Introducción de marcas de agua en imágenes mediante LSB y resistencia a transformaciones y ruido

## J Sardón. Carlos

**Abstract**

La naturaleza digital de la información hace que su manipulación y copia sea algo trivial e inherente a esta, apareciendo usos no deseados que pueden ser detectados o evitados mediante marcas de agua. Estas pueden ser introducidas en cualquier soporte digital con finalidades como certificar la legitimidad de este, evitar o evidenciar usos no autorizados por los propietarios u ocultar información. Mediante MATLAB implementaremos la introducción de marcas de agua en imágenes mediante el método del bit menos significativo (LSB) y analizaremos cómo reacciona frente a distintas modificaciones como ruido, compresión o alteraciones intencionadas.

**Palabras clave:** marca de agua, ruido, ocultación, transformación, lsb

# **Introducción**

Las marcas de agua en las imágenes es un concepto intuitivo que consiste en introducir una imagen en otra, ya sea de forma más o menos transparente para cualquier espectador. Si bien su popularización llegó con Internet, existía ya mucho antes en el medio físico para evitar la falsificación de billetes o indicar el fabricante de folios de papel, ambos aún presentes.

Las imágenes son transmitidas con gran facilidad, y normalmente, no necesitan de metadatos o algún otro archivo para tener sentido por si mismas. A la vez que supone una gran ventaja para la distribución de estas, también hace que conceptos como la autoría o los derechos sobre las mismas sean prescindibles y normalmente no se tengan en cuenta.

Con las marcas de agua se brinda un método para obtener seguridad sin perder las ventajas anteriormente expuestas: el contenido de la imagen para el espectador (debe) permanece igual, no se necesita de información adicional a la que hay en la imagen y la facilidad de transmisión es similar.

El marcado puede ser visible, pero para el fin que se plantea aquí buscamos que sea lo más transparente posible y que el espectador no note diferencia alguna entre la imagen original y la que tiene el marcado. De esta forma se entra en el campo de la esteganografía, que consiste la ocultación de mensajes o información dentro de otros.

En el procesamiento de imágenes digitales, y por tanto en el marcado de agua como parte de este, se puede trabajar dentro del dominio espacial o de la frecuencia. En este documento solo trabajaremos con el primero a través del método del bit menos significativo (LSB) y experimentaremos con distintas aplicaciones y posteriores modificaciones para estudiar su resistencia.

# **Planteamiento teórico**

Una imagen digitalizada, al margen de su formato, consiste en una matriz de valores por cada canal de color (uno para escala de grises, tres para RGB). Es en esta estructura donde se debe conseguir introducir una marca de agua, que en el caso más sencillo será una matriz binaria, teniendo en cuenta los siguientes factores:

* Transparencia: la marca insertada en la imagen original solo debe poder verse en caso de que sea recuperada de forma intencionada, y en ningún caso debe modificar aparentemente el contenido original, ni suponer una perdida noticiable en este.
* Seguridad: la información insertada no debe poder ser interceptada por un agente no deseado.
* Saturación: en función de las características de la imagen original, hay una cantidad de información máxima que podremos insertar.
* Resistencia: nuestra marca de agua debe ser capaz de resistir a modificaciones de la imagen, ya sean con el fin de atacar el marcado o no.

Con estas cuestiones experimentaremos jugando con conceptos básicos relacionados de forma directa con cómo se almacena la información. Si bien las cuatro están relacionadas entre si y el hecho de centrarse en una puede hacer que las otras sufran, buscaremos el mejor equilibrio entre todas.

En la **transparencia** hay dos factores a tener en cuenta: no se debe ver la marca insertada y esta no debe afectar a la imagen original. Al ser el último bit el que modificamos para introducir la marca, es despreciable la diferencia que produce con respecto a la imagen original. Si se quisiera introducir la marca con más información que un único bit por pixel, una opción que aseguraría que no sea noticiable es la **encriptación.** Se podrían usar más bits más allá del menos significativo sin que la marca de agua sea más evidente, ya que al estar encriptada, si bien su información estará en las matrices, no será de una forma representable. Hay que buscar un equilibrio entre número de bits que se usan para introducir información y conservación de la imagen original.

En cuanto a la **resistencia** a modificaciones, el hecho de encriptar la información que introduzcamos para la marca hace que si hay alguna pérdida el resultado sea peor que si simplemente modificamos el último bit de cada pixel según la marca binaria, ya que la información estaría encriptada en conjunto y podría perder sentido. A priori, la modificación más fatal para el LSB se antoja la compresión, por el hecho de que sea un único bit el que guarda la información crucial para nuestro propósito y que la compresión busca la reducción de tamaño del archivo.

# **Resolución Práctica**

Esta sección debe incluir la descripción de la implementación (pero no el código), especificando las tecnologías usadas, cómo se ha diseñado la aplicación, cuáles son sus módulos o partes principales. *Debe quedar muy claro cuál es la parte original implementada en el programa, qué librerías se han usado y de dónde se han cogido, con referencias apropiadas.*

Se debe describir (tanto en la aplicación, como en la documentación, como en la presentación) los pasos seguidos para resolver el problema que se plantea de la forma más divulgativa posible.

Se puede usar el lenguaje de programación que se quiera. Se pueden usar librerías o códigos fuentes de trabajos de otros años, de internet, etc., siempre y cuando se referencien adecuadamente y se razone por qué se usan y qué realiza esa implementación.

