UNIVERSIDAD DE GUANAJUATO

Educación

Compresión y encriptación de información utilizando SVD

UG VERANOS DE LA CIENCIA 25 ANIVERSARIO

➔ **Eje investigación:**

° *GRACE EDITH MOLINA CHAVEZ*

(Lic. en Matemáticas, UGTO, Campus Guanajuato)

° *DAVID ORLANDO PALACIOS MORAN*

(Lic. en Ing. Electrónica, Universidad San Carlos de Guatemala)

➔ **Eje educación:**

° *GABRIEL ALBERTO TORRES MENDOZA*

(Bach. de Ingeniería, CNMS, Escuela de Nivel Medio Superior de Moroleón)

Veranos de la ciencia UG 2019

Educación

13 de julio de 2019

Contenido

1 MOTIVACIÓN

La llamada ``era de la información´´, además de grandes avances ha traído nuevos problemas y retos, entre ellos el que representa el almacenamiento y envío de una enorme cantidad de datos, imágenes o archivos de todo tipo (muchos de ellos de gran tamaño), así como el resguardar su seguridad al transitar por las redes.

Partiendo de la problemática aún latente, se desarrolló el proyecto de encriptación y compresión de información utilizando descomposición en valores singulares (SVD), buscando, de ser posible, aportar datos de valor al campo de estudio, específicamente, en el procesamiento de imágenes y señales.

2 INTRODUCCIÓN

El presente trabajo es realizado como parte de los resultados alcanzados/productos realizados durante el VERANO DE LA CIENCIA DE LA UNIVERSIDAD DE GUANAJUATO 2019, **área de educación**, en el proyecto de **``Compresión y encriptación de información utilizando descomposición en valores singulares''**, con el fin de servir como una guía breve pero directa y concisa sobre el proceso de compresión y encriptado de imágenes digitales, así como su procesamiento en MATLAB.

Contiene información básica necesaria para poder realizar los procesos mencionados y comprenderlos, estando dirigido a alumnos del nivel medio superior y a estudiantes de carreras como informática y computación o afines, como curso introductorio básico a lo que es la imagen digital, su procesamiento, MATLAB y cómo utilizarlo en el procesamiento de imágenes, así como al álgebra lineal básica necesaria para comprender los procesos realizados por MATLAB, al menos en términos generales.

La parte del documento enfocada al tutorial está además apoyada por contenido audiovisual del manejo de Matlab, en el que se explica de manera más detallada el proceso de compresión y encriptado de las imágenes, así como la demostración de que el algoritmo implementado funciona.

Cabe aclarar que el algoritmo utilizado para el encriptado/desencriptado en este documento no es el definitivo desarrollado en el programa Veranos de la Ciencia UG 2019 en el proyecto, sino que se optó por utilizar uno más sencillo a fin de que el documento no pierda el carácter de ¨básico y de introducción¨ y cualquiera pueda realizarlo comprendiendo de la mejor manera posible los procesos que se están llevando a cabo. Si desea conocer el algoritmo final desarrollado puede buscar el informe de nuestro proyecto en la revista virtual **``jóvenes en la ciencia´´ de la Universidad de Guanajuato.**

3 Relación del álgebra lineal con las imágenes

En grados anteriores a la universidad se tiene un acercamiento a los sistemas de ecuaciones lineales y con ellos, a las matrices, usualmente para resolverlos utilizando métodos como el de Gauss o Gauss-Jordan. Para esto, la matriz se construye con los coeficientes de las ecuaciones lineales a resolver, junto a sus igualdades (Matriz aumentada), con el fin de modificarla usando las operaciones elementales de la fila (reemplazo, intercambio y escalamiento) y así conseguir el ¨conjunto solución¨ que da validez a las ecuaciones al sustituirlo en las variables.

Sin embargo, esa no es la única utilidad de las matrices, sino que éstas tienen infinidad de aplicaciones útiles puesto que se pueden realizar muchos procesos con ellas, como operaciones elementales (suma, resta, multiplicación), y la posibilidad de manipularlas, por ejemplo, cambiando las filas por las columnas y viceversa (trasponer). El conjunto de esas posibilidades hace que las matrices tengan una vital importancia en el día a día del mundo moderno, por ejemplo, las matrices se usan para los gráficos por ordenador, la pantalla es una matriz de pixeles y multiplicando o sumando por las matrices adecuadas podemos hacer giros, rotaciones, traslaciones, y en general movimientos con los gráficos, siendo ésta la manera en que se consiguen los gráficos en 2D y en 3D de los videojuegos.

El álgebra lineal es la rama de las matemáticas que estudia conceptos tales como vectores, matrices, sistemas de ecuaciones lineales y en un enfoque más formal, espacios vectoriales, y sus transformaciones lineales.

Las matrices pueden modelar cualquier tipo de sistema, por lo que resulta en una poderosa herramienta para manipular, igualmente, cualquier tipo de sistema, y las imágenes no son la excepción.

Ejemplos de matrices son los siguientes:

A =

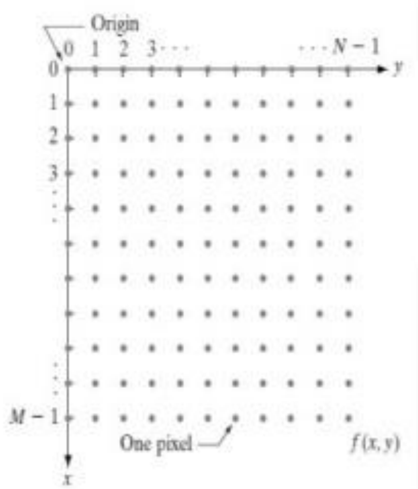
$$\begin{matrix} 42 & 23 & 54 \\ 23 & 65 & 98 \\ 26 & 64 & 89 \end{matrix}$$

B =

$$\begin{matrix} 4 & 7 & 9 & 3 \\ 9 & 6 & 1 & 4 \\ 12 & 4 & 10 & 8 \end{matrix}$$

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{pmatrix}$$

Una imagen es una matriz de pixeles y desde el punto de vista matemático nos podemos referir a una imagen como una función bidimensional f(x,y) de intensidad luminosa denominada nivel de gris, donde **x** e **y** son las coordenadas espaciales y el valor de la función **f** en cada punto (x, y) es proporcional a la luminosidad de la imagen en dicho punto.



