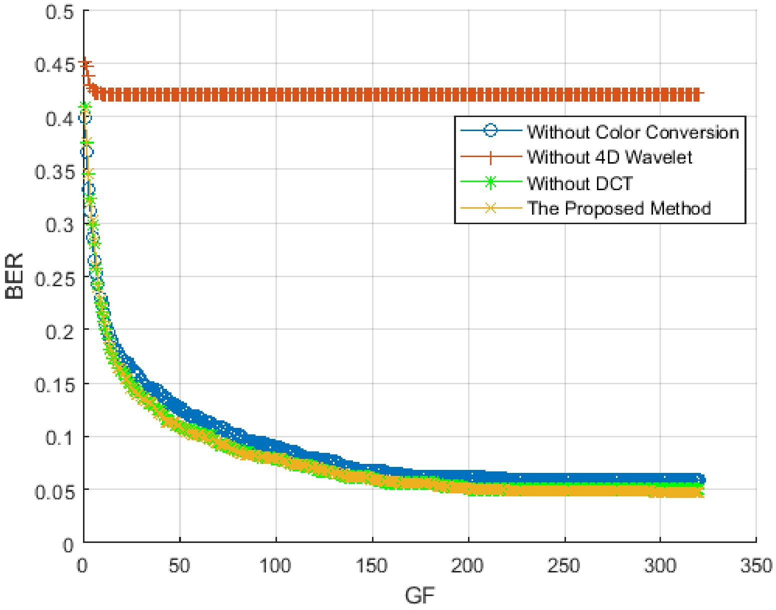


Figure 3.40: The BER of watermarked LF (man) passed through median filter.

MSSIM in different configurations. As can be seen from Fig. 3.34 and Fig. 3.35, the exclusion of color conversion boosts the *PSNR*. Similar to the argument made in section 3.6.2, the achieved *PSNR* figure is by far high enough for HVS to regard the watermarked LF identical to the host LF. Therefore, the higher *PSNR* figure obtained by the removal of the color conversion makes no perceptible visual difference with other methods. For $gf < 159$, the *MSSIM* always remains higher than 0.99.

Figure 3.36 and Fig. 3.37 demonstrate the robustness of the mentioned methods against Gaussian noise. If the noise attack is not extreme, then the performance of the aforementioned methods are quite similar. Nevertheless, once the watermarked LF is exposed to more agressive noise attacks, the influence of the integrity gets highlighted. Letting the power of Gaussian noise be 100, 225, no major difference is observed among

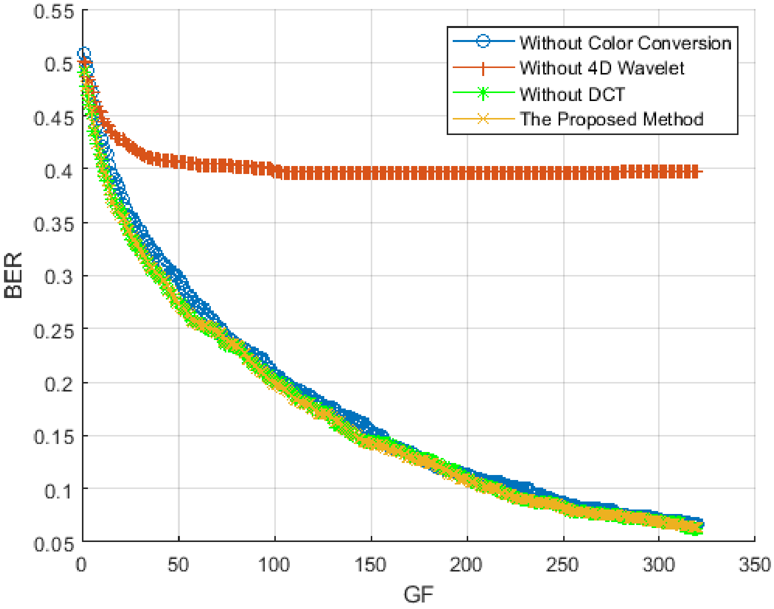
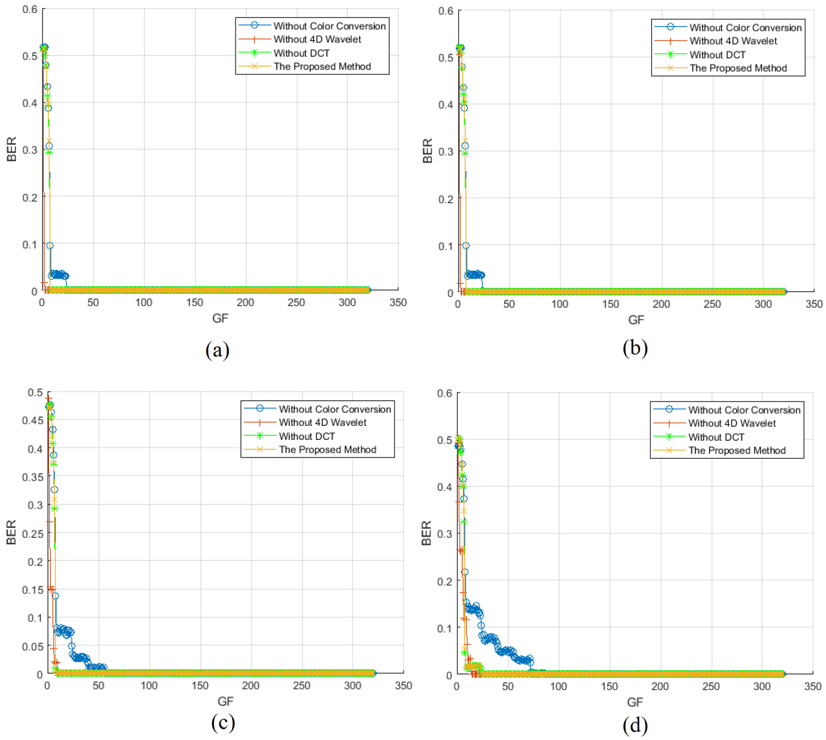


Figure 3.41: The BER of watermarked LF (toys) passed through median filter.

the*BER* of the the extracted watermark for the man LF. Once the noise power rises to 625, the importance of using all transformations becomes more pronounced. When the watermarked LF is exposed to more aggressive Gaussian noise attack with power of 1225, the superiority of the proposed method is emphasized. Letting $gf = 80$, the removal of the color conversion and DCT increases the *BER* by 4.88% and 6.48% respectively, while removing the 4D wavelet will cause the *BER* to rise 42.28%. In case of the toys LF, the difference of the aforementioned methods is a bit lower but the results of the proposed method is still better than any other method lacking one of the transforms. The lower difference between these methods can be contributed to the larger portion of the high frequency content of the toys LF.

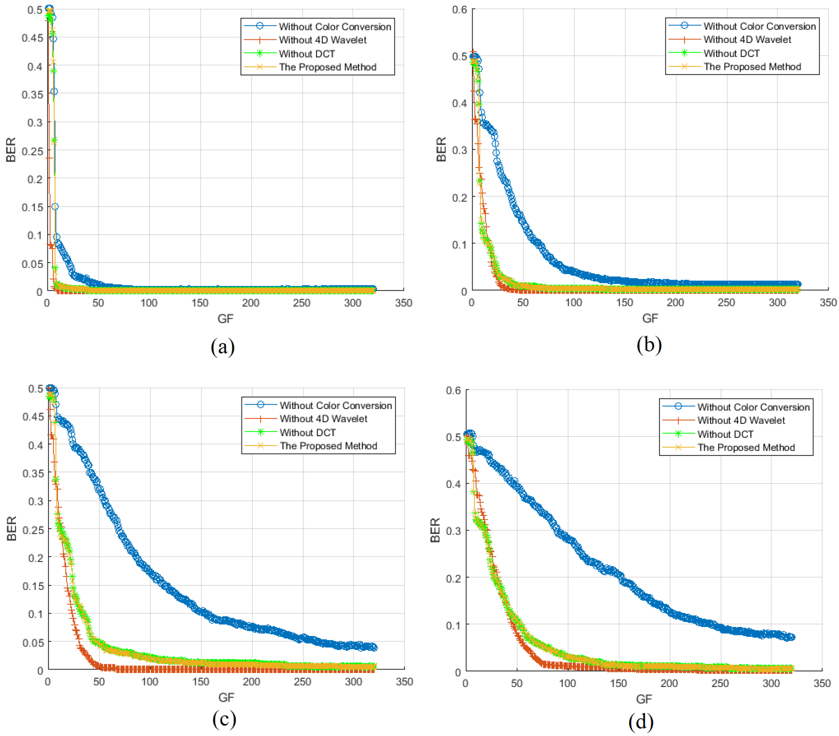
Figure 3.38 and Fig. 3.39 illustrate the robustness of the aforemen-



(a) (b) (c) (d)

Figure 3.42: The BER of the extracted watermark against JPEG 2000 compression (man): (a) CR= 10; (b) CR= 20; (c) CR= 30; and (d) CR= 40.

tioned methods against JPEG compression. As expected, in case of JPEG compression with $qf = 100\%, 75\%$, the both man and toys LF are minimally degraded. However, the removal of color conversion causes substantially bigger BER. Except the color conversion, it was observed that the exclusion of other transforms did not cause noticeable difference on the BER of the aforementioned methods for none of the LFs. Letting $gf = 80$, the BER of all the mentioned methods would be zero for the man LF. The BER of the proposed method for the toys LF is 3.90% which can easily be tolerated. When it comes to $qf = 75\%$, the BER of the proposed method is 4.00% for the man LF and 4.59% for the toys LF. Either BER figures are



(a) (b) (c) (d)

Figure 3.43: The BER of the extracted watermark against JPEG 2000 compression (toys): (a) CR= 10; (b) CR= 20; (c) CR= 30; and (d) CR= 40.

significantly lower than the results obtained by removing the color conversion. Similarly, the removal of the other transformations will solely deteriorate the robustness of the proposed method against JPEG compression with $qf = 25\%$, 50%. Even though the removal of the 4D wavelet may improve the *BER* of the proposed method against JPEG compression with $qf = 75\%, 100\%$ infinitesimally, it introduces enormous loss in case of more agressive compression ratio. The lower the quality factor of JPEG compression is, the higher figure of the *BER* will occur due to exclusion of the 4D wavelet and color conversion. Letting $gf = 80, qf = 50\%$ the proposed method delivers *BER* of 11.72% for the man LF while exclusion of the color conversion and 4D wavelet results in BER of 22.17% and 24.61%. Under the same condition, the proposed method achieves $BER = 11.72\%$ for the toys LF while the removal of the color conversion and 4D wavelet yields BER of 22.17% and 24.61% respectively.