Si se usa java para un trabajo de imágenes 2D, se puede usar el paquete ImageJ que se puede descargar en <http://rsbweb.nih.gov/ij/>. El trabajo dirigido se puede insertar como un plugin. También existen paquetes específicos para trabajar con imágenes 3D. Si se desea trabajar en C++ o en Python, se recomienda usar la librería openCV, que se puede descargar de la página <https://opencv.org/releases.html>. Matlab posee también un toolbox de procesamiento de imágenes que se puede usar en los laboratorios de la escuela. Éstas sólo son algunas indicaciones, no hay por qué usar nada de ello.

Se recomienda realizar una planificación inicial del trabajo, estableciendo los hitos importantes en el desarrollo, tareas que hay que realizar y asignaciones de esas tareas a miembros del grupo. Se puede usar cualquier gestor de proyectos para realizar esta planificación inicial y posterior seguimiento del desarrollo del trabajo. Téngase en cuenta que las horas de dedicación de **cada alumno** han de ser **70 horas** (correspondientes a 7 semanas de trabajo del alumno). Será obliga- torio también presentar la revisión final de esta planificación o tabla de tiempos (Anexo I).

# **Experimentación**

Una sección de ejemplos comentados y pruebas realizadas con el programa desarrollado es impre- scindible en este trabajo. Igualmente importantes serán las conclusiones que se puedan obtener de la experimentación realizada.

# **Manual de usuario**

Se debe incluir un breve manual de usuario. Este manual se incluirá también en la misma aplicación.

# **Conclusiones**

Introducir una sección de conclusiones que incluya propuestas claras de mejora o extensión del trabajo (por ejemplo, si no se han podido alcanzar todos los objetivos iniciales). También con- clusiones sobre los resultados obtenidos, en qué medida difieren de los esperados. También son apropiadas conclusiones sobre las desviaciones en cuanto a la planificación inicial y conclusiones sobre la experiencia de la realización del trabajo.

# **Referencias**

Las referencias se citan así: bla, bla [[2](#_bookmark1)], bla, bla [[1](#_bookmark0)]. La bibliografía debe seguir el estilo de este documento. A continuación aparecen dos ejemplos de referencias bibliográficas: un artículo en una revista y un libro.

\begin{thebibliography}{9}

\bibitem{clave:revista}

Y. O. Mismo, ‘‘Título del artículo", \emph{Revista Publicación Periódica}, Vol. 17, pp. 1-100, 1997.

\bibitem{clave:libro}

U. N. Experto, \emph{Un libro que escribí}, Editorial, 1996.

\end{thebibliography}

El resultado de estos ejemplos puede verse a continuación, con las referencias ordenadas al- fabéticamente por autores.

# **References**

1. U. N. Experto, *Un libro que escribí*, Editorial, 1996.
2. Y. O. Mismo, “Título del artículo”, *Revista Publicación Periódica*, Vol. 17, pp. 1-100, 1997.

**Anexo I: Tabla de tiempos**

Se debe justificar el trabajo realizado por cada componente del grupo, indicando el **tiempo total que cada miembro del grupo ha dedicado** al trabajo (lo que puede implicar diferencia de notas obtenidas por los distintos miembros del grupo). El trabajo realizado debe ser de **70 horas por alumno**. Además, debe haber un plan de trabajo detallado con las tareas realizadas por cada miembro del grupo. Para esto último, se puede usar la tabla siguiente o bien documentos generados por la herramienta de gestión de proyectos que se use, como Projetsii o Microsoft Project, por ejemplo.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fecha de la actividad | Inicio | Fin | Tiempo | Miembros | Actividad realizada |
|  |  |  |  |  |  |

**Anexo II: Cómo introducir ciertos elementos en LaTex**

# **More about how to write an article in Overleaf**

## **How to include Figures**

First you have to upload the image file from your computer using the upload link the project menu. Then use the includegraphics command to include it in your document. Use the figure environment and the caption command to add a number and a caption to your figure. See the code for Figure [1](#_bookmark2) in this section for an example.



Figure 1: This frog was uploaded via the project menu.

## **How to add Tables**

Use the table and tabular commands for basic tables. See Table [1](#_bookmark3), for example.

Item Quantity Widgets 42

Gadgets 13

Table 1: An example table.

## **How to write Mathematics**

LATEX is great at typesetting mathematics. Let *X*1*, X*2*, . . . , Xn* be a sequence of independent and identically distributed random variables with E[*Xi*] = *µ* and Var[*Xi*] = *σ*2 *< ∞*, and let

*n*

*S* = *X*1 + *X*2 + *· · ·* + *Xn* = 1 L *X*

*i*

*n*

*n*

*n*

*i*

denote their mean. Then as *n* approaches infinity, the random variables *√n*(*Sn − µ*) converge in distribution to a normal *N* (0*, σ*2).

## **How to create Sections and Subsections**

Use section and subsections to organize your document. Simply use the section and subsection buttons in the toolbar to create them, and we’ll handle all the formatting and numbering auto- matically.

## **How to add Lists**

You can make lists with automatic numbering . . .

* + 1. Like this,
    2. and like this.

. . . or bullet points . . .

* Like this,
* and like this.