Anexo 1

Pseudocódigo de los algoritmos 1 y 2 definidos en la sección 4.6 y pseudocódigo del algoritmo que realiza el cómputo del SSIM. Las variables se muestran en color rojo y las funciones en color azul.

Anexo 1.1 Pseudocódigo del algoritmo 1

Inicio
procedimiento // Definir el procedimiento de inserción a utilizar
ataque // Definir el ataque a realizar así como sus características
clave // Definir la clave para la generación de la secuencia pseudoaleatoria
wm // Leer la marca de agua binaria, convertir la marca de agua binaria de matriz a vector
im // Leer imagen original
gama // Definir el factor de multiplicación
BC = descContourlet (im) // Realizar la descomposición contourlet de la imagen im
BCWM = Incrustarwm(BC, procedimiento, gama, clave) // Incrustar la marca de agua
im2 = sintesisContourlet(BCWM) // Reconstruir la imagen a partir de los coeficientes contourlet marcados
PSNR = psnr (im, im2) //Calcular la PSNR entre la imagen original y la imagen marcada
SSIM = ssim (im, im2) Calcular el SSIM entre la imagen original y la imagen marcada
Si ataque = JPEG, entonces
    Realizar compresión JPEG de la imagen marcada
Si ataque = ruido "sal y pimienta", entonces
    Atacar imagen con ruido "sal y pimienta"
Si ataque = ruido Gaussiano, entonces
    Atacar imagen con ruido Gaussiano
Si ataque = recorte, entonces
    Recotar imagen
Si ataque = filtrado
    Realizar filtrado paso bajas de la imagen
Si ataque = 0
    No realizar ataque alguno
wmrec = recwm(im2, procedimiento, clave) // Recuperar la marca de agua de la imagen marcada y reacomodar la marca de agua recuperada de vector a matriz
Mostrar la marca de agua recuperada
BER=biterr(wm, wmrec) Calcular la tasa de bits en error de la marca de agua recuperada respecto a la original
Fin

Anexo 1.2 Pseudocódigo del algoritmo 2

Inicio
procedimiento // Definir el procedimiento de inserción a utilizar
ataque // Definir el ataque a realizar así como sus características particulares
clave // Definir la clave para la generación de la secuencia pseudoaleatoria
wm // leer la marca de agua binaria
decisión // Definir el tipo de decisión a utilizar en la decodificación de la marca de agua
convertir la marca de agua binaria de matriz a vector
im // leer la imagen original
gama // Definir el factor de multiplicación
wmc = codifConv(wm) //Realizar la codificación convolucional de la marca de agua
BC = descContourlet (im) //Realizar la descomposición contourlet de la imagen original
BCWM = Incrustarwm(BC, procedimiento, gama, clave) Incrustar la marca de agua codificada
im2 = sintesisContourlet(BCWM) //Reconstruir imagen a partir de los coeficientes contourlet marcados
PSNR = psnr (im, im2) //Calcular la PSNR entre la imagen original y la imagen marcada
SSIM = ssim (im, im2) //Calcular el SSIM entre la imagen original y la imagen marcada
Si ataque = JPEG, entonces
    Relizar compresión JPEG de la imagen marcada
Si ataque = ruido "sal y pimienta", entonces
    Atacar imagen con ruido sal y pimienta
Si ataque = ruido Gaussiano, entonces
    Atacar imagen con ruido Gaussiano
Si ataque = recorte, entonces
    Recotar imagen
Si ataque = filtrado
    Realizar filtrado paso bajas de la imagen
Si ataque = 0
    No realizar ataque alguno
wmcrec = recwm(im2, procedimiento, clave) //Recuperar la marca de agua codificada de la imagen marcada
wmrec = decodeViterbi(wmcrec, decisión) //Realiza la decodificación de la marca de agua recuperda
Reacomodar la marca de agua recuperada en forma de matriz
Mostrar marca de agua recuperada
BER=biterr(wm, wmrec) // Calcular tasa de bits en error de marca de agua recuperada respecto a la original
Fin

Anexo 1.3 Funciones utilizadas en los algoritmos 1 y 2

función descContourlet (im)

Realiza la descomposición contourlet de la imagen im

Regresa BC, conjunto de matrices que contienen las bandas derivadas de la descomposición contourlet

función Incrustarwm(BC, procedimiento, gama, clave)

si procedimiento =1, entonces

Incrustar la marca de agua mediante la técnica de espectro disperso mejorado en la banda Y, utilizando el valor gama y la secuencia pseudoaleatoria generadas por la clave.

si procedimiento =1, entonces

Incrustar la marca de agua mediante la técnica de espectro disperso mejorado en las subbandas Z3, Z4, Z7 y Z8, utilizando el valor gama y la secuencia pseudoaleatoria generadas por la clave.

si procedimiento =3, entonces

Incrustar la marca de agua mediante la técnica de espectro disperso mejorado en las subbandas Y1, Y4, Y5, Y8, Z9, Z16, utilizando el valor gama y la secuencia pseudoaleatoria generadas por la clave.

Regresa BCWM, conjunto de matrices que contienen las bandas contourlet marcadas con la marca de agua

función sintesisContourlet(BCWM)

Realiza la reconstrucción de la imagen original a partir de las bandas contourlet marcadas

regresa im2, imagen marcada

función recwm(im2, procedimiento, clave)

Realiza la recuperación de la marca de agua incrustada en la las bandas contourlet definidas por procedimiento de la imagen im2. Para hacer la recuperación reproduce la secuencia pseudoaleatoria generada a partir de clave.

Regresa wmrec, vector binario que contiene los bits incrustados en las bandas contourlet de la imagen.

función codifConv(wm)

Realizar la codificación convolucional de la marca de agua

regresa wmc, la marca de agua codificada

función decodeViterbi(wmcrec, decisión)

Si decisión = dura, entonces

Realiza la decodificación de la marca de agua utilizando decisión dura

Si decisión = suave, entonces

Realiza la decodificación de la marca de agua utilizando decisión suave con 8 niveles de decisión

regresa wmrec, la marca de agua decodificada

función biterr(wm, wmrec)

Calcular tasa de bits en error de marca de agua recuperada respecto a la original

regresa BER, la tasa de bits en error en la marca de agua recuperada

función psnr(im, im2)

Calcula la PSNR entre la imagen original y la imagen marcada

Regresa PSNR.

función SSIM(x, y)

Computar el promedio de intensidad sobre ventanas de tamaño 11 × 11 para la imagen x y la imagen y

Computar la desviación estándar sobre ventanas de tamaño 11 × 11 para la imagen x y la imagen y.

Computar la covarianza entre la imagen x y la imagen y

Evaluar el índice SSI de similaridad estructural para cada una de las ventanas

Hacer un promedio del índice SSI de cada una de las ventanas de tamaño 11 × 11 pixeles. El promedio es SSIM

Regresa SSIM