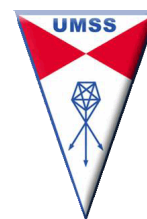


**UNIVERSIDAD MAYOR DE SAN SIMÓN**  
**FACULTAD DE CIENCIAS Y TECNOLOGÍA**  
**INGENIERÍA SISTEMAS - INFORMÁTICA**



# **TEMPLATE MATCHING**

## **INFORME - SOFTWARE**

Estudiantes: Adrián Huaylla Arturo (Ing. Sistemas)  
Maceda Choque Antony Brian (Ing. Sistemas)  
Orellana Ríos Alison (Ing. Informática)

Docente: Lic. García Pérez Carmen Rosa

Materia: Reconocimiento de Voz

Gestión: I - 2019

Cochabamba, Mayo de 2019

# TEMPLATE MATCHING

## SOFTWARE - DAMAS

### 1. Introducción

La coincidencia de plantillas es la técnica más simple y tiene la mayor precisión cuando se usa correctamente, pero también tiene limitaciones. Al igual que con cualquier método de reconocimiento de voz, el primer paso es que el usuario diga una palabra o frase en un micrófono. Para determinar el "significado" de esta entrada de voz, la computadora intenta hacer coincidir la entrada con una muestra de voz digitalizada o plantilla que tenga un significado conocido. El programa contiene la plantilla de entrada e intenta hacer coincidir esta plantilla con la entrada real mediante una declaración condicional simple. Este tipo de sistema se conoce como "dependiente del hablante".

### 2. Justificación

El análisis de las señales del habla humana por medio de un reconocedor de voz a través de la computadora, permite comprender la variación de los sonidos, tonalidades, según diferentes situaciones, emociones, ambiente y otros.

#### 2.1 Objetivo General

Implementar Reconocimiento de Voz en un software de juego llamado “Damas” elaborado con la herramienta Matlab, la cual también será usada para el desarrollo de la parte de Reconocimiento de la Voz, usando la técnica de Template Matching. De tal manera que el usuario “jugador” tan solo tenga que apretar un botón y dar la orden deseada.

##### 2.1.1 Objetivos específicos

- Implementar el juego clásico de “Damas” en Matlab.
- Desarrollar las funciones necesarias para aplicar la técnica de Template Matching de Reconocimiento de Voz en Matlab.
- Unión y adaptación de las funciones de RV en el juego de “Damas”.
- Creación de plantillas a usar grabando las voces dependientes del hablante.

### 3. Planteamiento y diseño

#### Planteamiento del Juego de Damas

Las damas es un juego de mesa para dos contrincantes. El juego consiste en mover las piezas en diagonal a través de los cuadros negros o blancos de un tablero de 64 cuadros con la intención de capturar (comer) las piezas del jugador contrario, pasando por encima de dichas piezas.

Las piezas normales se mueven 1 cuadrado verticalmente y si se llega al borde en la zona del enemigo la pieza se convertirá en reina, la cual se puede mover en diagonal, hacia cualquier lado, los cuadros que desee.

Cada jugador dispone de 12 piezas de un mismo color (uno blanco y otro negro) que al principio de la partida se colocan en las casillas negras de las tres filas más próximas a él. El objetivo del juego de damas es capturar las fichas del oponente o acorralarlas para que los únicos movimientos que puedan realizar sean los que lleven a su captura.

Se juega por turnos alternos. Empieza a jugar quien tiene las fichas claras (blancas). En su turno cada jugador mueve una pieza propia.

Las piezas se mueven (cuando no comen) una posición adelante (nunca hacia atrás) en diagonal a la derecha o a la izquierda, a una posición adyacente vacía.

Una partida de damas finaliza cuando estamos en una de estas 2 situaciones:

Pierde quien se queda sin piezas sobre el tablero.

Pierde si cuando llega el turno de un jugador no puede mover, puesto que todas las piezas que le restan en juego están bloqueadas.

La partida también puede terminar en tablas (empate) si ambos jugadores quedan con un número muy reducido de piezas, tal que por muchos movimientos que se hagan no se resolvería la partida. La dama/reina siempre tiene prioridad para comer antes que cualquiera otra ficha. También la dama solo se mueve un cuadro tras cada captura. Una pieza normal puede capturar a la dama.

### **Diseño**

El diseño del juego de Damas elaborado en Matlab consiste en que existen 2 jugadores, donde cada uno juega una pieza por turno, ambos poseen 12 piezas y el campo de movimiento es un tablero de 8x8 donde aparte se añaden 2 columnas y filas extra para el control de movimiento de las piezas. Los movimientos se realizan no solo mediante eventos del mouse al hacer click sobre el tablero, sino al presionar un botón y dar la orden mediante comando de voz para seleccionar la ficha a mover.