$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, N-1) \\ f(1,0) & f(1,1) & \cdots & f(1, N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1, 0) & f(M-1, 1) & \cdots & f(M-1, N-1) \end{bmatrix}..$$

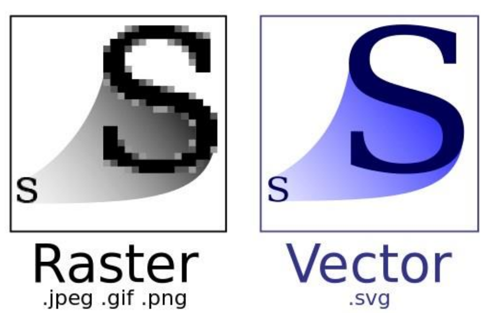
MyN son enteros positivos mientras que L (Nro de niveles discretos permitidos para cada pixel) es típicamente una potencia entera de dos

$L=2^k$ Rango → [0 - L-1]

Alto rango dinámico → Buen contraste

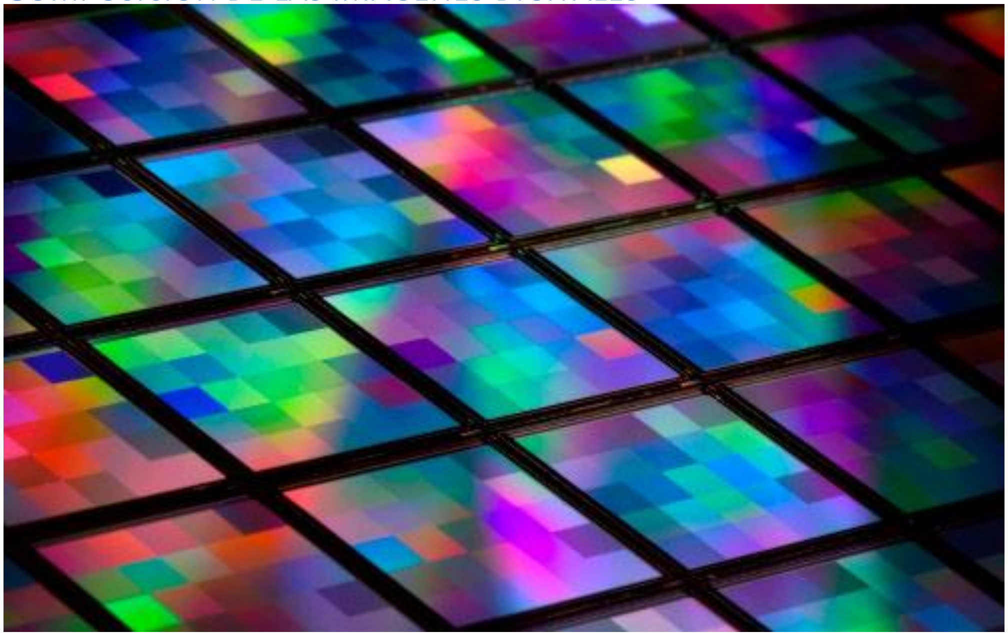
Las imágenes se obtienen por medio de periféricos tales como escáneres y cámaras, la imagen se representa por patrones de bits generados por el periférico correspondiente.

Hay dos formas básicas de representación de imágenes, mapa de bits (o matriciales) y mapa de vectores (o basada en objetos).



Nosotros trabajaremos solamente con imágenes matriciales y es necesario aclarar que para manipular la imagen dentro de un ordenador, la función imagen f(x, y) debe ser digitalizada, éste proceso se explicará en el apartado de ¨Digitalización¨, más adelante.

4 COMPOSICIÓN DE LAS IMÁGENES DIGITALES



Una imagen digital es una representación bidimensional de una imagen a partir de una matriz numérica, es un **mapa de bits**, de millones de puntos de colores.

Las imágenes están compuestas de pixeles, el pixel es la unidad mínima de información de la imagen, conteniendo información de color, saturación y brillo.

Más pixeles= mayor resolución.

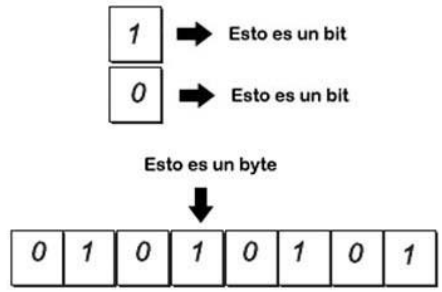
Megapíxeles= millones de pixeles que el sensor de una cámara es capaz de capturar en el momento de capturar la imagen, sirve para medir la resolución de la imagen. Así una cámara de 24MP puede producir una imagen de 4500px de alto y 5000 pixeles de ancho = 24,000,000px siendo la cantidad de MP efectivos que tiene el sensor de la cámara y determina la resolución.

Resolución de la imagen es la cantidad de filas y columnas de pixeles que tenga de alto y ancho un archivo digital. La resolución se define por la densidad de pixeles en una pulgada dpi, a mayor número de pixeles/puntos por pulgada se tendrá mayor definición o resolución.