For JPEG compression with $qf = 25\%$ and $gf = 80$, the BER of the man LF is 8.89% while the removal of the color conversion and 4D wavelet lead to BER of 35.74% and 17.68% respectively. Given the considerable amount of high frequency contents of the toys LF, the proposed method delivers a BER of 22.07% whereas removing color conversion and 4D wavelet result in $BER = 47.17\%$ and 21.00% respectively. The effect of JPEG compression with $qf = 5\%$is so destructive that even the host LF is compeltely ruined and it never makes any sense to expect extracting watermark from such degraded LF. However, our simulations prove that even if such aggressive attack takes place, the existance of the color conversion is advantegous for both man and toys LF. While the removal of the 4D wavelet deteriorates the BER of the man LF, it causes minor improvement to the toys LF.

Concerning the median filter, it is observed that the removal of the color conversion only degrades the robustness of the proposed method. However, the removal of the 4D wavelet causes dramatic rise of the BER against median filtering. Figure 3.40 and Fig. 3.41 show the robustness of the mentioned methods against median filtering. For the man LF, if $gf = 80$, then the BER of the proposed method equals 8.50% while the removal of the color conversion and 4D wavelet result in $BER = 10.06\%, 42.19\%$. As for the toys LF, the proposed method attains $BER = 23.44\%$ whereas the exclusion of the color conversion and 4D wavelet causes BER of 23.53% and 40.33%, respectively.

Figure 3.42 and Fig. 3.43 show the robustness of the aforementioned methods against JPEG 2000. No noticeable difference was found for the man LF. However, the exclusion of the color conversion slows down the rate of the convergence of BER to zero. This is specially more pronounced for more agressive compression ratio of $CR = 30, 40$. With respect to the toys LF, it is observed that the removal of the 4D wavelet slightly improves BER for more aggressive compression ratio of $CR = 30$ and 40. Provided that $gf = 80$, the removal of the 4D wavelet delivers BER of 2.73% which is 3.42% lower than the proposed method. The removal of color conversion considerably slows down the convergence rate of the *BER* to zero for $CR =$ 10, 20. Further more, the impact of excluding color conversion is even more destructive for $CR = 30, 40$ such that the *BER* never converges to zero.

To summarize this section, the performed simulations prove that removing every individual transform reduces the performance of the proposed method against Gaussian noise and the optimal robustness is accomplished only when all the transforms are employed jointly. The exclusion may bearly improve the *BER* of the proposed method against JPEG 2000 but it will cause catastrophic rise of the *BER* against median filtering. On the other hand, removing the color conversion also results in significant increase of *BER* of the proposed method II against JPEG and JPEG 2000 compression. Taking all the arguments of this section into account, it sounds absolutely plausible to conclude that each of the transformations provide a good fit to the robustness of the proposed method against the addressed attacks.

3.6.5 Impact of block size and number of DCT coefficients

As mentioned earlier in section3.4 and 3.5, not all the DCT coefficients are used for watermark insertion. The bigger the size of blocks used for watermark insertion, the higher percentage of watermarked LF will be used to carry watermark. This signifies that if some portion of the watermarked coefficients (for any reason) are altered, there will be a chance to obtain the embedded watermark from the remaining coefficients. On the other hand, the number of the DCT coefficients also has an underlying effect on the watermarked LF. Using more DCT coefficients implies involving more high frequency content (e.g. the edges) which deteriorates the robustness of the proposed method against some attacks.

The impact of *BlockSize* and number of DCT coefficients have been put under close scrutiny. *BlockSize* has been set to 2, 4 while *n_dct* has been set to 1, 3, 6, 10, 13 and 15. Our simulations showed that the modification of *BlockSize* and *n_dct* causes no observable impact on *PSNR*, *BER* and *MSSIM*. Figure 3.44, Fig. 3.45, Fig. 3.46 and Fig. 3.47 indicate that the *BlockSize* and *n_dct* has no visual impact on the watermarked

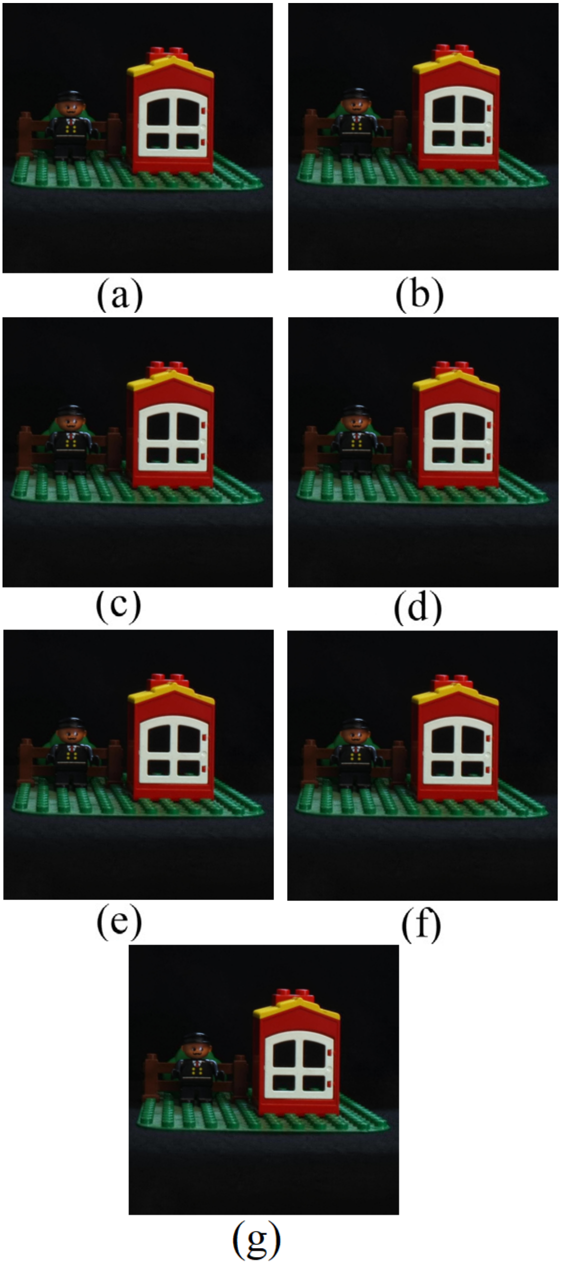

(a) (b) (c) (d) (e) (f) (g)

Figure 3.44: Illustration of the impact of block size and number of DCT coefficients on the visual appearance of the central view of watermarked man LF. (a) The host LF. Watermarked LF with BlockSize $= 4$*: (b)* $n\_dct = 1$ *(c);* $n\_dct = 3$*; (d)* $n\_dct = 6$*; (e)* $n\_dct = 10$*; (f)* $n\_dct = 13$*; (g)* $n\_dct = 15$*.*

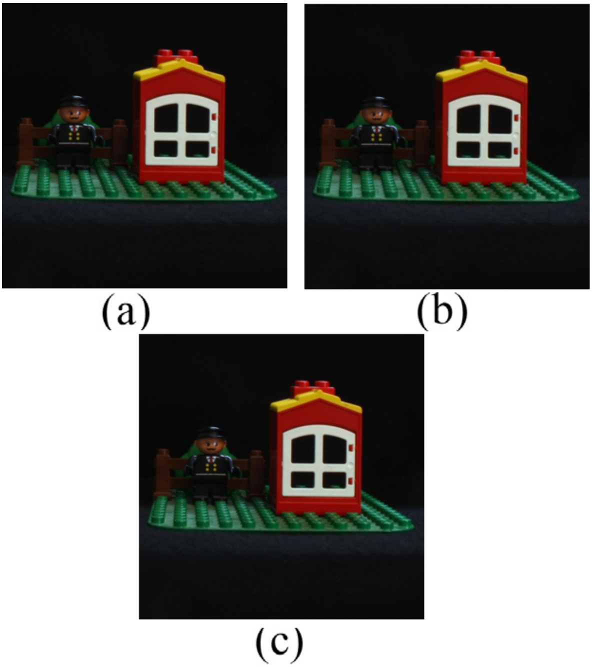


Figure 3.45: Illustration of the impact of block size and number of DCT coefficients on the visual appearance of the central view of watermarked man LF. (a) The host LF. Watermarked LF with BlockSize $= 2$*: (b)* $n\_dct = 1$*; (c)* $n\_dct = 3$.

LF. Figure 3.48 and Fig. 3.49 also show that there is no noticeable impact on the *PSNR*, *BER* and *MSSIM* of the watermarked LF in the practical range. Fig. 3.50 shows that using 2×2 blocks causes no noticeable effect on the robustness of the embedded watermark against Gaussian noise with noise power of 100, 225. Once the noise power rises to 625, some differences are observed and the *BER* rises slightly. The difference becomes even larger for Gaussian noise with noise power of 1225. As expected, using 2×2 blocks will have a huge adverse effect on robustness of the proposed method II. This will be exacerbated if the number of DCT coefficients decrease to three. Anyway, using one DCT coefficient vastly ruins the robustness of the watermarking scheme against other attacks. Letting $BlockSize = 4$ would substantially improve the BER values against Gaussian noise. However, as shown in Fig. 3.50 using more DCT coefficients will undermine the positive impact of bigger block size even though most of the times the results of $BlockSize = 4$ remains superior to $BlockSize = 2$. For example, letting $n\_dct = 10, 13$ introduces a significant rise of BER and for $n\_dct = 15$, the BER will never reach zero. This can be fully justified due to the fact that utilizing more DCT coefficients implies more high frequency content to carry the watermark and makes the whole watermarking process more vulnerable to Gaussian noise.

As per JPEG compression, according to Fig. 3.52 letting $qf = 25\%$, 50% and $BlockSize = 2$, leads to results considerably inferior to $BlockSize = 4$. For $qf = 75\%$, letting $BlockSize = 2$ slightly improves the results. This can be related to minimal distortion of compressed image with $qf = 75\%$. As $BlockSize = 4$ alters more coefficients, a smaller block will cause less distortion than a big one and hence a lower BER will be achieved. In case of $qf = 100\%$, the compressed image has such a good quality that no significant difference was found based on $BlockSize$.

It can be seen from Fig. 3.54, with regard to median filtering, $BlockSize = 2$ yields poor BER figures such that the BER will never fall below 18.84%. Conversely, for $BlockSize = 4$ and $n\_dct = 6$, the BER will attain values less than 8.20%. It can be seen from Fig. 3.56 that as far as JPEG 2000 is concerned, no noticeable difference was found between $BlockSize = 2, 4$.

As can be seen from Fig. 3.46, Fig. 3.47, Fig. 3.49, Fig. 3.51, Fig. 3.53, Fig. 3.55 and Fig. 3.57, all the aforementioned statements also hold for the toys LF except

(i) The robustness of the proposed method remains quite constant against Gaussian noise regardless of $BlockSize$ and n_dct.

(ii) $BlockSize = 4$ yields a lower BER for JPEG compression with $qf = 75\%$.

(iii) $BlockSize = 2$ generally results in a lower BER for JPEG 2000.

Taking all these considerations into account, it seems that letting $BlockSize = 4$ and $n\_dct = 6$ is a balanced choice to adjust the robustness against different attacks.

