



Escuela
Politécnica
Superior

Reconstrucción del cuerpo humano mediante cámaras RGBD y técnicas de visión artificial y deep learning



Grado en Ingeniería Informática

Trabajo Fin de Grado

Autora:

Camelia Beltrá García

Tutores:

Jorge Azorín López

Andrés Fuster Guilló

Mayo 2022



Universitat d'Alacant
Universidad de Alicante

Reconstrucción del cuerpo humano mediante cámaras RGBD y técnicas de visión artificial y deep learning

Subtítulo del proyecto

Autora

Camelia Beltrá García

Tutores

Jorge Azorín López

TECNOLOGIA INFORMATICA Y COMPUTACION

Andrés Fuster Guilló



Grado en Ingeniería Informática



Escuela
Politécnica
Superior



Universitat d'Alacant
Universidad de Alicante

ALICANTE, Mayo 2022

Resumen

La visión artificial se encuentra en constante evolución en parte motivada por los avances de la Inteligencia Artificial y la optimización basada en Deep Learning.

El Deep Learning es un enfoque de aprendizaje automático no supervisado (es decir, son necesarios datos de entrenamientos, pero estos no requieren ser etiquetados) que se asemeja o se fundamenta en el funcionamiento del sistema neurológico humano.

La visión artificial es una disciplina científica que incluye métodos para adquirir, procesar y analizar imágenes del mundo real con el fin de producir información que pueda ser tratada por una máquina, la principal finalidad de esta tecnología es dotar a la máquina de "ojos" para ver que ocurre en el mundo real, y así poder tomar decisiones para automatizar cualquier proceso.

La razón principal que me ha llevado a la realización de este trabajo ha sido el uso de ambas tecnologías comentadas anteriormente, dado están cada vez más en auge además de la propia curiosidad sobre el aprendizaje y funcionamiento de estas.

Este trabajo tiene como finalidad el estudio de la visión artificial dentro de un marco dietético nutricional de la obesidad aprovechando las tecnologías para abordar este problema proponiendo una solución de la representación del cuerpo humano 3D con el uso único de una cámara y realizar unas comparativas sobre los modelos conseguidos para comprobar la veracidad de los datos obtenidos a partir de una red neuronal.

Agradecimientos¹

Este trabajo no habría sido posible sin el apoyo y el estímulo de mi colega y amigo, Doctor Rudolf Fliesning, bajo cuya supervisión escogí este tema y comencé la tesis. Sr. Quentin Travers, mi consejero en las etapas finales del trabajo, también ha sido generosamente servicial, y me ha ayudado de numerosos modos, incluyendo el resumen del contenido de los documentos que no estaban disponibles para mi examen, y en particular por permitirme leer, en cuanto estuvieron disponibles, las copias de los recientes extractos de los diarios de campaña del Vigilante Rupert Giles y la actual Cazadora la señorita Buffy Summers, que se encontraron con William the Bloody en 1998, y por facilitarme el pleno acceso a los diarios de anteriores Vigilantes relevantes a la carrera de William the Bloody.

También me gustaría agradecerle al Consejo la concesión de Wyndham-Pryce como Compañero, el cual me ha apoyado durante mis dos años de investigación, y la concesión de dos subvenciones de viajes, una para estudiar documentos en los Archivos de Vigilantes sellados en Munich, y otra para la investigación en campaña en Praga. Me gustaría agradecer a Sr. Travers, otra vez, por facilitarme la acreditación de seguridad para el trabajo en los Archivos de Munich, y al Doctor Fliesning por su apoyo colegial y ayuda en ambos viajes de investigación.

No puedo terminar sin agradecer a mi familia, en cuyo estímulo constante y amor he confiado a lo largo de mis años en la Academia. Estoy agradecida también a los ejemplos de mis difuntos hermano, Desmond Chalmers, Vigilante en Entrenamiento, y padre, Albert Chalmers, Vigilante. Su coraje resuelto y convicción siempre me inspirarán, y espero seguir, a mi propio y pequeño modo, la noble misión por la que dieron sus vidas.

Es a ellos a quien dedico este trabajo.

¹Por si alguien tiene curiosidad, este “simpático” agradecimiento está tomado de la “Tesis de Lydia Chalmers” basada en el universo del programa de televisión Buffy, la Cazadora de Vampiros.<http://www.buffy-cavampiros.com/Spiketesis/tesis.inicio.htm>

*A mi esposa Marganit, y a mis hijos Ella Rose y Daniel Adams,
sin los cuales habría podido acabar este libro dos años antes*²

²Dedicatoria de Joseph J. Roman en "An Introduction to Algebraic Topology"

*Si consigo ver más lejos
es porque he conseguido auparme
a hombros de gigantes*

Isaac Newton.

