Improved DCT-based detection of copy-move forgery in images

**INTRODUCCION**

Con el desarrollo de las técnicas sofisticadas para la manipulación de imágenes digitales y el bajo costo para obtener una imagen digital de alta calidad, cualquier persona puede “operar” una imagen fácilmente sin dejar pistas visibles. Como resultado, ya no se puede confiar en las imágenes digitales y no tendrán la estatura única como un registro definitivo de un evento. Para recuperar la confianza de las personas hacia la autenticidad de la imagen digital, es necesario desarrollar un conjunto de métodos para autenticar imágenes digitales.

Ya se ha utilizado la marca de agua digital en esta área y se han realizado muchas investigaciones (para una encuesta general, véase [1]). Aunque existen varios tipos de esquemas de marcas de agua de autenticación, como marcas de agua frágiles borrables [2], marcas de agua semi-frágiles [3] y marcas de agua robustas [4], todos estos esquemas asumen que la marca de agua digital incrustada en imágenes digitales Ser alterada después de la manipulación digital, por lo que esto requiere que las filigranas se inserten en las imágenes digitales cuando se toman. Este requisito es el principal inconveniente de estos esquemas, ya que sólo las cámaras digitales costosas especializadas integran la marca de agua en la imagen cuando se toma una foto.

Por el contrario, existen otros métodos que se denominan técnicas de detección ciega que no necesitan la presencia de marcas de agua digitales. Estos enfoques trabajan asumiendo que cualquier forma de alteración alteraría las estadísticas subyacentes de una imagen.

Hay algunos procesos clásicos utilizados en la manipulación y uno de ellos es el proceso de copia-movimiento. Un proceso de copia-movimiento se utiliza generalmente para ocultar un objeto especial en la imagen original. Para ello, una región en una imagen se copia y se pega en otra región en la misma imagen. Aunque algunos post proceso, como el borde de suavizado, difuminado y el ruido de adición, se tomaría para eliminar las pistas visibles, el proceso ha creado dos partes casi idénticas en la imagen alterada. Esto no es común en las imágenes naturales y por lo tanto puede ser utilizado para detectar este tipo de imágenes.

Se han desarrollado algunos métodos de detección de movimiento de copia. En [5], Fridrich primero analizó la búsqueda exhaustiva y luego propuso un método de detección de coincidencia de bloques basado en la transformación discreta coseno (DCT) que es más eficaz que la búsqueda exhaustiva. Popescu propuso un método similar [6], que utiliza el análisis de componentes principales (PCA) en lugar de DCT. Utilizando una de las características de PCA, Popescu utilizó aproximadamente la mitad de los números de las características utilizadas por Fridrich y, al hacerlo, su método se demuestra para ser más eficaz. Pero el método también tiene sus inconvenientes. Uno de ellos es que no funciona cuando la región de copia se gira ligeramente antes de que se pegue. Por el contrario, el método de Fridrich puede detectar un movimiento de copia-manipulación con la rotación de hasta 58 [7]. Recientemente, se presentaron algunos métodos que pueden detectar post-procesamiento más sofisticado, como el trabajo de Mahdian [8] puede detectar la región duplicada incluso la región de movimiento de copia se ha difuminado o se ha añadido ruido.

En este trabajo, proponemos un algoritmo de detección basado en DCT. Debido a que las ideas que aparecen en la técnica de Popescu [6], como la reducción de la dimensión y el control de coincidencia, se introducen en el algoritmo, el marco del algoritmo es como el de Popescu. Con los coeficientes de DCT como características, el algoritmo puede hacer un poco mejor en el caso de la detección de manipulación de movimiento de copia con rotación en comparación con la técnica de Popescu. En comparación con la técnica de Fridrich [5], nuestro algoritmo será más rápido debido a la reducción de dimensión, mientras que la capacidad de detección no se reduce. La robustez de la compresión JPEG con diferentes factores de calidad, borrosidad gaussiana y ruido gaussiano blanco aditivo se demuestra por una serie de experimentos, y estos experimentos están ausentes en [5]. El resto del trabajo se organiza de la siguiente manera: en la sección 2 se describe el método de detección de falsificaciones propuesto. Los resultados del experimento se muestran en la Sección 3. La conclusión se presenta en la Sección 4.