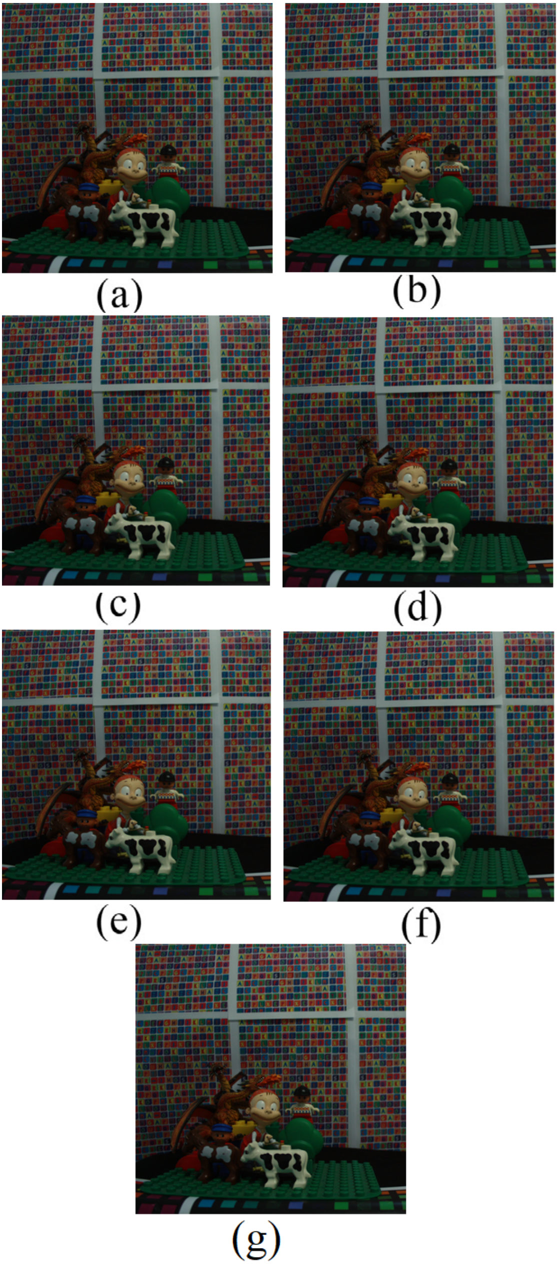


Figure 3.46: Illustration of the impact of block size and number of DCT coefficients on the visual appearance of the central view of watermarked toys LF. (a) The host LF. Watermarked LF with BlockSize $= 4$*: (b)* $n\_dct = 1$ *(c);* $n\_dct = 3$*; (d)* $n\_dct = 6$*; (e)* $n\_dct = 10$*; (f)* $n\_dct = 13$*; (g)* $n\_dct = 15$*.*

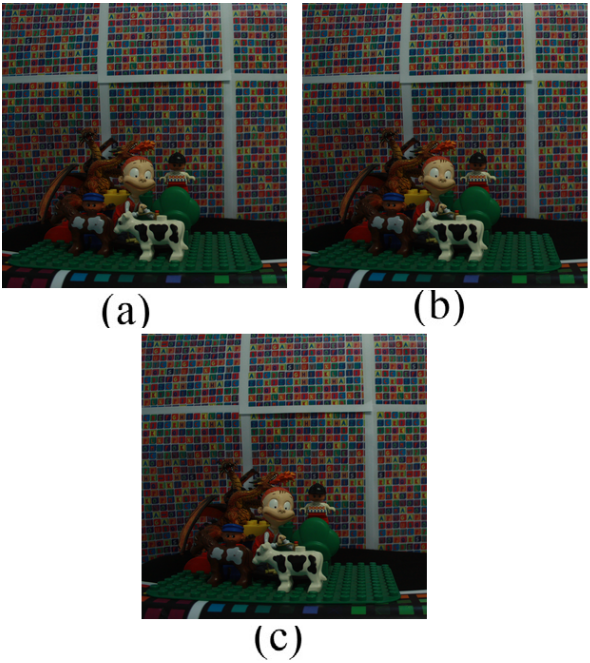


Figure 3.47: Illustration of the impact of block size and number of DCT coefficients on the visual appearance of the central view of watermarked toys LF. (a) The host LF. Watermarked LF with BlockSize $= 2$*: (b)* $n\_dct = 1$*; (c)* $n\_dct = 3$*.*

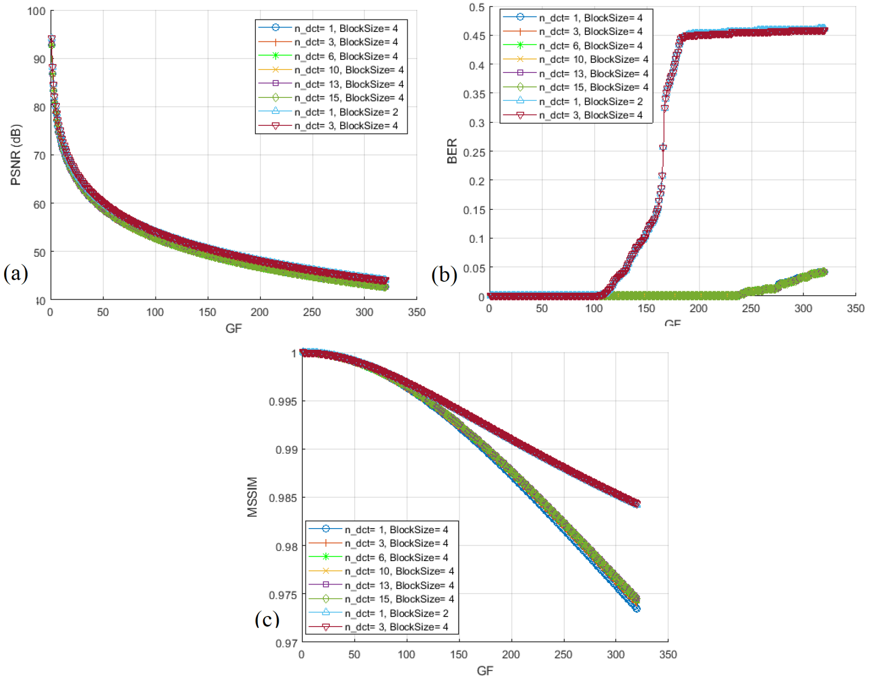
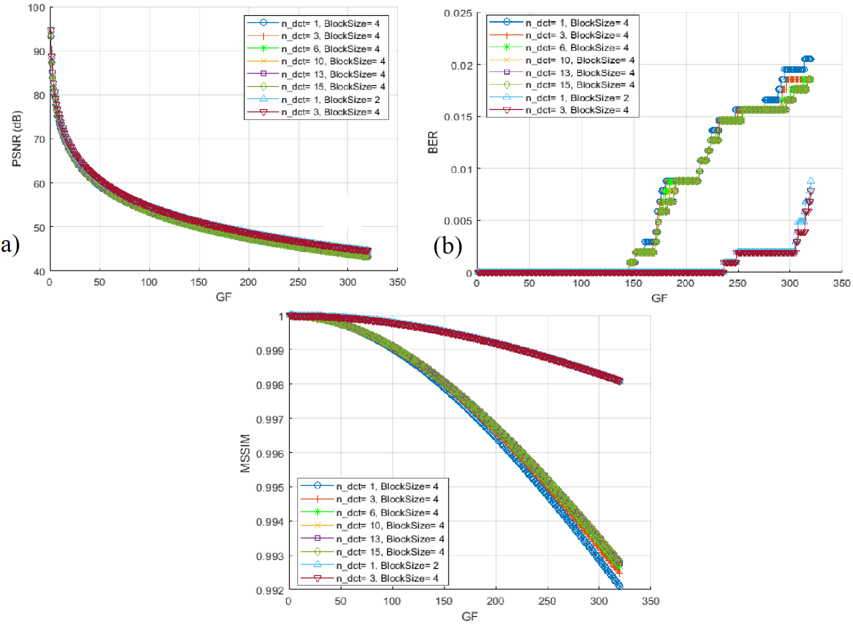
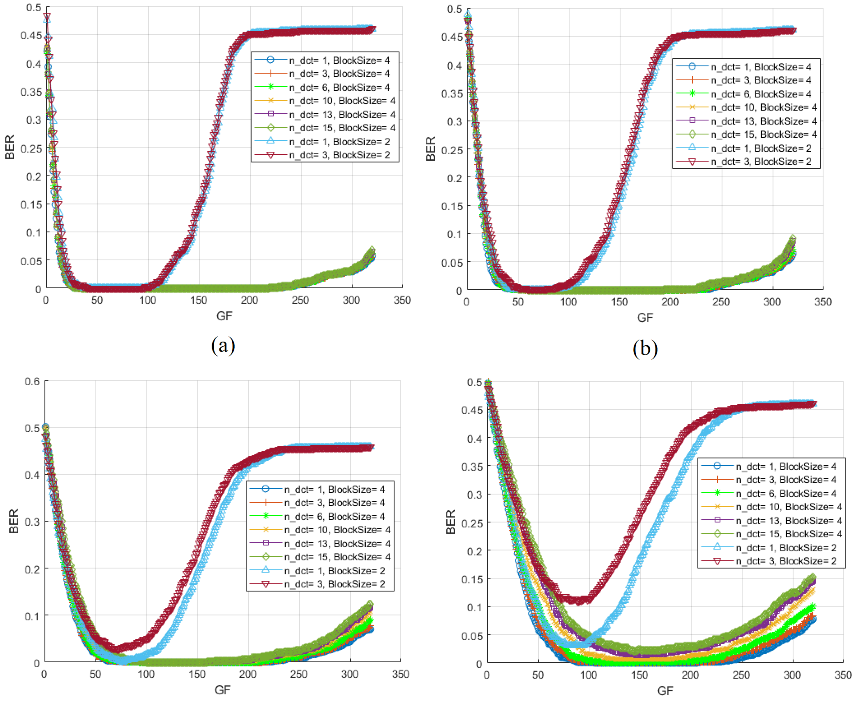


Figure 3.48: The impact of block size and number of DCT coefficients on (a) PSNR, (b) BER, and (c) MSSIM of the watermarked LF (man).