Índice general

1. Introducción	1
1.1. Motivación y contexto	1
1.2. Estado del arte	2
1.3. Objetivos	2
2. Marco Teórico (Con ejemplos de listas)	3
2.1. Listas	3
2.2. Listas de definición	4
3. Objetivos (Con ejemplos de tablas)	7
3.1. Tablas	7
3.2. Otros diseños de tablas	9
4. Metodología (Con ejemplos de figuras)	11
4.1. Inserción de figuras	11
5. Desarrollo (Con ejemplos de código)	13
5.1. Inserción de código	13
5.2. Usos y personalización	16
5.3. Importar archivos fuente	17
6. Resultados (Con ejemplos de gráficos)	19
6.1. Diagramas	19
6.2. Gráficas	20
6.2.1. Línea	20
6.2.2. Barras	21
6.2.3. Polar	22
6.3. Importados de MATLAB	23
6.4. Ejemplo avanzado	24
7. Conclusiones (Con ejemplos de matemáticas)	27
7.1. Matemáticas	27
Bibliografía	31
A. Anexo I	35
B. Páginas horizontales	37
C. Importar PDF	41

Índice de figuras

4.1. Ejemplo de subfiguras	12
4.2. Ejemplo de subfiguras vertical	12
6.1. Diagrama realizado en latex con Tikz.	20
6.2. Gráfica sencilla.	21
6.3. OP/S003	21
6.4. Gráfica barras.	22
6.5. Directividad normalizada del altavoz (0 dBV en el eje).	23
6.6. Ejemplo de gráfica obtenida con matlab2tikz.	24
6.7. Amplitud de la aceleración en el modo número 8.	24
6.8. Señal realizada con Tikz, sin imágenes.	25

Índice de tablas

3.1. Ejemplo de tabla.	7
3.2. Parámetros optativos de los entornos flotantes	8
3.3. Parámetros de los altavoces elegidos de la marca Beyma®.	9
3.4. Ejemplo 2	10
4.1. Esta es una tabla con múltiples imágenes. Útil cuando se deben mostrar varias juntas.	11

Índice de Códigos

5.1. ejemplo código C	13
5.2. ejemplo código C en color	14
5.3. ejemplo código PHP	14
5.4. ejemplo código PHP	14
5.5. ejemplo código Matlab en color	15
5.6. ejemplo código Matlab en blanco y negro	15
5.7. ejemplo código Python en color	16
5.8. ejemplo código Python en blanco y negro	16
5.9. Ejemplo de título abajo	17
5.10. Archivo C++ importado	18
5.11. Archivo Py importado	18
5.12. Archivo Matlab importado	18
6.1. Ejemplo de llamada a matlab2tikz	23

1. Introducción

1.1. Motivación y contexto

Mi interés por la visión artificial se desarrolló durante el tercer año de carrera, donde me presenté junto a la Universidad de Alicante a un proyecto llamado Vodafone Campus Lab[cita??], donde te proponen unos problemas a resolver y nosotros decidimos plantear una solución con tecnologías como la visión artificial, además esta propuesta requería de procesos como la adquisición del cuerpo 3D, reconocimiento de este, entre otras. Aquí fue cuando busqué propuestas de trabajo similares o sobre este tema, porque me quedé con la curiosidad de llevarlo a cabo y porque quería saber y aprender más sobre este tema, y conocí el proyecto de investigación Tech4Diet [cita]

El proyecto de investigación Tech4Diet cuenta con el apoyo de la Agencia Estatal de Investigación (AEI) y del Fondo Europeo de Desarrollo Regional (FEDER) con referencia "TIN2017-89069-R" perteneciente al programa Retos 2017 en el que su investigador jefe es Jorge Azorín. En este proyecto se busca facilitar el estudio de la evolución morfológica ocasionada por tratamientos de obesidad. Hoy en día, estos tratamientos son muy costosos pero a su vez muy necesarios, ya que los problemas de obesidad o sobrepeso pueden ocasionar enfermedades crónicas como la hipertensión, diabetes tipo II, cáncer. También pueden ocasionar enfermedades patológicas neurodegenerativas como el Alzheimer o demencias [Fuster-Guilló et al., 2020]

El sistema utilizado para esta finalidad, dispone de una red de cámaras RGB-D que obtienen un modelo 3D del cuerpo del paciente. El proceso de obtención de un modelo 3D se realiza en diferentes sesiones médicas, lo que permite una visualización real de la evolución del cuerpo del paciente. Para que el paciente pueda visualizar su progreso, no solo dispondrá de una aplicación de ordenador, sino que también podrá visualizar los diferentes modelos de su cuerpo mediante unas gafas de realidad virtual. La realidad virtual tiene como finalidad incrementar la adherencia del usuario al tratamiento. Además, podemos encontrar desarrollos tecnológicos como "Google Cardboard" que nos permiten convertir cualquier teléfono móvil en unas gafas de realidad virtual sin necesidad de realizar un gasto de dinero elevado. A parte de la visualización, sobre este modelo 3D obtenido con las cámaras se pueden realizar mediciones de diferentes partes del cuerpo a niveles de 1D, 2D y 3D.