En el juego de damas implementado, las filas y columnas del tablero no serán contadas, sino que lo que tomaremos en cuenta para seleccionar piezas son las propias casillas del tablero siendo en total 32, esto con tal de poder seleccionar la pieza que deseamos sin menor complicación ya que la técnica empleada es template matching, si usáramos la nomenclatura normal, ocurrirían varios enredos a la hora de recuperar la señal de voz más parecida.

## **4. Implementación**

Para la implementación del juego de Damas, se han dividido las funcionalidades según:

### **Interfaz gráfica**

La pantalla y gráficas se han realizado mediante la herramienta de MatLab, donde se hace uso de las GUI de MatLab, (también conocidas como interfaces gráficas de usuario o interfaces de usuario), las cuales permiten un control sencillo de las aplicaciones de software. Por lo general, la GUI incluye controles tales como menús, barras de herramientas, botones y controles deslizantes.

Se definen los colores de las fichas, en este caso, rojo y negro; y el tablero que tendrá 64 casillas, de las cuales sólo serán utilizadas 32 para los respectivos movimientos posibles.



Se crean las cajas que contendrán a las fichas, cada caja tiene su nombre respectivo, y se definen funciones para controlar cada una:

```
ha(11) = handles.box11;
ha(13) = handles.box13;
ha(15) = handles.box15;
ha(17) = handles.box17;
ha(31) = handles.box31;
ha(33) = handles.box33;
```

Se definen posiciones de inicio y de fin en fila y columna para verificar los posibles movimientos:

```
handles.ipx=0;
handles.ipy=0;
handles.fpx=0;
handles.fpy=0;
handles.x=0;
handles.y=0;
```

Se tiene la definición de las fichas en sus posiciones, con sus imágenes respectivas:

```

set(ha(11), 'CData', handles.plr1mark);
set(ha(13), 'CData', handles.plr1mark);
set(ha(15), 'CData', handles.plr1mark);
set(ha(17), 'CData', handles.plr1mark);
set(ha(22), 'CData', handles.plr1mark);
set(ha(24), 'CData', handles.plr1mark);

```

Para hacer el “Matching”, se comparan las palabras obtenidas, que en este caso, representan las coordenadas de las posiciones de las cajas dentro del tablero.

Al haber obtenido las posiciones, se puede proceder a ejecutar su acción, es decir, el movimiento dentro del tablero con las coordenadas dadas:

```

if min_error < 10
    set(handles.axes2); %Establece el axes para graficar
    axes(handles.axes2); %Axes habilitado para graficar
    plot(transff_bd); grid on; %Grafica del espectro de la letra de la base de da

    set(handles.text1, 'string', strcat(nombre(1:end-4))); % Muestra la letra cc
    nombreAudio = strcat(nombre(1:end-4));

    if strcmp(nombreAudio, '1 alpha 1_1') || strcmp(nombreAudio, '1 alpha 1_2')
        box11_Callback(hObject, eventdata, handles);
    end
end

```

Las funciones definidas para cada caja, permiten controlar cada posición dentro del tablero, y así también se evitan confusiones con las posiciones:

```

function box11_Callback(hObject, eventdata, handles)
global ha
handles.x=1; handles.y=1;
handles=playerturn(ha, handles);
guidata(hObject, handles);

```

Igualmente, se muestra mediante una imagen de las fichas, el turno del jugador:



Para las fichas rojas normales, negras normales y las de tipo reinas y también para el tablero, se utilizan imágenes en formato .png.

## Reconocimiento de Voz

Se utilizan las funciones de:

1.audiorecorder(Fs, nBits, NumChannels).-

Crea y devuelve un objeto audiorecorder con una frecuencia de muestreo “Fs”, los bits por muestra “nBits” y un número de canales “NumChannels”.

2.recordblocking(recorderObj, length).-

Graba el audio desde un dispositivo de entrada en el objeto audiorecorder por un tiempo “length”.

3.getaudiodata(recorderObj).-

Guarda el objeto audiorecorder grabado en una matriz numérica.

4.filter(b,a,x).-

Usa una función de transferencia racional definida por el numerador “b” y el denominador “a”, para filtrar la matriz numérica “x”.

5 audiowrite(filename,filteredAudioData,Fs).-

Escribe una matriz de datos de audio “filteredAudioData” con una frecuencia de muestreo “Fs”, y lo guarda como un archivo con el nombre “filename” donde también se especifica el formato que tendrá. Para el proyecto usaremos el formato .WAV .