(a) (b) (c)

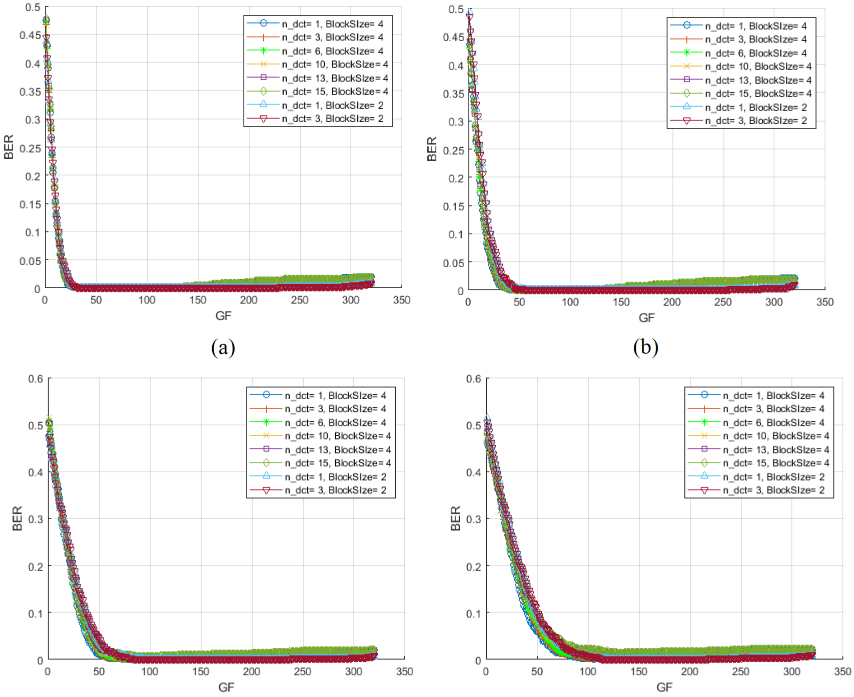
Figure 3.49: The impact of block size and number of DCT coefficients on a) PSNR, b) BER, and c) MSSIM of the watermarked LF (toys).



(a) (b)

(c) (d)

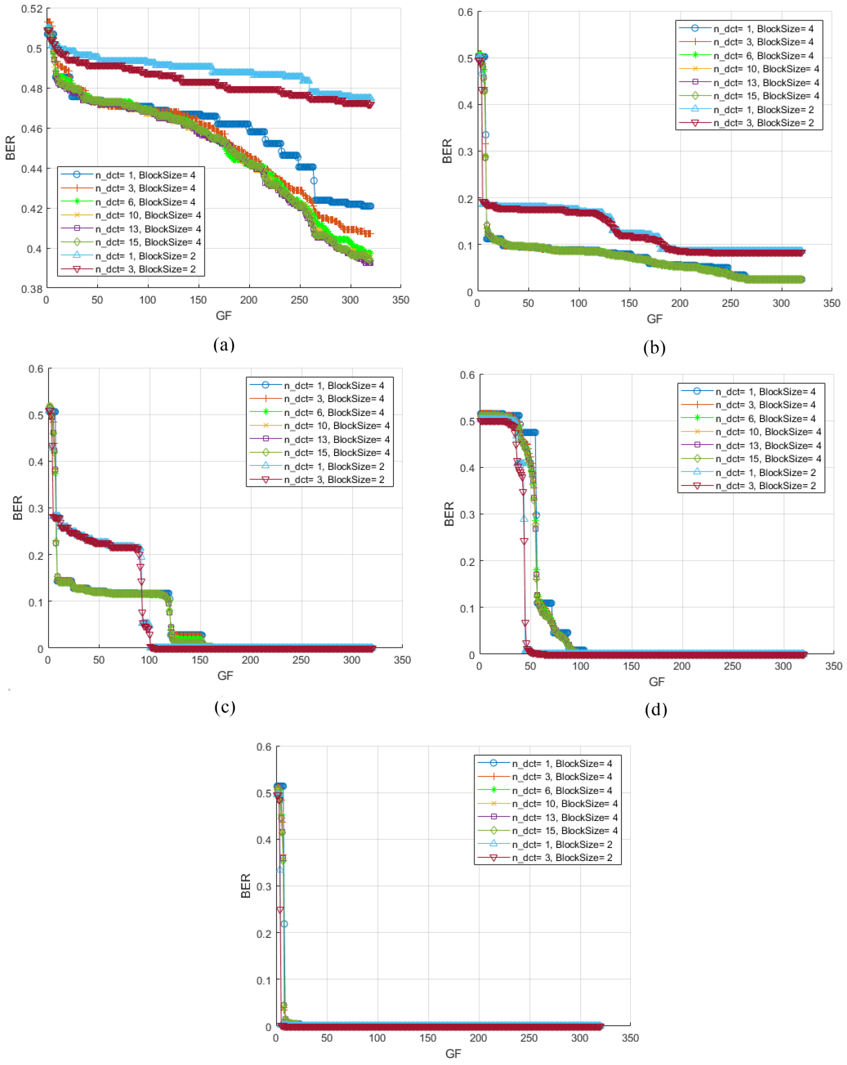
Figure 3.50: The impact of block size and number of DCT coefficients on robustness of the proposed method against Gaussian noise (man): (a) $\sigma_n^2 = 100$; (b) $\sigma_n^2 = 225$; (c) $\sigma_n^2 = 625$; and (d) $\sigma_n^2 = 1225$.



(a) (b)

(c) (d)

Figure 3.51: The impact of block size and number of DCT coefficients on robustness of the proposed method against Gaussian noise (toys): (a) $\sigma_n^2 = 100$; (b) $\sigma_n^2 = 225$; (c) $\sigma_n^2 = 625$; and (d) $\sigma_n^2 = 1225$.



(a)

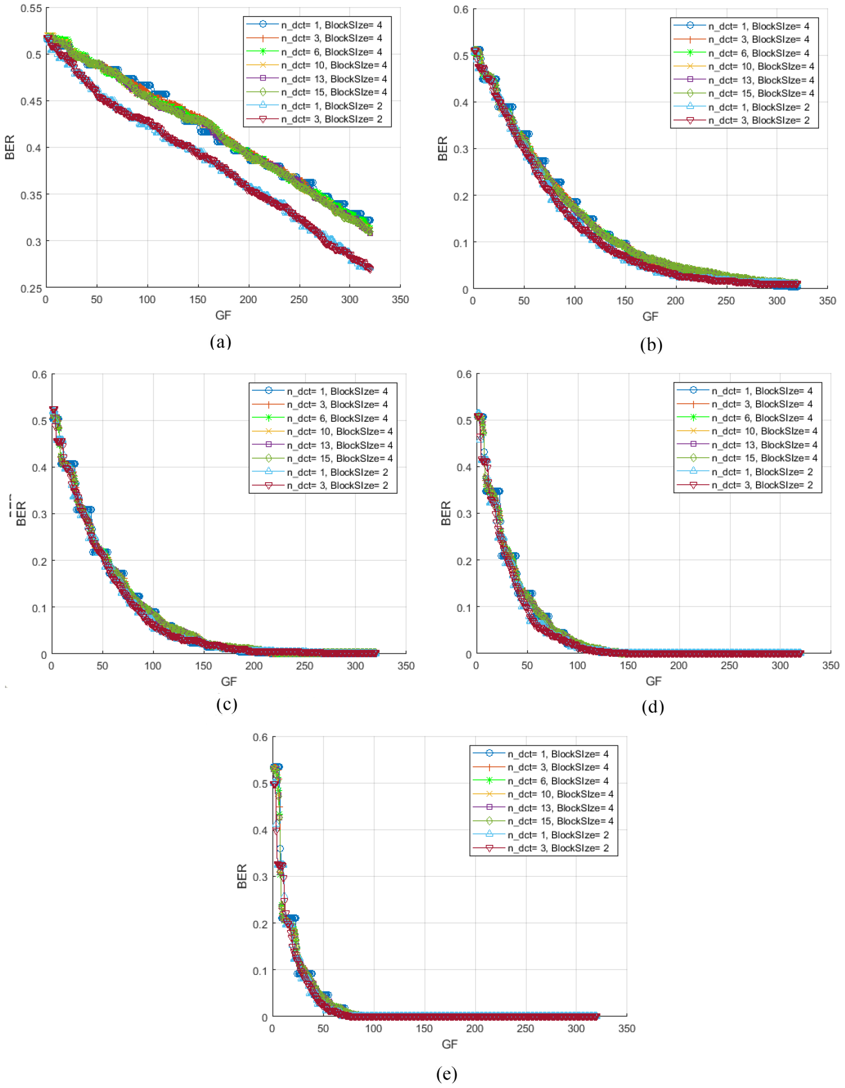
(b)

(c)

(d)

(e)

Figure 3.52: The impact of block size and number of DCT coefficients on robustness of the proposed method against JPEG compression (man): (a) qf = 5%; (b) qf = 25%; (c) qf = 50%; (d) qf = 75%; (e) qf = 100%.



(a) (b) (c) (d) (e)

Figure 3.53: The impact of block size and number of DCT coefficients on robustness of the proposed method against JPEG compression (toys): (a) qf = 5%; (b) qf = 25%; (c) qf = 50%; (d) qf = 75%; (e) qf = 100%.

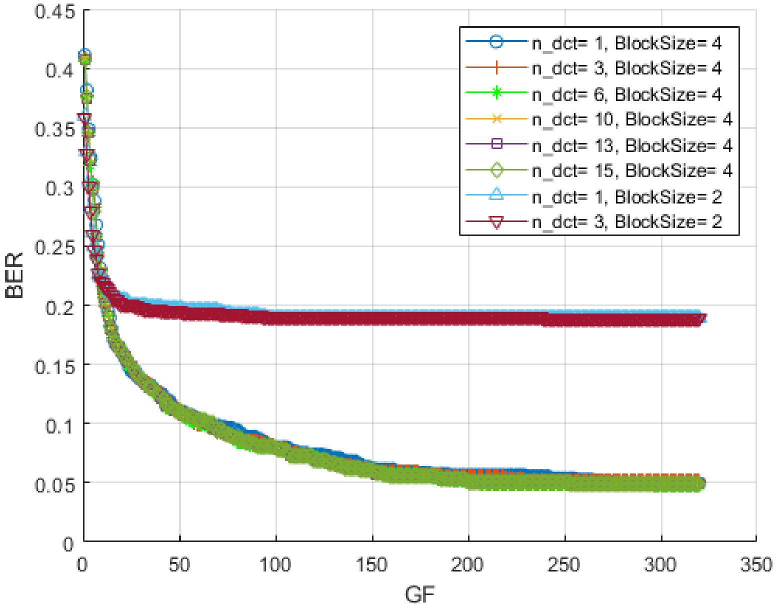


Figure 3.54: The impact of block size and number of DCT coefficients on robustness of the proposed method against median filtering (man).