Para que todo esto sea efectivo las mediciones han de corresponderse a las reales, por lo tanto el objetivo principal de este trabajo es comprobar y calcular si se pueden obtener buenas mediciones a partir de un modelo 3D generado con una sola imagen del cuerpo de la persona que será procesado en una red utilizando así el proyecto PIFu[cita] para ello.

1.2. Estado del arte

1.3. Objetivos

Este trabajo esta enmarcado dentro de un proyecto de investigación, y por lo tanto comparte el mismo objetivo general, proporcionar un sistema capaz de estudiar la evolución del cuerpo humano con el paso del tiempo mediante técnicas de visión 3D.

Para llegar al objetivo general es necesario disponer de una buena obtención del modelo 3D. Para ello, tenemos los siguientes objetivos:

- **Objetivo 1: Obtención del modelo 3D a partir de una imagen usando PIFu como red.**

Para ello se realizarán tareas como:

- Estudio de los diferentes parámetros necesarios para poder realizar dicho proceso.
- Generación de máscara de la imagen.
- Estudio de los resultados obtenidos.

- **Objetivo 2: Comparativa del modelo 3D obtenido con el modelo obtenido por el proyecto Tech4Diet. .**

Para este objetivo se tendrán que realizar las siguientes tareas específicas:

- Estudio sobre las diferentes formas de realizar el cálculo de las diferencias obtenidas.
 - Calculo de distancias usando Hausdorff Distance
 - Calculo de distancias usando
-

2. Marco Teórico (Con ejemplos de listas)

2.1. Listas

Hacer una lista es simple en \LaTeX . Para ello has de crear un entorno (así se llama) `itemize` con

```
\begin{itemize}  
...  
\end{itemize}
```

Y dentro de esa estructura, añadir cada elemento de la lista precedido de

```
\item primer ítem de lista  
\item segundo ítem de lista  
...  
\item ultimo ítem de lista
```

Es importante que revises este texto tal como aparece en la plantilla y relaciones el aspecto que tiene el PDF final con cómo está escrito el documento \LaTeX .

Aquí va una lista con subtérminos:

```
\begin{itemize}  
\item Ingeniería Informática.  
\item Ingeniería Sonido e Imagen en Telecomunicación.  
\item Ingeniería Multimedia.  
    \subitem Mención: Creación y ocio digital.  
    \subitem Mención: Gestión de Contenidos.  
\end{itemize}
```

El resultado es el siguiente:

- Ingeniería Informática.
- Ingeniería Sonido e Imagen en Telecomunicación.
- Ingeniería Multimedia.
 - Mención: Creación y ocio digital.
 - Mención: Gestión de Contenidos.

Aquí va una lista con subtérminos pero numerada:

```
\begin{enumerate}  
\item Ingeniería Informática.  
\item Ingeniería Sonido e Imagen en Telecomunicación.
```

```

\item Ingeniería Multimedia.
\begin{enumerate}
  \item Menció : Creaci n y ocio digital.
  \item Menció : Gesti n de Contenidos.
\end{enumerate}
\end{enumerate}

```

El resultado es el siguiente:

1. Ingeniería Informática.
2. Ingeniería Sonido e Imagen en Telecomunicación.
3. Ingeniería Multimedia.
 - a) Menció : Creaci n y ocio digital.
 - b) Menció : Gesti n de Contenidos.

2.2. Listas de definici n

Puedes realizar una lista de conceptos con su definici n del siguiente modo:

```

\begin{description} % Inicio de la lista
\item[MAPP XT:] Programa desarrollado por \textit{Meyer Sound} para el dise o y ajuste de sistemas
  ↪ formados por altavoces de su marca.
\begin{description} % Realiza una lista dentro de la lista
\item[Ventajas:]~
  El programa permite realizar m ltiples ajustes tal como se podr a realizar en la realidad con un
  ↪ procesador real.

  Permite analizar la fase recibida en cualquier punto y compararla con otras mediciones.

  Dispone de varios tipos de filtros, inversiones de fase, etc.
\item[Inconvenientes:]~
  No existe una lista global de los altavoces ubicados en el plano, por lo tanto solo se pueden editar
  ↪ seleccion ndolos sobre el plano.

  S lo permite dise ar en 2 dimensiones, principalmente sobre la vista lateral ya que los array de
  ↪ altavoces no permite voltearlos.
\end{description}
\end{description}

```

Y L^AT_EX genera lo siguiente:

MAPP XT: Programa desarrollado por *Meyer Sound* para el dise o y ajuste de sistemas formados por altavoces de su marca.