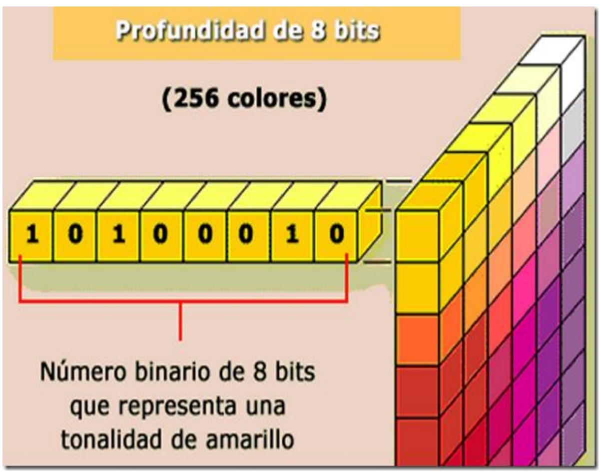
Calidad de la imagen es la combinación de la resolución de la imagen, profundidad del color y su tamaño que determinan la cantidad de detalles que pueden observarse en éstas.

Tamaño del archivo digital. Dado que la imagen digital es un archivo, la cantidad de información en ella se mide en bites o bytes. Bite es la abreviación de Binary Digit (dígito binario), que es la menor unidad de información de una computadora, un bit tiene solamente un valor, que puede ser 0 ó 1, varios bites originan los bytes (8 bits), megabites, gigabites, etc.

Toda **la información procesada por una computadora es medida y codificada en bits.** El tamaño de los archivos es medido en bits, l**as tasas de transferencia son medidas en bit,** toda la información en el lenguaje del usuario es convertida a bits para que la computadora la «entienda».



Los Bits también **son utilizados para la clasificación de colores de una imagen.** Por ejemplo: una imagen monocromática tiene 1 bit en cada punto (blanco o negro), mientras una imagen de 8 bits soporta hasta 256 colores.

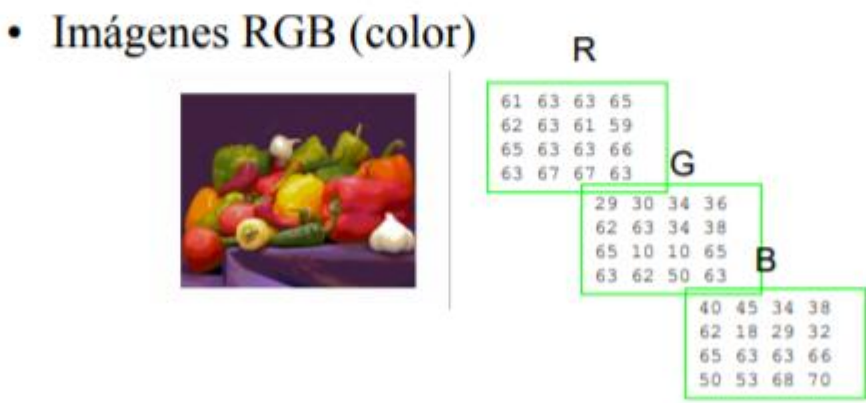


La **PROFUNDIDAD DE BITS o profundidad de color** es el número de bits necesarios para describir el color de un pixel. Cuanto mayor sea la profundidad de bits, tanto mayor será la cantidad de tonos (escala de grises o color) que puedan ser representados. Las imágenes digitales se pueden producir en blanco y negro (en forma bitonal), a escala de grises o a color. Una ***imagen bitonal*** está representada por **píxeles que constan de 1 bit cada uno**, que pueden representar dos tonos (típicamente negro y blanco), **utilizando los valores 0 para el negro y 1 para el blanco o viceversa.**

Una **imagen *a escala de grises*** está compuesta **por píxeles** representados por múltiples bits de información, **que típicamente varían entre 2 a 8 bits o más**.

Ejemplo: En una imagen de 2 bits, existen cuatro combinaciones posibles: 00, 01, 10 y 11. Si "00" representa el negro, y "11" representa el blanco, entonces "01" es igual a gris oscuro y "10" es igual a gris claro. La profundidad de bits es dos, pero la cantidad de tonos que pueden representarse es 2^2 ó 4. A 8 bits, pueden asignarse 256 (2^8) tonos diferentes a cada píxel.

Una ***imagen a color*** está típicamente representada por una **profundidad de bits entre 8 y 24 o superior a ésta**. **En una imagen de 24 bits, los bits por lo general están divididos en tres grupos: 8 para el rojo, 8 para el verde, y 8 para el azul.** Para representar otros colores se utilizan combinaciones de esos bits**. Una imagen de 24 bits ofrece 16,7 millones (2^{24}) de valores de color.** Cada vez más, los escáneres están capturando 10 bits o más por canal de color y por lo general imprimen a 8 bits para compensar el "ruido" del escáner y para presentar una imagen que se acerque en el mayor grado posible a la percepción humana.