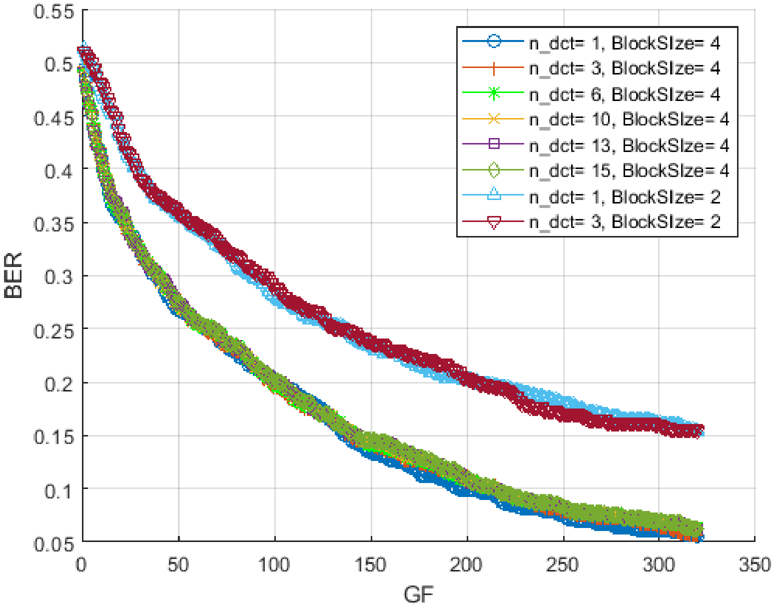
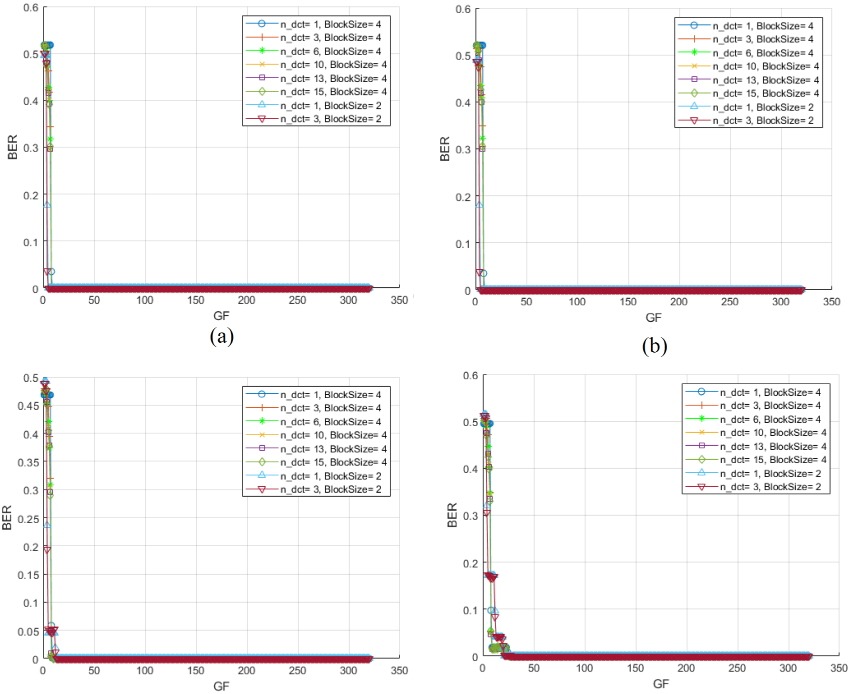


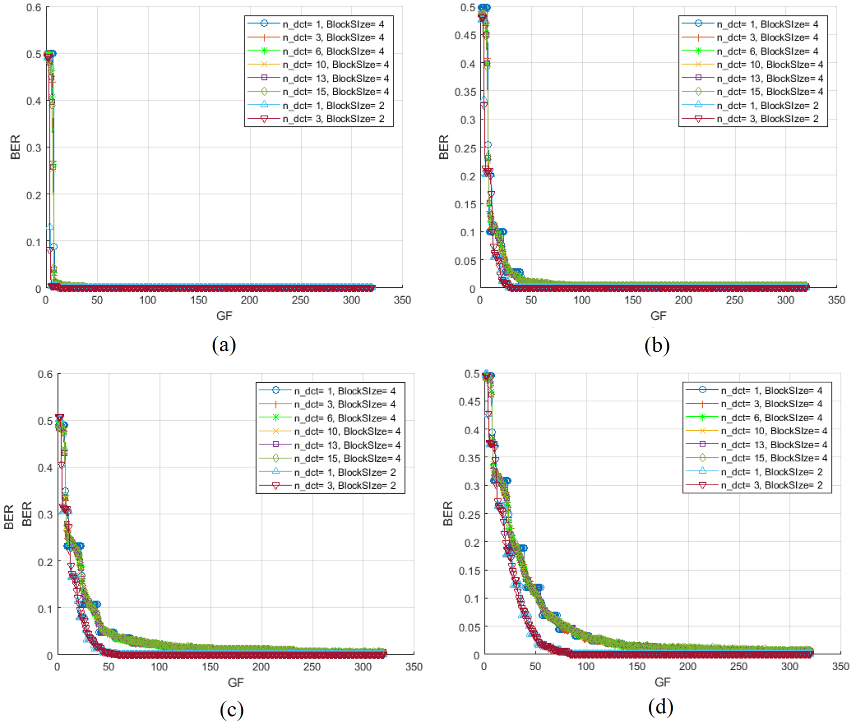
Figure 3.55: The impact of block size and number of DCT coefficients on robustness of the proposed method against median filtering (toys).



(a) (b)

(c) (d)

Figure 3.56: The impact of block size and number of DCT coefficients on robustness of the proposed method against JPEG 2000 compression (man): (a) CR= 10; (b) CR= 20; (c) CR= 30; (d) CR= 40.



(a) (b)

(c) (d)

Figure 3.57: The impact of block size and number of DCT coefficients on robustness of the proposed method against JPEG 2000 compression (toys): (a) CR= 10; (b) CR= 20; (c) CR= 30; (d) CR= 40.

Chapter 4

Conclusions

Over the past few years, the LF has been increasingly used for 3D reconstruction. On the other hand, the watermarking methods have always been used for ownership protection, authentication, fingerprinting, tamper detection and error detection/correction. Despite the countless number of the watermarking methods proposed for the conventional images, there are very few ones addressing the LF. Given the tremendous amount of the spatial and angular information of the LF, any watermarking method should protect both spatial and angular information and minimize the distortion of 3D reconstruction. Throughout this dissertation, two watermarking methods have been proposed. The method I employs DCT and SVD for LF watermarking. First, the basic premise of the method I and mathematical flow have been elaborated. The reason of discarding most of the DCT coefficients has been discussed in detail. Then assessment metrics have been properly generalized to LF images with higher dimensionality. The transparency of the watermarked LF has been verified subjectively and objectively both confirming the highest fidelity of the proposed method. The robustness of the method I has been measured against some common image processing attacks as Gaussian noise, JPEG compression, and median filtering. The simulations show that using too few or too many DCT coefficients deteriorates the robustness of the embedded watermark. Furthermore, the simulations demonstrated that the removal of the DCT from the watermarking system causes dramatic increment of the *BER*. Unlike most of the watermarking methods, if the watermarked LF is exposed to no attack, the embedded watermark can be extracted error-free.

Apart from the aforementioned LF watermarking method, another one has been proposed. The method II gains the huge redundancy of the LF along all dimensions. First, the importance of the 4D wavelet and its role in preserving spatial and angular information of the LF is highlighted. Both spatial and angular information of the LF vastly contribute to the the 3D reconstruction of the scene. The absolute importance of the 4D wavelet for LF watermarking has been explained in detail and the mathematical derivation of the 4D wavelet is discussed. Apart from the mathematical aspect of the wavelet, the exploitation of filter banks to realize 4D wavelet has been described. As powerful as 4D wavelet is in spatial and angular decorrelation, the enormous correlation of the color channels has to be reduced by color conversion. Similar to method I , SVD has also been used in method II to acquire a robust watermarking feature. The advantages of 4D wavelet in LF watermarking and mathematical representation of the method II have been described. Afterwards, the transparency of the watermarked LF has been measured subjectively and objectively. Even for the lowest values of gf, the embedded watermark is extracted error-free. Provided that the appropriate value of the watermark strength is chosen, the achieved *PSNR* is so high that the HVS cannot detect any difference. The robustness of the method II has been measured against Gaussian noise, JPEG compressian, median filtering and JPEG 2000. To put all the experimental results of the robustness analysis in a nutshell, the method II outperforms method I and SVD. Specifically in case of more extreme attacks, the superiority of the method II over method I and SVD is quite overwhelming. As the method II uses multiple transforms, the contribution of the individual transforms to the robustness of the method II has been investigated. It was observed that the optimal performance is accomplished only if all the transforms are used jointly. As mentioned earlier, the LF has higher dimensionality than the conventional 2D images. Each of the individual transforms serve to decorrelate the LF along one specific dimension which is not possible to achieve by other transforms. The impact of the block size and the number of DCT coefficients on the robustness of the method II have been examined. It is true that the larger the block size, the lower the BER will be. Nevertheless, using huge blocks will decrease the capacity of the watermarking method. Hence, 4×4 blocks are preferred for the method II. It was observed that using smaller blocks of 2×2 adversely affects the robustness of the proposed method. Completely in line with the experimental results of the method I, using too few or too many DCT coefficients lowers the robustness of the method II against the studied attacks.

Although we have used the 4D wavelet for LF watermarking, it may also be exploited in other areas of the LF image processing as denoising, compression, pattern recognition, steganography, etc. Besides the LF, the adoption of the proposed methods to video watermarking is straight forward. Regarding the immense size of video pixels, the proposed methods can be used for covert communication. The shearlet transform has already delivered promising results in LF compression [99]. Additional research on using shearlet for LF watermarking can be of interest for achieving higher robustness. Further work will concentrate on emloying multi-level decomposition or other variants of the wavelet transform for LF watermarking. This work was very much focused on maximizing the robustness of the proposed methods. However, we hope that our research will also be used in fragile watermarking e.g. for tamper detection. This goal can be attained by utilizing high-frequency subbands as $HHHH$. Taking the enormous amount of data required to represent the typical LF images, an appealing research topic is to speed up the proposed methods by GPU or cloud computing techniques. The proposed methods employ DCT and wavelet for watermarking which are perfect candidates for parallelization. Furthermore, as the location of the watermark bits is specified by the secret key, all the watermark bits can be embedded/extracted simultaneously. It makes perfect sense to assume that employing multiple GPUs may result in real time or near real time performance. We have already parallelized part of our future research on LF watermarking and the computation time has noticeably decreased. The simulations with other filter banks different from Haar wavelets is in progress and will be published in the near future. To further this research, it is also possible to investigate the proper generalization of other common transforms as contourlet, curvelet, ridgelet or other appropriate ones.

Bibliography

[1] F. A. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding-a survey," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999.

[2] J. Poort, J. Quintais, M. A. van der Ende, A. Yagafarova, and M. Hageraats, "Global online piracy study," tech. rep., Institute of Information Law, University of Amsterdam, 2018.

[3] Wikipedia, "Steganography." `https://en.wikipedia.org/wiki/Steganography`, accessed 2020-12-22.

[4] G. Kipper, *Investigator's guide to steganography*. CRC press, 2003.

[5] E. Zielińska, W. Mazurczyk, and K. Szczypiorski, "Trends in steganography," *Communications of the ACM*, vol. 57, no. 3, pp. 86–95, 2014.

[6] S. K. Bandyopadhyay, D. Bhattacharyya, D. Ganguly, S. Mukherjee, and P. Das, "A tutorial review on steganography," in *International conference on contemporary computing*, vol. 101, pp. 105–114, 2008.