Ventajas: El programa permite realizar m ltiples ajustes tal como se podr a realizar en la realidad con un procesador real.

Permite analizar la fase recibida en cualquier punto y compararla con otras mediciones.

Dispone de varios tipos de filtros, inversiones de fase, etc.

Inconvenientes: No existe una lista global de los altavoces ubicados en el plano, por lo tanto solo se pueden editar seleccion ndolos sobre el plano.

Sólo permite diseñar en 2 dimensiones, principalmente sobre la vista lateral ya que los array de altavoces no permite voltearlos.

3. Objetivos (Con ejemplos de tablas)

3.1. Tablas

Ahora veremos otra estructura más: las tablas.

Aquí va una tabla¹ para que se vea cómo insertar una tabla simple dentro del documento.

```
\begin{table}[h]
\centering
\begin{tabular}{llll}
& columna A & columna B & columna C \\
\hline
fila 1 & fila 1, columna A & fila 1, columna B & fila 1, columna C \\
fila 2 & fila 2, columna A & fila 2, columna B & fila 2, columna C \\
fila 3 & fila 3, columna A & fila 3, columna B & fila 3, columna C \\
\hline
\end{tabular}
\caption{Ejemplo de tabla.}
\label{tabladeejemplo}
\end{table}
```

	columna A	columna B	columna C
fila 1	fila 1, columna A	fila 1, columna B	fila 1, columna C
fila 2	fila 2, columna A	fila 2, columna B	fila 2, columna C
fila 3	fila 3, columna A	fila 3, columna B	fila 3, columna C

Tabla 3.1: Ejemplo de tabla.

L^AT_EX usa un sistema de parámetros para “decorar” las tablas. Puedes consultar estos parámetros en la tabla 3.2 de la página 8. La tabla se ubicará donde, a juicio de L^AT_EX, menos moleste por lo que puede no aparecer necesariamente donde se ha insertado en el texto original.

Existe la posibilidad de forzar que las tablas, figuras u otros objetos aparezcan en la zona del texto que se desea aunque en ocasiones puede dejar grandes espacios en blanco. El comando a utilizar es:

```
\FloatBarrier
```

Que introducido justo después de una tabla, figura, etc (después del comando `\end{...}`) fuerza la aparición en el texto, empujando el contenido.

¹En <http://www.tablesgenerator.com/> se puede encontrar un generador On-Line de tablas para L^AT_EX

Parámetro	Significado
h	Situa el elemento flotante <i>preferentemente</i> (es decir, si es posible) en la situación exacta donde se incluye este
t	Sitúa el elemento en la parte de arriba de la página
b	Sitúa el elemento en la parte de abajo de la página
p	Sitúa el elemento en una página aparte dedicada sólo a elementos flotantes; en el caso del formato article , ésta se sitúa al final del documento, mientras que para el book es colocada al final de cada capítulo

Tabla 3.2: Parámetros optativos de los entornos flotantes

También es posible elegir el ancho de cada columna y la orientación del texto en cada una. Por ejemplo:

```
\begin{table}[ht]
\centering
\begin{tabular}{|C{2cm}|C{2cm}|C{2cm}|C{2cm}|} % 4 columnas de 2cm — texto centrado y con bordes
\hline
\multicolumn{4}{|c|}{\textbf{\begin{tabular}{c}{@{}c@{}}FUENTE: TRÁFICO RODADO\\
↪ HORARIO: TARDE\end{tabular}}}} \\ \hline
\textbf{dB(A)} & \textbf{Población expuesta tarde} & \textbf{\%} & \textbf{\scriptsize{↪
↪ CENTENAS}} \\ \hline
\textbf{\textgreater70} & 0 & 0,000 & 0 \\ \hline
\textbf{65 — 70} & 348,9 & 9,792 & 3 \\ \hline
\textbf{60 — 65} & 1594,7 & 44,757 & 16 \\ \hline
\textbf{55 — 60} & 322,1 & 9,040 & 3 \\ \hline
\textbf{50 — 55} & 0 & 0,000 & 0 \\ \hline
\textbf{\textgreater50} & 1297,3 & 36,410 & 13 \\ \hline
\textbf{\textbf{TOTAL}} & 3563 & 100 & 35 \\ \hline
\end{tabular}
\label{my-label}
\end{table}
```