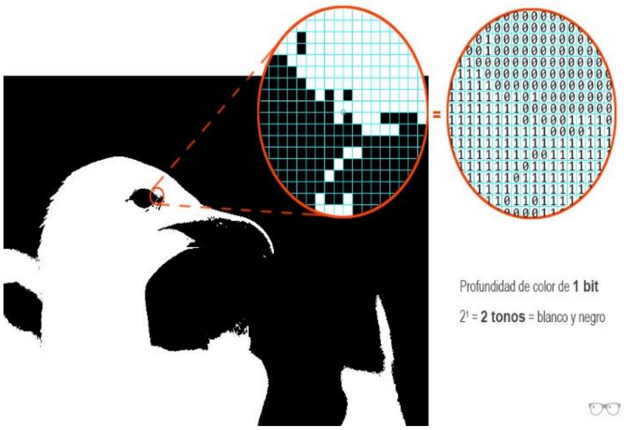
Profundidad de color	Tipo de imagen	Colores
1bit	Blanco y negro	2 colores
8bits	Escala de grises	256 tonos de gris
4 bits	Color	16 colores
8 bits	Color	256 colores
16 bits	Color	65.000 colores
16 bits	Escala de grises	65.000 tonos de grises
24 bits	Color, RGB	16,7 millones de colores
32 bits	Color, CMYK	16,7 millones de colores
48 bits	Color	281.000 millones de colores

5 DIGITALIZACIÓN

Digitalizar imagen

Digitalizar una imagen significa convertirla en un archivo que puede ser manipulado por la computadora, es decir en un conjunto de bits. Para digitalizar una imagen es necesario dividirla en unidades discretas, cada una de las cuales se llama píxel. Una vez dividida la imagen, se le asigna un valor a cada uno de los pixeles.

Imagen en blanco y negro



En el caso de una imagen en blanco y negro, si la mayor parte del pixel es negro, se le asigna el valor de 1, y si la mayor parte es blanco, se le asigna valor 0. La cantidad de pixeles que forman una imagen se llama resolución. Cuanto mayor es el número de píxeles utilizados para definir una imagen, mayor es el realismo. La resolución de una imagen indica que cantidad de píxeles por pulgada (PPI) se utilizó para componer esa imagen.

Imagen color.

Para digitalizar imágenes en color, se utiliza el mismo procedimiento, aunque es necesario asignarle a cada píxel un número mayor de bits para asignarle a la información sobre los colores. En la siguiente tabla se observa la cantidad de colores posibles según el número de bits utilizados para definir la profundidad del color:

1 bits	2^1	2 tonos (Blanco y negro)
2 bits	2^2	4 tonos (Escala de grises)
3 bits	2^3	8 tonos (Escala de grises)
4 bits	2^4	16 tonos (Escala de grises)
8 bits	2^8	256 tonos (Escala de grises)
16 bits	2^{16}	65.536 tonos (Color)
24 bits	2^{24}	16,7 millones de tonos (Color real)

Como ya se ha mencionado, los archivos de imágenes digitales suelen ser de gran tamaño, lo que hace difícil su procesamiento, manipulación y transporte. Para reducir el tamaño de los archivos se pueden utilizar distintos métodos, como el que se implementó en el proyecto de investigación (SVD), aunque ya existen algunos procedimientos de compresión, encargados de reducir la cantidad de información (bits) necesaria, con el límite de que dicha reducción no altere la percepción de la imagen.

Ejemplo:

100 dpi compresión JPEG baja (Tamaño de archivo 248 Kb)
100 dpi compresión JPEG media (Tamaño de archivo 49 Kb)
100 dpi compresión JPEG alta (Tamaño de archivo 22 Kb)

Existen varios formatos para comprimir el tamaño de las imágenes entre los más conocidos es el JPEG (en inglés Joint Photographic Expert Group, a menudo abreviado JPG)/ JFIF (JPEG File Interchange Format).

5.1 PROCESAMIENTO DE IMÁGENES

El procesamiento de imágenes se trata del estudio de algoritmos que toman una imagen como datos de entrada y devuelven una imagen como datos de salida, podemos cambiar las propiedades de la imagen usando procesamiento computacional de imágenes.

Ejemplos de procesos que podemos realizar con las imágenes son la compresión y el encriptado, con el fin de reducir el tamaño del archivo sin que éste pierda calidad y mantenerlo seguro en caso de ser interceptado por un tercero que pudiera hacer mal uso de la información contenida en la imagen.

En la **compresión de imágenes**, se remueve la información redundante de la imagen, después el tamaño de la imagen será comprimido.

Los avances en la tecnología de video, incluyendo televisión de alta definición, están creando una gran demanda para el desarrollo de nuevos, mejores y más rápidos algoritmos de compresión de imágenes.

Los enfoques de compresión se pueden clasificar en **tres categorías**:

Directo (detecta redundancias basándose en un análisis directo de las muestras actuales de señales), extracción de parámetros (extrae las características particulares de parámetros de la señal y después codifica los parámetros extraídos) y métodos de transformación (SVD por ejemplo).

La **encriptación de imágenes** busca que éstas viajen de manera segura por las redes, esto se logra modificando la imagen al nivel de la información que representan, los datos que representan, modificándola, como es de esperarse, de una forma que solamente el que envía la imagen y quien deba recibirla sepan lo que se hizo para poder desencriptarla.

Ambos procesos se pueden realizar de diferentes maneras en las imágenes, en nuestro caso, se realizará mediante un método llamado ``descomposición en valores singulares'', también conocido como SVD.

6 COMPRESIÓN Y ENCRIPTACIÓN DE INFORMACIÓN UTILIZANDO DESCOMPOSICIÓN EN VALORES SINGULARES SVD

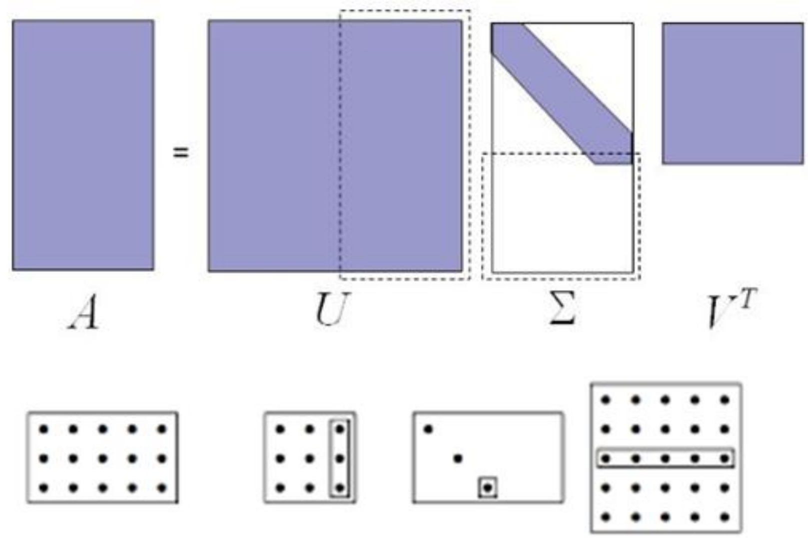
Las matrices describen las relaciones (y su intensidad) entre los componentes de varios tipos de sistemas, el sistema puede ser un puente, un circuito eléctrico, un reactor, etc.