[7] R. Mishra and P. Bhanodiya, "A review on steganography and cryptography," in *2015 International Conference on Advances in Computer Engineering and Applications*, pp. 119–122, IEEE, 2015.

[8] R. Gupta, S. Gupta, and A. Singhal, “Importance and techniques of information hiding: A review,” *arXiv preprint arXiv:1404.3063*, 2014.

[9] M. S. Henriques and N. K. Vernekar, “Using symmetric and asymmetric cryptography to secure communication between devices in iot,” in *2017 International Conference on IoT and Application (ICIOT)*, pp. 1–4, IEEE, 2017.

[10] D. P. Joseph, M. Krishna, and K. Arun, “Cognitive analytics and comparison of symmetric and asymmetric cryptography algorithms,” *International Journal of Advanced Research in Computer Science*, vol. 6, no. 3, 2015.

[11] N. Ferguson and B. Schneier, *Practical cryptography*, vol. 141. Wiley, New York, 2003.

[12] M. Barakat, C. Eder, and T. Hanke, *Lecture notes: An introduction to cryptography*. Rheinish-Westphalian Technical (RWTH) University, September 2018. `https://www.mathematik.uni-kl.de/~ederc/download/Cryptography.pdf`.

[13] A. S. Panah, R. Van Schyndel, T. Sellis, and E. Bertino, “On the properties of non-media digital watermarking: a review of state of the art techniques,” *IEEE Access*, vol. 4, pp. 2670–2704, 2016.

[14] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital watermarking and steganography*. Morgan Kaufmann, 2007.

[15] W. Szepanski, “Carnahan conference on crime countermeasures,” in *Proceedings*, p. 101, University of Kentucky, Office of Research and Engineering Services, 1979.

[16] J. P. Quintais and J. Poort, “The decline of online piracy: How markets-not enforcement-drive down copyright infringement,” *American University International Law Review*, vol. 34, no. 4, p. 807, 2019.

[17] J. Band, "The digital millennium copyright act," 1998. On-line document (`https://www.congress.gov/bill/105th-congress/house-bill/2281`, accessed 2020-12-22).

[18] P. W. Wong and N. Memon, "Secret and public key image watermarking schemes for image authentication and ownership verification," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1593–1601, 2001.

[19] A. K. Singh, "Robust and distortion control dual watermarking in LWT domain using DCT and error correction code for color medical image," *Multimedia Tools and Applications*, vol. 78, no. 21, pp. 30523–30533, 2019.

[20] S. Gull, N. A. Loan, S. A. Parah, J. A. Sheikh, and G. M. Bhat, "An efficient watermarking technique for tamper detection and localization of medical images," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 5, pp. 1799–1808, 2020.

[21] K. J. Giri and R. Bashir, "Digital watermarking: a potential solution for multimedia authentication," in *Intelligent Techniques in Signal Processing for Multimedia Security*, pp. 93–112, Springer, 2017.

[22] A. Ansari, S. Hong, G. Saavedra, B. Javidi, and M. Martinez-Corral, "Ownership protection of plenoptic images by robust and reversible watermarking," *Optics and Lasers in Engineering*, vol. 107, pp. 325–334, 2018.

[23] R. Liu and T. Tan, "An SVD-based watermarking scheme for protecting rightful ownership," *IEEE Transactions on Multimedia*, vol. 4, no. 1, pp. 121–128, 2002.

[24] A. Ansari, A. Dorado, G. Saavedra, and M. Martinez-Corral, "Plenoptic image watermarking to preserve copyright," *Proceedings of the SPIE*, vol. 10219, p. 102190A, 2017.

[25] W.-C. Chen and M.-S. Wang, "A fuzzy c-means clustering-based fragile watermarking scheme for image authentication," *Expert Systems with Applications*, vol. 36, no. 2, pp. 1300–1307, 2009.

[26] C.-J. Cheng, W.-J. Hwang, H.-Y. Zeng, and Y.-C. Lin, "A fragile watermarking algorithm for hologram authentication," *IEEE Journal of Display Technology*, vol. 10, no. 4, pp. 263–271, 2014.

[27] K. Sreenivas and V. K. Prasad, "Fragile watermarking schemes for image authentication: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 7, pp. 1193–1218, 2018.

[28] I. A. Ansari, M. Pant, and C. W. Ahn, "SVD based fragile watermarking scheme for tamper localization and self-recovery," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 6, pp. 1225–1239, 2016.

[29] K. Sreenivas and V. K. Prasad, "Fragile watermarking schemes for image authentication: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 7, pp. 1193–1218, 2018.

[30] J.-P. Boyer, P. Duhamel, and J. Blanc-Talon, "Scalar DC–QIM for semifragile authentication," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 4, pp. 776–782, 2008.

[31] H. M. Al-Otum, "Semi-fragile watermarking for grayscale image authentication and tamper detection based on an adjusted expanded-bit multiscale quantization-based technique," *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 1064–1081, 2014.

[32] X. Qi and X. Xin, "A singular-value-based semi-fragile watermarking scheme for image content authentication with tamper localization," *Journal of Visual Communication and Image Representation*, vol. 30, pp. 312–327, 2015.

[33] A. Ansari, G. Saavedra, and M. Martinez-Corral, “Robust light field watermarking by 4D wavelet transform,” *IEEE Access*, vol. 8, pp. 203117–203133, 2020.

[34] N. Nikolaidis and I. Pitas, “Robust image watermarking in the spatial domain,” *Signal Processing*, vol. 66, no. 3, pp. 385–403, 1998.

[35] Q. Su and B. Chen, “Robust color image watermarking technique in the spatial domain,” *Soft Computing*, vol. 22, no. 1, pp. 91–106, 2018.

[36] Q. Su, Z. Yuan, and D. Liu, “An approximate Schur decomposition-based spatial domain color image watermarking method,” *IEEE Access*, vol. 7, pp. 4358–4370, 2018.

[37] Q. Su, D. Liu, Z. Yuan, G. Wang, X. Zhang, B. Chen, and T. Yao, “New rapid and robust color image watermarking technique in spatial domain,” *IEEE Access*, vol. 7, pp. 30398–30409, 2019.

[38] Q. Cheng and T. S. Huang, “Robust optimum detection of transform domain multiplicative watermarks,” *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 906–924, 2003.

[39] K. Fares, K. Amine, and E. Salah, “A robust blind color image watermarking based on Fourier transform domain,” *Optik*, vol. 208, p. 164562, 2020.

[40] Y. Guo, B.-Z. Li, and N. Goel, “Optimised blind image watermarking method based on firefly algorithm in DWT-QR transform domain,” *IET Image processing*, vol. 11, no. 6, pp. 406–415, 2017.

[41] H. V. Singh and A. Rai, “Medical image watermarking in transform domain,” in *Smart Innovations in Communication and Computational Sciences*, pp. 485–493, Springer, 2019.

[42] W. C. Chu, “DCT-based image watermarking using subsampling,” *IEEE Transactions on Multimedia*, vol. 5, no. 1, pp. 34–38, 2003.

[43] L. Dong, Q. Yan, Y. Lv, and S. Deng, “Full band watermarking in DCT domain with Weibull model,” *Multimedia Tools and Applications*, vol. 76, no. 2, pp. 1983–2000, 2017.

[44] S. Liu, Z. Pan, and H. Song, “Digital image watermarking method based on DCT and fractal encoding,” *IET Image Processing*, vol. 11, no. 10, pp. 815–821, 2017.

[45] H.-J. Ko, C.-T. Huang, G. Horng, and W. Shiuh-Jeng, “Robust and blind image watermarking in DCT domain using inter-block coefficient correlation,” *Information Sciences*, vol. 517, pp. 128–147, 2020.

[46] M. Barni, F. Bartolini, and A. Piva, “Improved wavelet-based watermarking through pixel-wise masking,” *IEEE Transactions on Image Processing*, vol. 10, no. 5, pp. 783–791, 2001.

[47] Z. Dawei, C. Guanrong, and L. Wenbo, “A chaos-based robust wavelet-domain watermarking algorithm,” *Chaos, Solitons & Fractals*, vol. 22, no. 1, pp. 47–54, 2004.

[48] V. S. Jabade and S. R. Gengaje, “Literature review of wavelet based digital image watermarking techniques,” *International Journal of Computer Applications*, vol. 31, no. 7, 2011.

[49] T.-S. Nguyen, C.-C. Chang, and X.-Q. Yang, “A reversible image authentication scheme based on fragile watermarking in discrete wavelet transform domain,” *AEU-International Journal of Electronics and Communications*, vol. 70, no. 8, pp. 1055–1061, 2016.

[50] R. Mehta, N. Rajpal, and V. P. Vishwakarma, “Robust image watermarking scheme in lifting wavelet domain using GA-LSVR hybridization,” *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 1, pp. 145–161, 2018.

[51] M. Rabizadeh, M. Amirmazlaghani, and M. Ahmadian-Attari, “A new detector for contourlet domain multiplicative image watermark-

ing using bessel K form distribution,” *Journal of Visual Communication and Image Representation*, vol. 40, pp. 324–334, 2016.

[52] Q. Su, G. Wang, G. Lv, X. Zhang, G. Deng, and B. Chen, “A novel blind color image watermarking based on contourlet transform and Hessenberg decomposition,” *Multimedia Tools and Applications*, vol. 76, no. 6, pp. 8781–8801, 2017.

[53] H. Sadreazami and M. Amini, “A robust image watermarking scheme using local statistical distribution in the contourlet domain,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 1, pp. 151–155, 2018.

[54] S. Etemad and M. Amirmazlaghani, “A new multiplicative watermark detector in the contourlet domain using t location-scale distribution,” *Pattern Recognition*, vol. 77, pp. 99–112, 2018.

[55] R. S. R. Channapragada and M. V. Prasad, “Watermarking techniques in curvelet domain,” in *Computational Intelligence in Data Mining-Volume 1*, pp. 199–211, Springer, 2015.

[56] W.-H. Kim, S.-H. Nam, J.-H. Kang, and H.-K. Lee, “Robust watermarking in curvelet domain for preserving cleanness of high-quality images,” *Multimedia Tools and Applications*, vol. 78, no. 12, pp. 16887–16906, 2019.

[57] C. R. S. Rao and M. V. Prasad, *Digital watermarking techniques in curvelet and ridgelet domain.* Springer, 2016.

[58] H. Sadreazami and M. Amini, “A robust spread spectrum based image watermarking in ridgelet domain,” *AEU-International Journal of Electronics and Communications*, vol. 66, no. 5, pp. 364–371, 2012.

[59] A. Abadpour and S. Kasaei, “Color PCA eigenimages and their application to compression and watermarking,” *Image and Vision Computing*, vol. 26, no. 7, pp. 878–890, 2008.

[60] F.-N. Lang, J.-L. Zhou, S. Cang, H. Yu, and Z. Shang, "A self-adaptive image normalization and quaternion PCA based color image watermarking algorithm," *Expert Systems with Applications*, vol. 39, no. 15, pp. 12046–12060, 2012.