L^AT_EX genera esto:

FUENTE: TRÁFICO RODADO HORARIO: TARDE			
dB(A)	Población expuesta tarde	%	CENTENAS
>70	0	0,000	0
65 - 70	348,9	9,792	3
60 - 65	1594,7	44,757	16
55 - 60	322,1	9,040	3
50 - 55	0	0,000	0
>50	1297,3	36,410	13
TOTAL	3563	100	35

Donde C{2cm} indica que la columna tiene el texto centrado y un ancho de 2 cm. Tambien

se puede utilizar $L\{\}$ o $R\{\}$ para poner el texto a la izquierda o derecha y definir un ancho concreto.

Páginas como <https://www.tablesgenerator.com/> ayudan a realizar tablas fácilmente, es lo más recomendado, ahorra mucho tiempo de trabajo y luego si falta algún detalle se puede retocar en el documento.

El formato estándar de las columnas es c , l o r , así lo genera la web mencionada antes, pero una vez generada puedes cambiar ese formato por el definido anteriormente para ajustar el ancho de las columnas, o mantenerlo así si el resultado ya es el deseado.

Para conocer más sobre las tablas puedes leer manuales como este: <https://latexlive.files.wordpress.com/2009/04/tablas.pdf> que contiene muchos ejemplos y explicaciones.

3.2. Otros diseños de tablas

Modelo	15LEX1600Nd	15P1000Fe V2
f_s (Hz)	41	45
R_e (ohm)	5.5	5.2
L_e (μH)	1600	1500
B_l (N/A)	25.7	27.4
M_{MS} (g)	175	157
C_{MS} ($\mu m/N$)	84	78
R_{MS} (kg/s)	6.8	7.6
d (cm)	33.5	33
V_{as} (dm ³)	91	80.7
Q_{TS}	0.36	0.30
Q_{MS}	6.6	5.9
Q_{ES}	0.38	0.31
Sens (dB @ 2.83V/1m)	96	98
η	1.7%	2.4%
S_d (cm ²)	880	855

Tabla 3.3: Parámetros de los altavoces elegidos de la marca Beyma®.

		140PU				50PU			
		Phase II		Phase I		Phase II		Phase I	
# BJet		≥ 4	2 or 3	≥ 4	2 or 3	≥ 4	2 or 3	≥ 4	2 or 3
# Bkg		123	76	12	7	84	35	7	3
Asimov	NM1	13	6	9	3	15	9	11	4
	NM2	6	2	4	1	7	3	5	1
	NM3	3	1	2	0	4	1	2	0
	STC	6	3	4	1	7	5	5	2

Tabla 3.4: Ejemplo 2

4. Metodología (Con ejemplos de figuras)

4.1. Inserción de figuras

Las figuras son un caso un poco especial ya que \LaTeX busca el mejor lugar para ponerlas, no siendo necesariamente el lugar donde está la referencia. Por ello es importante añadirle un “caption” y un “label” para poder hacer referencia a ellas en el párrafo correspondiente. Nosotros ponemos la referencia a la figura 4.1 que está en la página 11, justo aquí debajo, pero \LaTeX puede que la ubique en otro lugar. (observa el código \LaTeX de este párrafo para observar como se realizan las referencias. Estos detalles también se aplican a tablas y otros objetos).