**ALGORITMO PROPUESTO**:

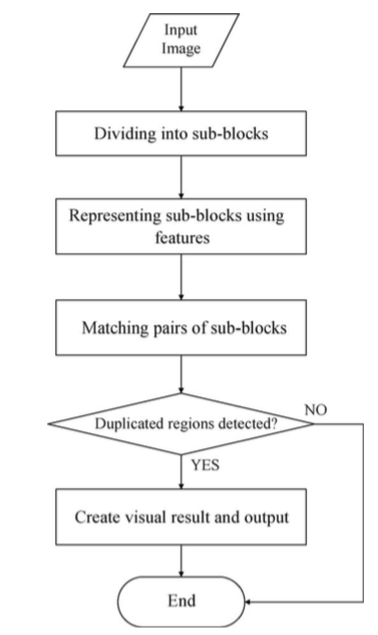
Debido a la naturaleza de la duplicación de regiones, debe haber al menos dos regiones similares en la imagen manipulada. Una imagen natural, por el contrario, es muy poco probable que tenga dos grandes regiones similares (excepto para las imágenes que tienen dos grandes regiones lisas). Por lo tanto, una imagen puede considerarse como copia-movimiento alterada si se detectan dos regiones grandes similares. Aquí asumimos que las regiones duplicadas no se superponen. La tarea del método de detección es determinar si una imagen contiene regiones duplicadas. Dado que la forma y el tamaño de las regiones son desconocidos, es definitivamente computacionalmente imposible tratar de examinar todos los posibles pares de regiones con diferentes formas y tamaños. Es más eficaz dividir una imagen en bloques superpuestos de tamaño fijo y probar si se duplican pares de bloques.

Antes de la prueba, debe realizarse otro paso importante, para representar los bloques por características. Un buen método de extracción de características puede hacer que el algoritmo de detección sea efectivo y robusto. Después de que todos los bloques se representan correctamente por algunas características, el algoritmo continúa en el paso correspondiente. Como las regiones similares tendrían características similares, si las características se ordenan en un orden lexicográfico antes de hacer coincidir, puede hacer que las regiones similares sean consecutivas y hacer que la coincidencia sea más efectiva. Para cada par de características se comprueba si coinciden. En el caso de que un par de características coincidan, se supone que los bloques correspondientes son similares. Si muchos de esos bloques son adyacentes, puede haber ocurrido una alteración de movimiento de copia en la imagen de entrada. Las regiones duplicadas detectadas se pueden marcar en una imagen de mapa. El operador puede juzgar si una imagen ha sido alterada de acuerdo con la imagen del mapa.

* MARCO DEL ALGORITMO:

Las discusiones anteriores llevan al marco del algoritmo de detección de movimiento de copias, que también se muestra en la Fig. 1.

* Paso 1-División: Divide la imagen de entrada en sub-bloques.
* Paso 2-Representación: Representa cada sub-bloque usando las características apropiadas.
* Paso 3: hacer coincidir: probar cada par de sub-bloques si son similares.
* Paso 4-Salida: Salida de las regiones duplicadas (si existen) visualmente.
* La siguiente es la forma en que el algoritmo se implementa en el marco. Un cuadrado con pixels B B se desliza una vez a lo largo de la imagen desde la esquina superior izquierda hasta la esquina inferior derecha. Supongamos que la imagen tiene M N píxeles, el deslizamiento generará (M B + 1) (N B + 1) bloques.



DCT se aplica a cada bloque para extraer las características. Los coeficientes de DCT se reconforman en un vector de hileras en orden de zigzag, por lo que se agrupan frecuencias similares y se generan vectores de fila (M B + 1) (N B + 1) cuando se extraen las características de todos los bloques. A continuación, los vectores se ordenan en orden lexicográfico. Sea la matriz A los vectores ordenados, el tamaño de A será (M B + 1) (N B + 1) B2.

La i-ésima fila de A se denota como ai, y la coordenada de la esquina superior izquierda del bloque correspondiente se denota como (xi, yi). En el paso de coincidencia, cada pareja de vector consecutivo ai, aj. Aj se prueba si son similares. Si lo son, el vector de cambio s entre los dos bloques correspondientes se calcula como:

S = (s1, s2) = (xi-xj, yi-yj)

Obsérvese que s y s corresponden al mismo turno, por lo que multiplicar s por 1 puede ser necesario para asegurarse de que s1 >= 0. Se puede llamar una normalización. Entonces el algoritmo explora todo el vector de desplazamiento s, y busca los vectores s(1), s(2),. . ., S(n) cuya ocurrencia excede un umbral predefinido T.

Para cada s(i), se encuentran y se colorean todos los bloques coincidentes que contribuyeron a este cambio de vectores. Este algoritmo emite una imagen de mapa negro, y las regiones que se consideran duplicadas están marcadas con un color especial distinto del negro. Con la imagen del mapa y el juicio visual humano, se puede determinar rápidamente si la imagen de entrada es original o alterada.