6.audioread(filename).-

Lee los datos del archivo “filename” y devuelve los datos muestreados como la matriz y la frecuencia de muestreo.

7.fft(X).-

Calcula la transformada discreta de Fourier usando un algoritmo de la transformada rápida de Fourier de “X”.

1. Si X es un vector, fft(X) devuelve la transformada de Fourier del vector.

2. Si X es una matriz, fft(X) trata las columnas de X como vectores y devuelve la transformada de Fourier de cada columna.

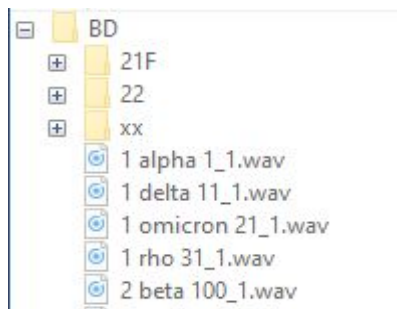
3. Si X es un array multidimensional, fft(X) trata los valores a lo largo de la primera dimensión del array cuyo tamaño no sea igual a 1 como vectores y devuelve la transformada de Fourier de cada vector.

Para la grabación de las voces en la base de datos y la entrada de voces posteriores, se ha considerado un tiempo de dos segundos, para evitar la lentitud de la corrida del juego.

### **Base de datos**

Las grabaciones obtenidas por MatLab, son almacenadas en formato .wav, en la carpeta definida como base de datos, de donde se podrán recuperar los datos y voces.

Cada que se crea una palabra o conjunto de palabras, se les aumenta el sufijo .wav.



## 7. Ejecución

Para ejecutar el programa, lo primero que haremos es pulsar el botón de corrida de MatLab. Luego de que cargue el programa, se verá una pantalla donde se encuentra el juego y los distintos botones que servirán para realizar acciones:

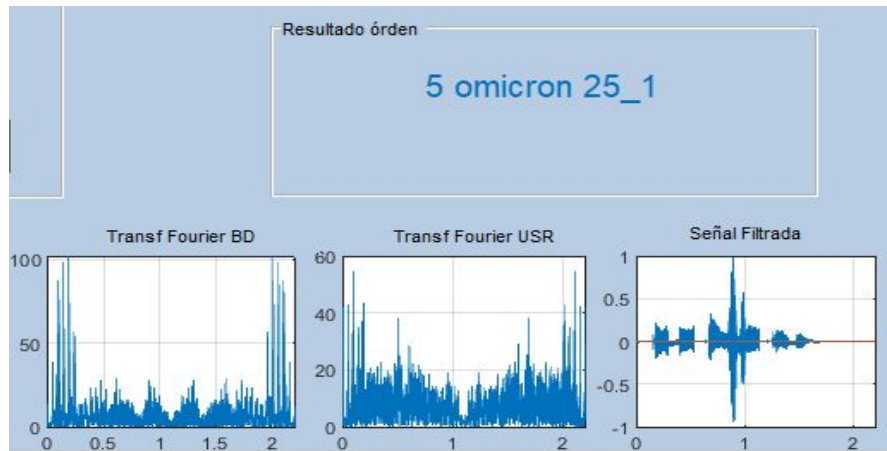


- Orden: Servirá para ingresar la voz
- Jugar: Servirá para realizar la ejecución del comando definido en la orden.
- Rendirse: En caso de que un jugador quiera abandonar la partida.
- Salir: permite salir del juego y aplicación.

La sección Almacenar Voz, permite grabar una nueva voz y almacenarla en la base de datos.

En la parte inferior, se tienen las gráficas de las voces, en la base de datos, las del usuario y el resultado, los cuales permiten ver las señales obtenidas.

Las señales de voz se grafican en la parte inferior:



Para jugar, primero se presiona el botón para realizar la orden, y se graba la primera voz con la coordenada de la posición de la cual parte la ficha; luego se presiona el botón para insertar la segunda voz que dará la coordenada de la posición de movida de la ficha respectiva.

## 8. Ventajas y Desventajas del Template Matching

### 8.1. Ventajas

- Las plantillas permiten hacer una comparacion rapida con las señales emitidas y la base de datos.
- Realiza un buen reconocimiento cuando las palabras son diferentes.

### 8.2. Desventajas

- Cuando se tienen palabras con tonalidades semejantes, no reconoce bien las palabras.

## 9. Conclusiones

Template Matching es bastante eficiente en el caso en que las palabras que tenga que reconocer sean muy diferentes, dado que la acústica y tonalidades generan las señales que van a ser comparadas.

No tiene un óptimo funcionamiento cuando se tienen palabras parecidas, dado que es imposible decir una palabra de la misma manera en diferentes ocasiones.