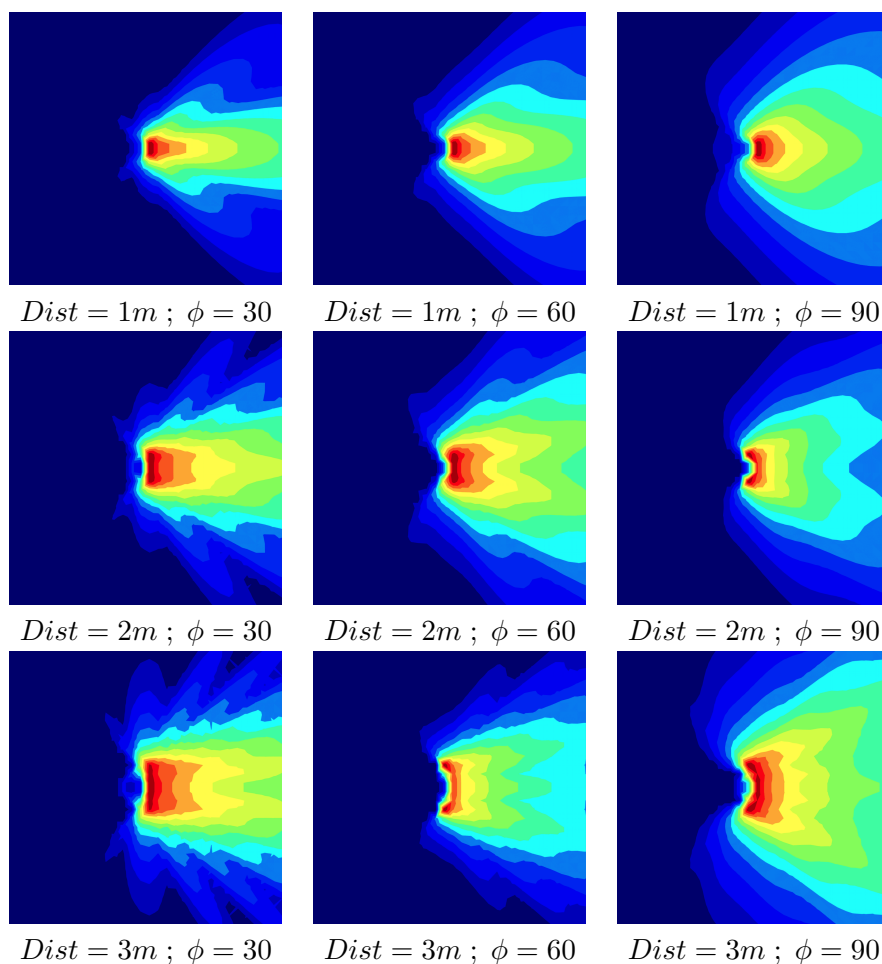
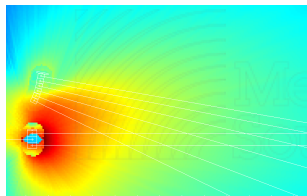


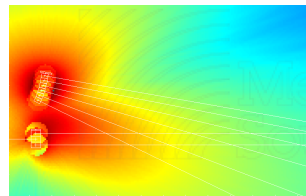
Tabla 4.1: Esta es una tabla con múltiples imágenes. Útil cuando se deben mostrar varias juntas.

Existe también la posibilidad de realizarlo sin tablas, con subfiguras:

```
\begin{figure}[h]
  \centering
  \begin{subfigure}[b]{0.4\textwidth} % Espacio horizontal ocupado por la subfigura
    \centering
    \includegraphics[width=4cm]{archivos/subs-sin} % Tamaño de la imagen
    \caption{Sin procesado.}
    \label{fig:gull}
  \end{subfigure}
  ~ % Añadir el espacio deseado, si se deja la línea en blanco la siguiente subfigura irá en una nueva línea
  \begin{subfigure}[b]{0.4\textwidth} % Espacio horizontal ocupado por la subfigura
    \centering
    \includegraphics[width=4cm]{archivos/subs-con} % Tamaño de la imagen
    \caption{Con procesado.}
    \label{fig:tiger}
  \end{subfigure}
  \caption{Ejemplo de subfiguras} \label{sistemass}
\end{figure}
```

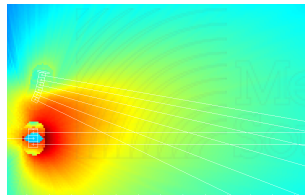


(a) Sin procesado.

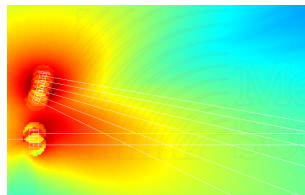


(b) Con procesado.

Figura 4.1: Ejemplo de subfiguras



(a) Sin procesado.



(b) Con procesado.

Figura 4.2: Ejemplo de subfiguras vertical

Si eliminas la línea '`\caption`' de las subfiguras, tendrás las imágenes sin la información individual, aunque sí con la principal. Y obviamente, si eliminas el de la figura no se mostrará ninguna información.

5. Desarrollo (Con ejemplos de código)

5.1. Inserción de código

A veces tendrás que insertar algún pedazo de código fuente para explicar algo relacionado con él. No sustituyas explicaciones con códigos enormes. Si pones algo de código en tu TFG que sea para demostrar algo o explicar alguna solución.

L^AT_EX te ayuda a escribir código de manera que su presentación tenga las marcas y tabulaciones propias de este tipo de texto. Para ello, debes poner el código que escribas DENTRO de un entorno que se llama “listings”. La plantilla ya tiene una serie de instrucciones para incluir el paquete “listings” y añadirle algunos modificadores por lo que no tienes que incluirlo tú. Simplemente, mete tu código en el entorno “lstlisting” y ya está. Puedes indicar el lenguaje en el que está escrito el código y así L^AT_EX lo mostrará mejor.