En nuestro caso manipularemos las imágenes digitales como matrices de valores (que describen sus características a partir de sus pixeles), para comprimirlas y encriptarlas, haciendo uso de herramientas computacionales (Matlab) y conocimientos de álgebra lineal.

Partiendo de la imagen inicial como la Matriz original, la computadora es usada para **simplificar la matriz**, **y lograr que la mayoría de las entradas sean ceros**, un método con el que lo logra es conocido como **``descomposición en valores singulares´´** (funciona donde la descomposición LU o la eliminación gaussiana fallan, esto es, con matrices singulares o sistemas ``sin solución''):

La matriz original es representada como el producto de tres matrices:

$\boldsymbol{A = U\sum V^T}$, $\boldsymbol{A = USV^T}$ **ó A=UD**$\boldsymbol{V^T}$



Donde:

A(m x n) representa la matriz original.

S ó $\sum(\mathrm{n\ x\ n})$ la matriz en la que los únicos elementos que no son ceros se encuentran en **la diagonal principal**, éstos son los **valores singulares de la matriz original**, los cuales determinan **las frecuencias de oscilación del sistema/matriz original.**

Las otras dos matrices **U**(m x n) y **V**(n x n) **deben preservar las propiedades de la matriz original,** entonces son **matrices de columnas ortogonales** (U*V=0), cada una es el producto de matrices elementales que afectan solo a dos filas o columnas.

¨Si A es una matriz de m x n, los valores singulares de A son las raíces cuadradas de los valores propios que resulten de la multiplicación de $\boldsymbol{A^T A}$**.** ¨

El procedimiento para obtener los valores singulares de una matriz se ejemplifica a continuación, pero para poder realizarlo debes saber: **1 cómo trasponer una matriz, 2 cómo se realiza la multiplicación, suma y resta de matrices, 3 obtener el determinante de una matriz y 4 álgebra elemental para obtener los valores propios de una matriz, para luego sacarles raíz cuadrada y así obtener los valores singulares.**

1.- Trasposición: se refiere a simplemente intercambiar filas por columnas, de modo que, por ejemplo, la columna uno la escribas como la fila uno, respetando el orden de los datos y así con todas las filas/columnas.

$$A=\begin{pmatrix}2&0&1\\3&0&0\\5&1&1\end{pmatrix}\qquad B=\begin{pmatrix}1&0&1\\1&2&1\\1&1&0\end{pmatrix}$$

$$A+B=\begin{pmatrix}2+1&0+0&1+1\\3+1&0+2&0+1\\5+1&1+1&1+0\end{pmatrix}=\begin{pmatrix}3&0&2\\4&2&1\\6&2&1\end{pmatrix}$$

$$A-B=\begin{pmatrix}2-1&0-0&1-1\\3-1&0-2&0-1\\5-1&1-1&1-0\end{pmatrix}=\begin{pmatrix}1&0&0\\2&-2&-1\\4&0&1\end{pmatrix}$$

2.- La suma/resta de matrices es muy sencilla pues es ¨puntual¨, es decir, solo sumas los datos que están en las mismas coordenadas en las matrices (cabe aclarar que es necesario que sean matrices del mismo tamaño).

$$A\cdot B=\begin{pmatrix}2&0&1\\3&0&0\\5&1&1\end{pmatrix}\cdot\begin{pmatrix}1&0&1\\1&2&1\\1&1&0\end{pmatrix}=$$

$$=\begin{pmatrix}2\cdot1+0\cdot1+1\cdot1&2\cdot0+0\cdot2+1\cdot1&2\cdot1+0\cdot1+1\cdot0\\3\cdot1+0\cdot1+0\cdot1&3\cdot0+0\cdot2+0\cdot1&3\cdot1+0\cdot1+0\cdot0\\5\cdot1+1\cdot1+1\cdot1&5\cdot0+1\cdot2+1\cdot1&5\cdot1+1\cdot1+1\cdot0\end{pmatrix}=$$

$$=\begin{pmatrix}3&1&2\\3&0&3\\7&3&6\end{pmatrix}$$

En la **multiplicación** debe cumplirse que los tamaños de las matrices deben ser m x n y n x p respectivamente y qué originarán una matriz m x p como resultado. La forma específica de las matrices al multiplicar se debe a que la multiplicación se realiza de la forma ¨filas por columnas¨, como se muestra en la imagen.

3.- **El determinante** de una matriz se obtiene multiplicando los datos de la diagonal principal de la matriz y restándole la multiplicación de los datos de la diagonal opuesta.

4.- En el proceso, el **álgebra elemental** se utilizará para obtener el valor o valores de una incógnita denominada ¨lambda λ¨, que serán los valores propios de la matriz y se les sacará raíz cuadrada para obtener los valores singulares.