[61] V. S. Verma, A. Bhardwaj, and R. K. Jha, "A new scheme for watermark extraction using combined noise-induced resonance and support vector machine with PCA based feature reduction," *Multimedia Tools and Applications*, pp. 1–22, 2019.

[62] E. Ganic and A. M. Eskicioglu, "Robust DWT-SVD domain image watermarking: embedding data in all frequencies," in *Proceedings of the 2004 Workshop on Multimedia and Security*, pp. 166–174, 2004.

[63] C.-C. Chang, P. Tsai, and C.-C. Lin, "SVD-based digital image watermarking scheme," *Pattern Recognition Letters*, vol. 26, no. 10, pp. 1577–1586, 2005.

[64] K.-L. Chung, W.-N. Yang, Y.-H. Huang, S.-T. Wu, and Y.-C. Hsu, "On SVD-based watermarking algorithm," *Applied Mathematics and Computation*, vol. 188, no. 1, pp. 54–57, 2007.

[65] C.-C. Lai and C.-C. Tsai, "Digital image watermarking using discrete wavelet transform and singular value decomposition," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 11, pp. 3060–3063, 2010.

[66] J. Wang and S. Lian, "On the hybrid multi-watermarking," *Signal Processing*, vol. 92, no. 4, pp. 893–904, 2012.

[67] C. Song, S. Sudirman, and M. Merabti, "A robust region-adaptive dual image watermarking technique," *Journal of Visual Communication and Image Representation*, vol. 23, no. 3, pp. 549–568, 2012.

[68] M. Cedillo-Hernández, F. García-Ugalde, M. Nakano-Miyatake, and H. M. Pérez-Meana, "Robust hybrid color image watermarking method based on DFT domain and 2D histogram modification," *Signal, Image and Video Processing*, vol. 8, no. 1, pp. 49–63, 2014.

[69] N. M. Makbol, B. E. Khoo, and T. H. Rassem, “Security analyses of false positive problem for the SVD-based hybrid digital image watermarking techniques in the wavelet transform domain,” *Multimedia Tools and Applications*, vol. 77, no. 20, pp. 26845–26879, 2018.

[70] M. Hamidi, M. El Haziti, H. Cherifi, and M. El Hassouni, “Hybrid blind robust image watermarking technique based on DFT-DCT and Arnold transform,” *Multimedia Tools and Applications*, vol. 77, no. 20, pp. 27181–27214, 2018.

[71] J. J. Eggers, J. K. Su, and B. Girod, “Robustness of a blind image watermarking scheme,” in *Proceedings of International Conference on Image Processing (Cat. No. 00CH37101)*, vol. 3, pp. 17–20, IEEE, 2000.

[72] W.-H. Lin, Y.-R. Wang, S.-J. Horng, T.-W. Kao, and Y. Pan, “A blind watermarking method using maximum wavelet coefficient quantization,” *Expert Systems with Applications*, vol. 36, no. 9, pp. 11509–11516, 2009.

[73] M. A. Akhaee, S. M. E. Sahraeian, and C. Jin, “Blind image watermarking using a sample projection approach,” *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 883–893, 2011.

[74] N. M. Makbol and B. E. Khoo, “Robust blind image watermarking scheme based on redundant discrete wavelet transform and singular value decomposition,” *AEU-International Journal of Electronics and Communications*, vol. 67, no. 2, pp. 102–112, 2013.

[75] A. Houmansadr, N. Kiyavash, and N. Borisov, “Non-blind watermarking of network flows,” *IEEE/ACM Transactions on Networking*, vol. 22, no. 4, pp. 1232–1244, 2013.

[76] O. Jane, E. Elbaşi, *et al.*, “Hybrid non-blind watermarking based on DWT and SVD,” *Journal of Applied Research and Technology*, vol. 12, no. 4, pp. 750–761, 2014.

[77] P. Vaidya and C. M. PVSSR, “A robust semi-blind watermarking for color images based on multiple decompositions,” *Multimedia Tools and Applications*, vol. 76, no. 24, pp. 25623–25656, 2017.

[78] S. Lee, C. D. Yoo, and T. Kalker, “Reversible image watermarking based on integer-to-integer wavelet transform,” *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 321–330, 2007.

[79] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, “Reversible watermarking algorithm using sorting and prediction,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 989–999, 2009.

[80] L. Luo, Z. Chen, M. Chen, X. Zeng, and Z. Xiong, “Reversible image watermarking using interpolation technique,” *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 187–193, 2009.

[81] S. Lee, C. D. Yoo, and T. Kalker, “Reversible image watermarking based on integer-to-integer wavelet transform,” *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 321–330, 2007.

[82] A. Ustubioglu and G. Ulutas, “A new medical image watermarking technique with finer tamper localization,” *Journal of Digital Imaging*, vol. 30, no. 6, pp. 665–680, 2017.

[83] M. Kutter and S. Winkler, “A vision-based masking model for spread-spectrum image watermarking,” *IEEE Transactions on Image Processing*, vol. 11, no. 1, pp. 16–25, 2002.

[84] P. S. Huang, C.-S. Chiang, C.-P. Chang, and T.-M. Tu, “Robust spatial watermarking technique for colour images via direct saturation adjustment,” *IEE Proceedings-Vision, Image and Signal Processing*, vol. 152, no. 5, pp. 561–574, 2005.

[85] A. K. Parthasarathy and S. Kak, “An improved method of content based image watermarking,” *IEEE Transactions on Broadcasting*, vol. 53, no. 2, pp. 468–479, 2007.

[86] X.-y. Wang, C.-p. Wang, H.-y. Yang, and P.-p. Niu, “A robust blind color image watermarking in quaternion Fourier transform domain,” *Journal of Systems and Software*, vol. 86, no. 2, pp. 255–277, 2013.

[87] I. Ihrke, J. Restrepo, and L. Mignard-Debise, “Principles of light field imaging: Briefly revisiting 25 years of research,” *IEEE Signal Processing Magazine*, vol. 33, no. 5, pp. 59–69, 2016.

[88] M. Faraday, “LIV. Thoughts on ray-vibrations,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 28, no. 188, pp. 345–350, 1846. `https://doi.org/10.1080/14786444608645431`.

[89] Wikipedia, “Light field.” `https://en.wikipedia.org/wiki/Light_field`, accessed 2021-08-08.

[90] A. Gershun, “The light field,” *Journal of Mathematics and Physics*, vol. 18, no. 1-4, pp. 51–151, 1939.

[91] Statista, “Number of digital 3D cinema screens worldwide from 2006 to 2019.” `https://www.statista.com/statistics/271863/number-of-3d-cinema-screens-worldwide/`, accessed 2020-12-22.

[92] G. Wu, B. Masia, A. Jarabo, Y. Zhang, L. Wang, Q. Dai, T. Chai, and Y. Liu, “Light field image processing: An overview,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 926–954, 2017.

[93] E. H. Adelson and J. R. Bergen, *”The plenoptic function and the elements of early vision”*, vol. 2. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of Technology, 1991.

[94] M. Levoy and P. Hanrahan, “Light field rendering,” in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 31–42, 1996.

[95] Y. Chen, M. Alain, and A. Smolic, “A study of efficient light field subsampling and reconstruction strategies,” *arXiv preprint arXiv:2008.04694*, 2020.

[96] d. V. Leonardo and M. Walker, “Leonardo on painting: An anthology of writings by Leonardo da Vinci with a selection of documents relating to his career as an artist,” 1989.

[97] Y. Lu, S. You, W. Zhang, B. Yang, R. Peng, and S. Zhuang, “Watermarking scheme for microlens-array-based four-dimensional light field imaging,” *Applied Optics*, vol. 55, no. 13, pp. 3397–3404, 2016.

[98] M. A. Magnor, A. Endmann, and B. Girod, “Progressive compression and rendering of light fields.,” *Proceedings of Vision, Modeling, and Visualization*, pp. 199–204, 2000.

[99] S. Vagharshakyan, R. Bregovic, and A. Gotchev, “Light field reconstruction using shearlet transform,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 1, pp. 133–147, 2017.

[100] X. Huang, P. An, F. Cao, D. Liu, and Q. Wu, “Light-field compression using a pair of steps and depth estimation,” *Optics express*, vol. 27, no. 3, pp. 3557–3573, 2019.

[101] K. Li, J. Zhang, R. Sun, X. Zhang, and J. Gao, “EPI-based oriented relation networks for light field depth estimation,” *arXiv preprint arXiv:2007.04538*, 2020.

[102] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, “High performance imaging using large camera arrays,” in *ACM SIGGRAPH 2005 Papers*, pp. 765–776, 2005.

[103] J. C. Yang, M. Everett, C. Buehler, and L. McMillan, “A real-time distributed light field camera.,” *Rendering Techniques*, vol. 2002, pp. 77–86, 2002.

[104] B. S. Wilburn, M. Smulski, H.-H. K. Lee, and M. A. Horowitz, “Light field video camera,” in *Media Processors 2002*, vol. 4674, pp. 29–36, International Society for Optics and Photonics, 2001.

[105] B. Wilburn, N. Joshi, V. Vaish, M. Levoy, and M. Horowitz, “High-speed videography using a dense camera array,” in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 2, pp. II–II, IEEE, 2004.

[106] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, “High performance imaging using large camera arrays,” in *ACM SIGGRAPH 2005 Papers*, pp. 765–776, 2005.

[107] C. Zhang and T. Chen, “A self-reconfigurable camera array,” in *ACM SIGGRAPH 2004 Sketches*, p. 151, 2004.

[108] S.-C. Chan, K.-T. Ng, Z.-F. Gan, K.-L. Chan, and H.-Y. Shum, “The plenoptic video,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 12, pp. 1650–1659, 2005.

[109] V. Vaish, M. Levoy, R. Szeliski, C. L. Zitnick, and S. B. Kang, “Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, vol. 2, pp. 2331–2338, IEEE, 2006.

[110] Y. Liu, Q. Dai, and W. Xu, “A real time interactive dynamic light field transmission system,” in *2006 IEEE International Conference on Multimedia and Expo*, pp. 2173–2176, IEEE, 2006.

[111] E. Miandji, *Sparse representation of visual data for compression and compressed sensing*, vol. 1963. Linköping University Electronic Press, 2018.

[112] U. of Heidelberg, "Light field acquistion system." `https://hci.iwr.uni-heidelberg.de/hci/softwares/light_field_analysis/acquisition`, accessed 2020-12-22.

[113] Z. Yu, X. Guo, H. Lin, A. Lumsdaine, and J. Yu, "Line assisted light field triangulation and stereo matching," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2792–2799, 2013.

[114] D. G. Dansereau, G. Schuster, J. Ford, and G. Wetzstein, "A wide-field-of-view monocentric light field camera," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5048–5057, 2017.

[115] I. Ihrke, T. Stich, H. Gottschlich, M. Magnor, and H.-P. Seidel, "Fast incident light field acquisition and rendering," *Journal of the World Society for Computer Graphics*, vol. 16, pp. 25–32, 2008.