En el archivo *estiloscodigoprogramacion.tex* están definidos algunos lenguajes para mostrarlos con un diseño concreto, se pueden modificar para cambiar el coloreado del código, qué términos se ponen en negrita, etc. Si se quiere profundizar más en la función “listings” se puede consultar su manual en <http://osl.ugr.es/CTAN/macros/latex/contrib/listings/listings.pdf>, aunque hay mucha información en foros y blog’s que es más fácil de comprender.

Veamos un ejemplo en la figura 5.1:

```
\begin{lstlisting}[style=C, caption={ejemplo código C},label=C_code]
#include <stdio.h>
int main(int argc, char* argv[]) {
    puts("Hola mundo!");
}
\end{lstlisting}
```

El resultado será:

Código 5.1: ejemplo código C

```
1 #include <stdio.h>
2 // Comentario
3 int main(int argc, char* argv[]) {
4     puts("Hola mundo!");
5 }
```

Si lo quieres en color, está definido el estilo C-color en el archivo *estiloscodigoprogramacion.tex*, con algunos parámetros para mejorar la visualización:

```
\begin{lstlisting}[style=C-color, caption={ejemplo código C en color},label=C_code-color]
#include <stdio.h>
// Comentario
int main(int argc, char* argv[]) {
```

```
puts("Hola mundo!");
}
\end{lstlisting}
```

Código 5.2: ejemplo código C en color

```
1  #include <stdio.h>
2  // Comentario
3  int main(int argc, char* argv[]) {
4  puts("Hola mundo!");
5  }
```

Por supuesto, puedes mejorar esta presentación utilizando más modificadores. En la sección 5.2 se indican algunos detalles.

Otro ejemplo, ahora para mostrar código PHP, sería escribir en tu fichero \LaTeX lo siguiente:

```
\begin{lstlisting}[style=PHP, caption={ejemplo código PHP},label=PHP_code]
/*
Ejemplo de código en PHP para escribir tu primer programa en este lenguaje
Copia este código en tu ordenador y ejecútalo
*/
<html>
<head>
<title>Prueba de PHP</title>
</head>
<body>
<?php echo '<p>Hola Mundo</p>'; ?> //esto lo escribe TODO el mundo
</body>
</html>
\end{lstlisting}
```

y el resultado es el siguiente:

Código 5.3: ejemplo código PHP

```
100 /*
101 Ejemplo de código en PHP para escribir tu primer programa en este lenguaje. Copia este código en tu ↩
    ↩ ordenador y ejecútalo
102 */
103 <html>
104 <head>
105 <title>Prueba de PHP</title>
106 </head>
107 <body>
108 <?php echo '<p>Hola Mundo</p>'; ?> //esto lo escribe TODO el mundo
109 </body>
110 </html>
```

O también en color:

Código 5.4: ejemplo código PHP

```
1 /*
2 Ejemplo de código en PHP para escribir tu primer programa en este lenguaje. Copia este código en tu ↩
    ↩ ordenador y ejecútalo
3 */
4 <html>
5 <head>
6 <title>Prueba de PHP</title>
```

```

7 </head>
8 <body>
9   <?php echo '<p>Hola Mundo</p>'; ?> //esto lo escribe TODO el mundo
10 </body>
11 </html>

```

Observa cómo L^AT_EX ha puesto los comentarios en gris y ajustado el código para que se muestre más claro.

A continuación se muestran otros ejemplos:

Código 5.5: ejemplo código Matlab en color

```

1 %% Code sections are highlighted.
2 % System command are supported...
3 !touch testFile.txt
4 A = [1, 2, 3;... %... as is line continuation.
5     4, 5, 6];
6 fid = fopen('testFile.text', 'w');
7 for k=1:10
8     fprintf(fid, '%6.2f \n', k)
9 end
10 x=1; %% this is just a comment, not the start of a section
11 % Context-sensitive keywords get highlighted correctly...
12 p = properties(person); %(here, properties is a function)
13 x = linspace(0,1,101);
14 y = x(end:-1:1);
15 % ... even in nonsensical code.
16 ]end()()(((end while { end )end )))end (end
17 %{
18     block comments are supported
19 %} even
20 runaway block comments are

```

Código 5.6: ejemplo código Matlab en blanco y negro

```

1 %% Code sections are highlighted.
2 % System command are supported...
3 !touch testFile.txt
4 A = [1, 2, 3;... %... as is line continuation.
5     4, 5, 6];
6 fid = fopen('testFile.text', 'w');
7 for k=1:10
8     fprintf(fid, '%6.2f \n', k)
9 end
10 x=1; %% this is just a comment, not the start of a section
11 % Context-sensitive keywords get highlighted correctly...
12 p = properties(person); %(here, properties is a function)
13 x = linspace(0,1,101);
14 y = x(end:-1:1);
15 % ... even in nonsensical code.
16 ]end()()(((end while { end )end )))end (end
17 %{
18     block comments are supported
19 %} even
20 runaway block comments are