A continuación, un ejemplo simple de descomposición en valores singulares:

Las matrices necesarias son:

Matriz A original

0	1
1	0
1	1

A traspuesta (A^T)

0	1	1
1	0	1

Matriz Lambda

λ	0
0	λ

El procedimiento es:

1.- $A^T A$ (el orden de la multiplicación es importante)

$$\begin{matrix} 2 & 1 \\ 1 & 2 \end{matrix}$$

2.- Obtener Nueva matriz (NewA) restándole la matriz lambda.

$$\begin{matrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{matrix}$$

3.- Obtener el determinante de NewA.

Det(NewA)= $(2-\lambda)(2-\lambda)-(1)(1)$

$= \lambda^2 - 4\lambda + 3$

4.- Obtener el valor de λ.

$\lambda = 3 \quad \lambda = 1$

5.- Obtener los valores singulares.

<u>Valor singular 1= $\sqrt{3}$ Valor singular 2 = $\sqrt{1}$</u>

Como se puede notar, el procedimiento no es tan complicado, sin embargo, en matrices grandes puede volverse una tarea muy complicada.

<u>Estos procesos los realiza Matlab con solo usar un comando, así que no necesitarás dominar demasiado el tema.</u>

- Compresión:

Se trata de tomar una imagen y eliminar la información redundante, a fin de conseguir una imagen lo más nítida posible, reduciendo considerablemente su tamaño. Aquí la función del SVD es, a grandes rasgos, descomponer la imagen en las tres matrices mencionadas para luego recombinarlas y obtener una matriz como la original pero diferente en sus propiedades, tomando para esto un rango de valores singulares.

- Encriptado:

Es la adición de una ``llave'' al contenido u archivo, para que solo pueda ser visto por quien se desea. Aquí la función del SVD es aprovechar la generación de la matriz S para manipularla de una manera que solo el que envía la imagen y quien la recibe saben, para después simplemente aplicar la inversa del proceso que se haya hecho.

6.1 Compresión utilizando SVD

Para realizar la compresión de una imagen utilizando SVD se toma la matriz diagonal que contiene los valores singulares de la matriz(imagen) original, para modificarla, esto es, para retirar la información redundante, delimitando la matriz, esto tomando un rango de valores singulares de un determinado tamaño o ¨blocksize¨, podrás utilizar diferentes blocksize que desees, solo tendrás que observar que el tamaño del archivo generado (la imagen) se reduzca y que la imagen no pierda tanta calidad .

Una imagen a color tiene asociadas tres matrices, una matriz de rojo, una de verde y una de azul, y cada una de esas matrices tiene valores entre 0 y 255 que son los diferentes tonos de esos colores y la imagen es la superposición de ellos.

En este caso, al ser una imagen a color (RGB) se tendrán que extraer los datos de cada color y obtener de cada uno su matriz diagonal de valores singulares para modificarla.

Código:

```
Im=imread('ugto.JPG');
 I=double(Im);
 R=I(:,:,1);
 G=I(:,:,2);
 B=I(:,:,3);

 [U1,S1,V1]=svd(R);
 [U2,S2,V2]=svd(G);
 [U3,S3,V3]=svd(B);

blocksize=200;
```

var=Imread('titlulo.archivo') lee la imagen que se tenga para guardarla en una variable.

var=double(variable que contenga la imagen) convierte los datos a doble precisión (esto para no ser manejados como unit8 que es el formato de los datos de las imágenes).

En **R/G/B=I(:,:,1/2/3)** se usa el operador dos puntos para extraer de la imagen las matrices de los colores primarios que la conforman (rojo, verde y azul), y así poder modificar la imagen.

[U,S,V]=svd(matriz en la variable) sirve para realizar la descomposición en valores singulares.

En la variable **blocksize** se guarda el tamaño de bloque con el que se delimitará la matriz, siendo éste el valor fundamental que al ser modificado nos resultará en una imagen más o menos nítida y definirá el peso final del archivo.

S1(block:m,block:n) se usa para seleccionar un trozo de la matriz: B=A(i:j,n:m) crear una matriz B compuesta por los elementos en las filas [i,j] y las columnas [m,n].

VT1=V1' realiza el proceso de trasposición de la matriz.

Figure, imshow(variable/255) nos permite mostrar la imagen construida, además de otras al mismo tiempo.

```
S1(blocksize+1:943,blocksize+1:707)=0;
 S2(blocksize+1:943,blocksize+1:707)=0;
 S3(blocksize+1:943,blocksize+1:707)=0;

 VT1=V1';
  VT2=V2';
```

```
VT3=V3';


ImOutR=U1(:,1:blocksize)*S1(1:blocksize,1:blocksize)*VT1
(1:blocksize,:);

ImOutG=U2(:,1:blocksize)*S2(1:blocksize,1:blocksize)*VT2
(1:blocksize,:);

ImOutB=U3(:,1:blocksize)*S3(1:blocksize,1:blocksize)*VT3
(1:blocksize,:);

comprimida(:,:,1)= ImOutR;

comprimida(:,:,2)= ImOutG;

comprimida(:,:,3)= ImOutB;

 figure, imshow(I/255);
 figure, imshow(comprimida/255);
```