[116] Wikipedia, "Plenoptic cameras." `http://lightfield-forum.com/en/`, accessed 2020-12-22.

[117] ViewPLUS, "Viewplus plenoptic camera." `http://www.viewplus.co.jp/products/product_newnex_products/`, accessed 2021-08-04.

[118] T. G. Georgiev and A. Lumsdaine, "Super-resolution with the focused plenoptic camera," Nov. 20 2012. US Patent 8,315,476.

[119] W. Ahmad, L. Palmieri, R. Koch, and M. Sjöström, "Matching light field datasets from plenoptic cameras 1.0 and 2.0," in *2018-3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, pp. 1–4, IEEE, 2018.

[120] B. R. Chen, I. A. Buchanan, S. Kellis, D. Kramer, I. Ohiorhenuan, Z. Blumenfeld, D. J. Grisafe II, M. F. Barbaro, A. S. Gogia, J. Y.

Lu, *et al.*, "Utilizing light-field imaging technology in neurosurgery," *Cureus*, vol. 10, no. 4, 2018.

[121] M. S. K. Gul and B. K. Gunturk, "Spatial and angular resolution enhancement of light fields using convolutional neural networks," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2146–2159, 2018.

[122] hviglobal, "Light field applications." `http://www.hviglobal.com/3D_Camera_Applications_English.htm`, Nov 2020. Accessed on 18-11-2020.

[123] N. Bedard, T. Shope, A. Hoberman, M. A. Haralam, N. Shaikh, J. Kovačević, N. Balram, and I. Tošić, "Light field otoscope design for 3D in vivo imaging of the middle ear," *Biomedical Optics Express*, vol. 8, no. 1, pp. 260–272, 2017.

[124] N. Bedard, I. Tošić, L. Meng, and K. Berkner, "Light field otoscope," in *Imaging Systems and Applications*, pp. IM3C–6, Optical Society of America, 2014.

[125] S. Saha, M. Tahtali, A. Lambert, and M. R. Pickering, "3D X-ray reconstruction using lightfield imaging," in *Advances in Computational Methods for X-Ray Optics III*, vol. 9209, p. 92090T, International Society for Optics and Photonics, 2014.

[126] A. Koz, C. Çığla, and A. A. Alatan, "Watermarking for light field rendering," in *IEEE 15th European Signal Processing Conference*, pp. 2296–2300, 2007.

[127] P. Paudyal, F. Battisti, A. Neri, and M. Carli, "A study of the impact of light fields watermarking on the perceived quality of the refocused data," in *IEEE 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, pp. 1–4, 2015.

[128] S. Kishk and B. Javidi, "3D object watermarking by a 3D hidden object," *Optics Express*, vol. 11, no. 8, pp. 874–888, 2003.

[129] A. Koz, C. Cigla, and A. A. Alatan, “Watermarking of free-view video,” *IEEE Transactions on Image Processing*, vol. 19, no. 7, pp. 1785–1797, 2010.

[130] A. Koz, C. Cigla, and A. A. Alatan, “Free-view watermarking for free-view television,” in *IEEE International Conference on Image Processing*, pp. 1405–1408, 2006.

[131] R. Zaharia, A. Aggoun, and M. McCormick, “Adaptive 3D-DCT compression algorithm for continuous parallax 3D integral imaging,” *Signal Processing: Image Communication*, vol. 17, no. 3, pp. 231–242, 2002.

[132] G. K. Wallace, “The JPEG still picture compression standard,” *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[133] A. B. Watson, “Image compression using the discrete cosine transform,” *Mathematica Journal*, vol. 4, no. 1, p. 81, 1994.

[134] B. Wohlberg and G. de Jager, “Fast image domain fractal compression by DCT domain block matching,” *Electronics Letters*, vol. 31, no. 11, pp. 869–870, 1995.

[135] K. Yamatani and N. Saito, “Improvement of DCT-based compression algorithms using Poisson’s equation,” *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3672–3689, 2006.

[136] G. J. Sullivan and T. Wiegand, “Video compression-from concepts to the H.264/AVC standard,” *Proceedings of the IEEE*, vol. 93, no. 1, pp. 18–31, 2005.

[137] K. Sayood, *Introduction to data compression*. Morgan Kaufmann, 2017.

[138] E. Y. Lam and J. W. Goodman, “A mathematical analysis of the DCT coefficient distributions for images,” *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1661–1666, 2000.

[139] K. Lee, D. S. Kim, and T. Kim, “Regression-based prediction for blocking artifact reduction in JPEG-compressed images,” *IEEE Transactions on Image Processing*, vol. 14, no. 1, pp. 36–48, 2004.

[140] Wikipedia, “Singular value decomposition.” `https://en.wikipedia.org/wiki/Singular_value_decomposition`, accessed 2021-08-26.

[141] K. Baker, “Singular value decomposition tutorial,” *The Ohio State University*, vol. 24, 2005.

[142] S. Kahu and R. Rahate, “Image compression using singular value decomposition,” *International Journal of Advancements in Research & Technology*, vol. 2, no. 8, pp. 244–248, 2013.

[143] M. Martínez-Corral, A. Dorado, H. Navarro, G. Saavedra, and B. Javidi, “Three-dimensional display by smart pseudoscopic-to-orthoscopic conversion with tunable focus,” *Applied Optics*, vol. 53, no. 22, pp. E19–E25, 2014.

[144] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, *et al.*, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[145] Z. Wang and A. C. Bovik, “A universal image quality index,” *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 81–84, 2002.

[146] D. Bhardwaj and V. Pankajakshan, “A JPEG blocking artifact detector for image forensics,” *Signal Processing: Image Communication*, vol. 68, pp. 155–161, 2018.

[147] M. Barni, F. Bartolini, V. Cappellini, and A. Piva, “A DCT-domain system for robust image watermarking,” *Signal Processing*, vol. 66, no. 3, pp. 357–372, 1998.

[148] Z. Xiong, K. Ramchandran, M. T. Orchard, and Y.-Q. Zhang, “A comparative study of DCT-and wavelet-based image coding,” *IEEE*

Transactions on Circuits and Systems for Video Technology, vol. 9, no. 5, pp. 692–695, 1999.

[149] W. Gao, S. Kwong, H. Yuan, and X. Wang, “DCT coefficient distribution modeling and quality dependency analysis based frame-level bit allocation for HEVC,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 139–153, 2015.

[150] C. Christopoulos, A. Skodras, and T. Ebrahimi, “The JPEG2000 still image coding system: an overview,” *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 1103–1127, 2000.

[151] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[152] S. Liu and A. C. Bovik, “Efficient DCT-domain blind measurement and reduction of blocking artifacts,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 12, pp. 1139–1149, 2002.

[153] D. Santa-Cruz, R. Grosbois, and T. Ebrahimi, “JPEG 2000 performance evaluation and assessment,” *Signal Processing: Image Communication*, vol. 17, no. 1, pp. 113–130, 2002.

[154] M. Asikuzzaman, M. J. Alam, A. J. Lambert, and M. R. Pickering, “Imperceptible and robust blind video watermarking using chrominance embedding: a set of approaches in the DT CWT domain,” *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 9, pp. 1502–1517, 2014.

[155] A. Papoulis, *Probability, Random Variables and Stochastic processes*. Cambridge, MA: McGraw-Hill, 1984.

[156] M. Podpora, G. Korba’s, and A. Kawala-Janik, “YUV vs RGB – choosing a color space for human-machine interaction,” *Annals of Computer Science and Information Systems*, vol. 3, pp. 29–34, 2014.

[157] J. Franco, G. Bernabé, J. Fernández, and M. E. Acacio, “A parallel implementation of the 2D wavelet transform using CUDA,” in *IEEE 17th Euromicro International Conference on Parallel, Distributed and Network-based Processing*, pp. 111–118, 2009.

[158] Polyvalens, “A really friendly guide to wavelets.” `http://www.polyvalens.com/blog/wavelets/theory/#7.+The+scaling+function+%5B7%5D`, August 2021. Accessed on 15-08-2021.

[159] Y. Sheng, “Wavelet transform,” in *The Transforms and Applications Handbook* (A. D. Poularikas, ed.), pp. 747–827, Boca Raton, FL (USA): CRC Press, 1996.

[160] S. G. Mallat, “A theory for multiresolution signal decomposition: the wavelet representation,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 7, pp. 674–693, 1989.

[161] S. Mallat, *A wavelet tour of signal processing*. Elsevier, 1999.

[162] S. G. Mallat, “Multiresolution approximations and wavelet orthonormal bases of $L^2(\mathbb{R})$,” *IEEE Transactions of the American Mathematical Society*, vol. 315, no. 1, pp. 69–87, 1989.

[163] R. C. Guido, “A note on a practical relationship between filter coefficients and scaling and wavelet functions of discrete wavelet transforms,” *Applied Mathematics Letters*, vol. 24, no. 7, pp. 1257–1259, 2011.

[164] L. Chun-Lin, “A tutorial of the wavelet transform,” 2010. National Taiwan University. `http://disp.ee.ntu.edu.tw/tutorial/WaveletTutorial.pdf`.

[165] W. Li, “Overview of fine granularity scalability in MPEG-4 video standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 301–317, 2001.

[166] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H. 264/AVC video coding standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[167] M. Alvarez-Mesa, C. C. Chi, B. Juurlink, V. George, and T. Schierl, “Parallel video decoding in the emerging HEVC standard,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1545–1548, IEEE, 2012.

[168] M. Sharma, A. Dhere, R. B. Pachori, and U. R. Acharya, “An automatic detection of focal EEG signals using new class of time–frequency localized orthogonal wavelet filter banks,” *Knowledge-Based Systems*, vol. 118, pp. 217–227, 2017.

[169] J. D. Villasenor, B. Belzer, and J. Liao, “Wavelet filter evaluation for image compression,” *IEEE Transactions on Image Processing*, vol. 4, no. 8, pp. 1053–1060, 1995.

[170] A. Alhasan, D. J. White, and K. De Brabanter, “Wavelet filter design for pavement roughness analysis,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 31, no. 12, pp. 907–920, 2016.

[171] W. Su, F. Wang, H. Zhu, Z. Zhang, and Z. Guo, “Rolling element bearing faults diagnosis based on optimal Morlet wavelet filter and autocorrelation enhancement,” *Mechanical Systems and Signal Processing*, vol. 24, no. 5, pp. 1458–1472, 2010.

[172] S. Yang, X. Gu, Y. Liu, R. Hao, and S. Li, “A general multi-objective optimized wavelet filter and its applications in fault diagnosis of wheelset bearings,” *Mechanical Systems and Signal Processing*, vol. 145, p. 106914, 2020.

[173] A. Kuzu, E. A. Baran, S. Bogosyan, M. Gokasan, and A. Sabanovic, “Wavelet packet transform-based compression for teleoperation,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 229, no. 7, pp. 639–651, 2015.