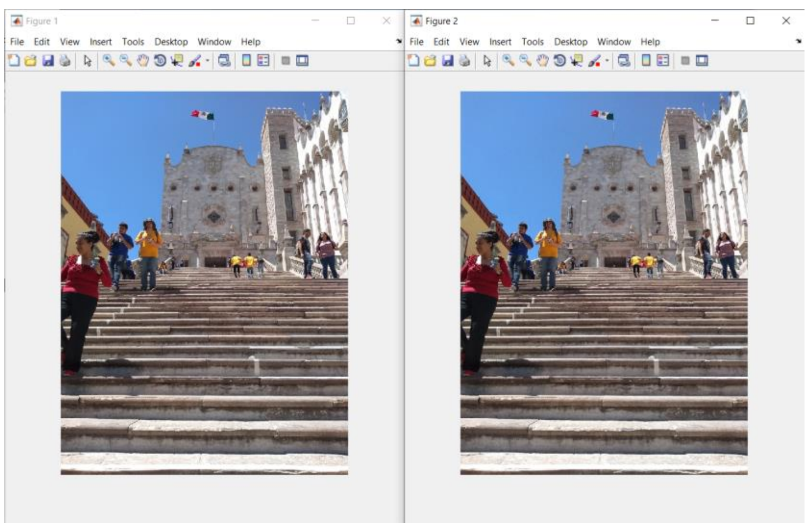


Figura1= Original

Imagen 204 Kb

Figura2= Comprimida

Imagen 69.1 Kb con blocksize de 200

6.2 ENCRIPTADO UTILIZANDO SVD

Para realizar el encriptado de la imagen utilizando SVD se toma de igual manera la matriz diagonal, ésta será modificada de una manera que solamente quien envía la imagen y quien se desea que la reciba conocen, un ejemplo sería sumándole una matriz o incluso 2 que contengan ciertos números, modificar el orden de los valores singulares de la matriz diagonal o cualquier otra cosa. En el ejemplo presentado en este tutorial se han utilizado dos matrices de números, es decir, 2 llaves, sumándole la llave 1 y restándole la llave 2 a las 3 matrices diagonales de valores singulares ``Sr'' generadas en cada uno de los "colores" (RGB).

Código:

```
[Ur1,Sr1,Vr1]=svd(ImOutR);
[Ur2,Sr2,Vr2]=svd(ImOutG);
[Ur3,Sr3,Vr3]=svd(ImOutB);


'///////////';
Id=size(Sr1);
 b=Id(2);
 d=Id(1);
 key1=(ones(d,b).*(100000)).*(2).*(3).*(5).*(8).*(397);
 key2=(eye(d,b).*(100000)).*(2).*(3).*(5).*(8).*(379);
 '///////////';


Sf1=Sr1+key1-key2;
Sf2=Sr2+key1-key2;
Sf3=Sr3+key1-key2;


 VrT1=Vr1';
  VrT2=Vr2';
   VrT3=Vr3';

 encricompR=Ur1*Sf1*VrT1;
 encricompG=Ur2*Sf2*VrT2;
 encricompB=Ur3*Sf3*VrT3;


 final(:,:,1)=encricompR;
 final(:,:,2)=encricompG;
 final(:,:,3)=encricompB;



  figure, imshow(comprimida/255);
  figure, imshow(final/255);
```

Servirá para no tener que estar modificando el tamaño de las matrices llave cada vez que cambiemos de imagen.

Var2=size (var1) obtiene el tamaño de la matriz para manipularlo

var=var2(1) sirve para extraer el tamaño o cantidad de filas.

var=var2(2) sirve para extraer el tamaño o cantidad de columnas.

Key es la variable en la que se guardará la matriz o arreglo que se usará como llave.

Key1 es una matriz de unos con las mismas dimensiones que las matrices diagonales S, que ha sido multiplicada por 100,000 y después por los primos 2, 3, 5, 8 y 397.

Key2 es una matriz diagonal de números 100,000 multiplicada por los mismos primos que la Key1 exceptuando el número final.

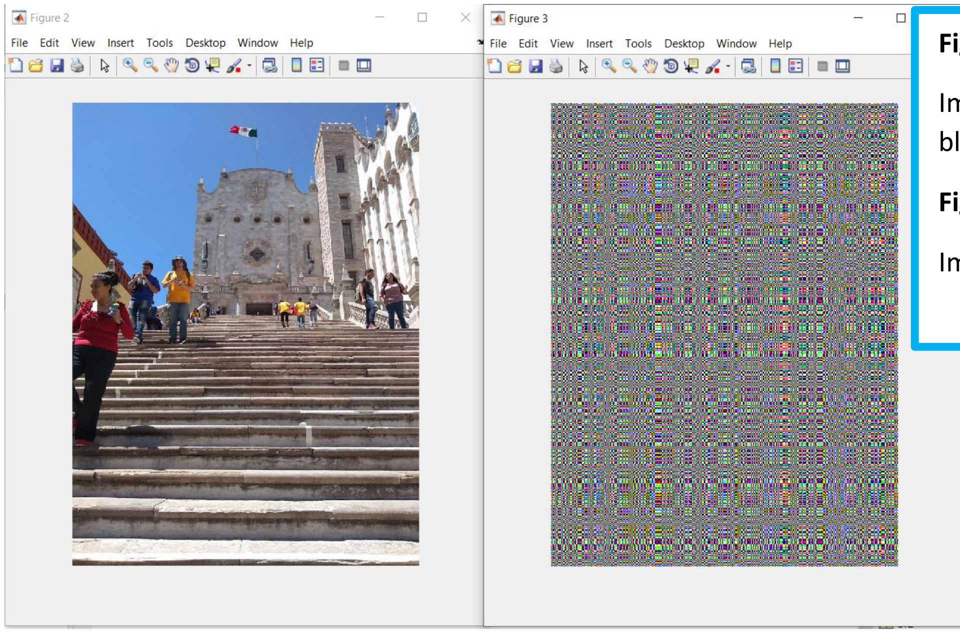


Figura1= Comprimida

Imagen 69.1 Kb con blocksize de 200

Figura1= Encriptada

Imagen 217 Kb

6.2.1 DESENCRIPTADO DE LA IMAGEN

Para desencriptar la imagen tan solo será necesario realizar el proceso inverso al utilizado para encriptarla, para ello deberemos contar con las matrices generadas en la descomposición SVD, guárdalas como un archivo.mat para luego cargarlas.

```
load('Sf1.mat')
load('Sf2.mat')
load('Sf3.mat')
load('Ur1.mat')
load('Ur2.mat')
load('Ur3.mat')
load('VrT1.mat')
load('VrT2.mat')
load('VrT3.mat')
```

Load('nombre.mat') es el comando que nos permite importar las matrices.

```
 encricompR=Ur1*Sf1*VrT1;
 encricompG=Ur2*Sf2*VrT2;
 encricompB=Ur3*Sf3*VrT3;

 encriptada(:,:,1)=encricompR;
 encriptada(:,:,2)=encricompG;
 encriptada(:,:,3)=encricompB;
```

```
'////////////';
Id=size(Sf1);
 b=Id(2);
 d=Id(1);
 key1=(ones(d,b).*(100000)).*(2).*(3).*(5).*(8).*(397);
 key2=(eye(d,b).*(100000)).*(2).*(3).*(5).*(8).*(379);
 '////////////';

S1=Sf1-key1+key2;
S2=Sf2-key1+key2;
S3=Sf3-key1+key2;

 ImOutR=Ur1*S1*VrT1;
 ImOutG=Ur2*S2*VrT2;
 ImOutB=Ur3*S3*VrT3;

desencriptada(:,:,1)= ImOutR;

desencriptada(:,:,2)= ImOutG;

desencriptada(:,:,3)= ImOutB;

figure, imshow(encriptada/255);
figure, imshow(desencriptada/255);
```

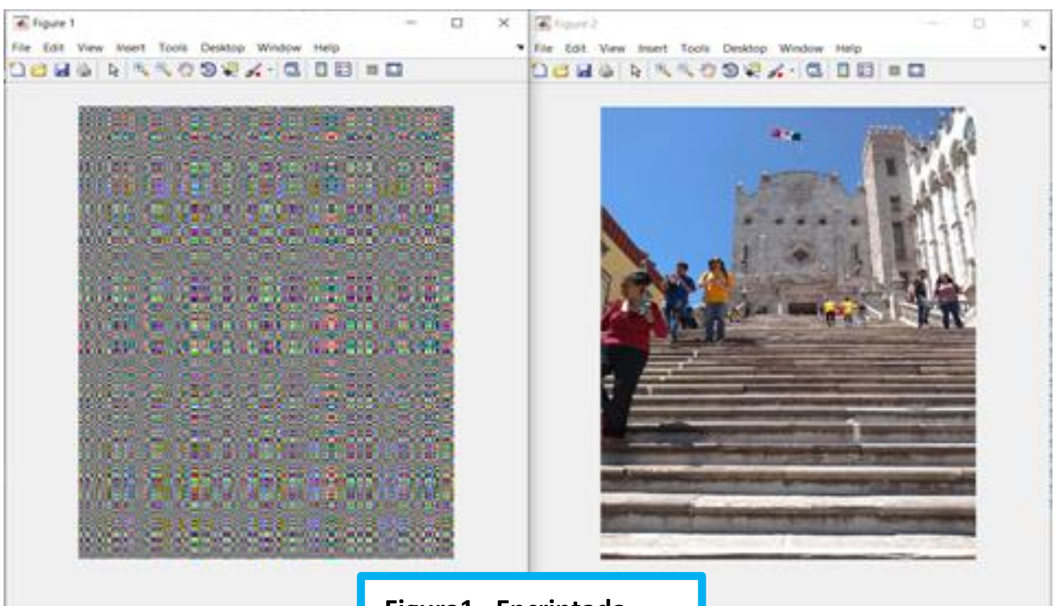


Figura1= Encriptada

Imagen 217 Kb

Figura2= Desencriptada

Imagen 69.1 Kb con blocksize de 200

7 FUENTES E INFORMACIÓN ADICIONAL

El presente documento se ha realizado mayormente a partir de conocimientos adquiridos durante el desarrollo del proyecto, del e-book ¨Fundamentos de Álgebra Lineal 7a. Ed. Ron Larson¨ así como de investigación en fuentes de internet, citando algunas de éstas a continuación.

https://es.mathworks.com

www.riunet.upv.es

www.informatica.uv.es

www.i-ciencias.com

www.jovenesenlaciencia.ugto.mx

www.juntadeandalucia.es

www.concurso.cnice.mec.es

www.lawebdelprogramador.com

www.elvisualista.com

www.yumpu.com

www.youtube.com

A sabiendas de que hoy en día muchos de los jóvenes preferimos el material audiovisual por sobre el texto tradicional, he de recomendar algunos videos que pueden encontrarse en Youtube y que podrían ser de ayuda a la hora de buscar comprender los temas aquí presentados, ya que actualmente con el crecimiento y mejora gradual de los contenidos generados por los denominados como ¨Edutubers¨ contamos con una novedosa forma de enseñar y aprender que cada día se vuelve más popular, listándolos a continuación.

° ¨1976 Matrix Singular Value Decomposition Film¨

° ¨MATRICES: de los gráficos de Fortnite a la física cuántica¨

° ¨Procesamiento de Imágenes 1.1 - Entorno, variables y memoria, carga y despliegue¨

Al navegar por los videos relacionados podrán encontrar mucha información que, o bien forma parte de cursos dedicados a realizar procesos específicos o buscan la mera divulgación de los temas de interés.

Espero que el presente documento le haya sido de ayuda en su formación y haya despertado su curiosidad sobre la ciencia y tecnología detrás del manejo de la información y su seguridad en los computadores, que al final y al cabo es lo principal.

¨Un ordenador es para mí la herramienta más sorprendente que hayamos ideado. Es el equivalente a una bicicleta para nuestras mentes¨. **Steve Jobs**