```

Código 5.7: ejemplo código Python en color

```

1 class Example (object):
2     def __init__ (self, account, password):
3         """e.g. account = 'bob@example.com/test'
4             password = 'bigbob'
5         """
6
7         reg = telepathy.client.ManagerRegistry()
8         reg.LoadManagers()
9
10        # get the gabble Connection Manager
11        self.cm = cm = reg.GetManager('gabble')
12
13        # get the parameters required to make a Jabber connection
14        # begin ex.basics.dbus.language-bindings.python.methods.call
15        cm[CONNECTION_MANAGER].RequestConnection('jabber',
16        {
17            'account': account,
18            'password': password,
19        },
20        reply_handler = self.request_connection_cb,
21        error_handler = self.error_cb)
22        # end ex.basics.dbus.language-bindings.python.methods.call

```

Código 5.8: ejemplo código Python en blanco y negro

```

1 class Example (object):
2     def __init__ (self, account, password):
3         """e.g. account = 'bob@example.com/test'
4             password = 'bigbob'
5         """
6
7         reg = telepathy.client.ManagerRegistry()
8         reg.LoadManagers()
9
10        # get the gabble Connection Manager
11        self.cm = cm = reg.GetManager('gabble')
12
13        # get the parameters required to make a Jabber connection
14        # begin ex.basics.dbus.language-bindings.python.methods.call
15        cm[CONNECTION_MANAGER].RequestConnection('jabber',
16        {
17            'account': account,
18            'password': password,
19        },
20        reply_handler = self.request_connection_cb,
21        error_handler = self.error_cb)
22        # end ex.basics.dbus.language-bindings.python.methods.call

```

5.2. Usos y personalización

El texto que acompaña al código puedes incluirlo o no, también puedes decidir si el texto va numerado o no. A continuación se muestra como:

```

% Con esta línea el código no tendrá título
\begin{lstlisting}[style=Python]
micodigo
\end{lstlisting}

```

```
1 micodigo
```

```
% Con esta línea el código tendrá el título abajo
\begin{lstlisting}[style=Python, caption={Ejemplo de título abajo},captionpos=b]
micodigo
\end{lstlisting}
```

```
1 micodigo
```

Código 5.9: Ejemplo de título abajo

```
% Con esta línea el código tendrá título no numerado
\begin{lstlisting}[style=Python, title={Ejemplo de título no numerado}]
micodigo
\end{lstlisting}
```

Ejemplo de título no numerado

```
1 micodigo
```

```
% Con esta línea el código no tendrá las líneas numeradas
\begin{lstlisting}[style=Python,numbers=none, title={Ejemplo de código sin número de líneas}]
micodigo
sin
número
de
líneas
\end{lstlisting}
```

Ejemplo de código sin número de líneas

```
micodigo
sin
número
de
líneas
```

5.3. Importar archivos fuente

Existe la posibilidad de importar un archivo de código en lugar de copiar su contenido y pegarlo en \LaTeX .

Para realizarlo debes escribir:

```
\lstinputlisting[style=C++-color,caption={Archivo C++ importado}]{archivos/ejemplos/holamundo.cpp}
```

Y se importará con el formato establecido entre los '[']':

Código 5.10: Archivo C++ importado

```

1 #include <stdio.h>
2 int main()
3 {
4     // printf() displays the string inside quotation
5     printf("Hello, World!");
6     return 0;
7 }

```

A continuación se muestran otros ejemplos

\lstinputlisting[style=Python-color,caption={Archivo Py importado},label=importado_py]{archivos/↩
↩ ejemplos/holamundo.py}

Código 5.11: Archivo Py importado

```

1 #-----↩
2 # helloworld.py
3 #-----↩
4
5 import stdio
6
7 # Write 'Hello, World' to standard output.
8 stdio.writeln('Hello, World')
9
10 #-----↩
11
12 # python helloworld.py
13 # Hello, World

```

\lstinputlisting[style=Matlab-color,caption={Archivo Matlab importado},label=importado_m]{archivos/↩
↩ ejemplos/holamundo.m}

Código 5.12: Archivo Matlab importado

```

1 function y = hello_world %#codegen
2
3 y = 'Hello World!';
4
5 end
6 % Copyright 2010 The MathWorks, Inc.

```

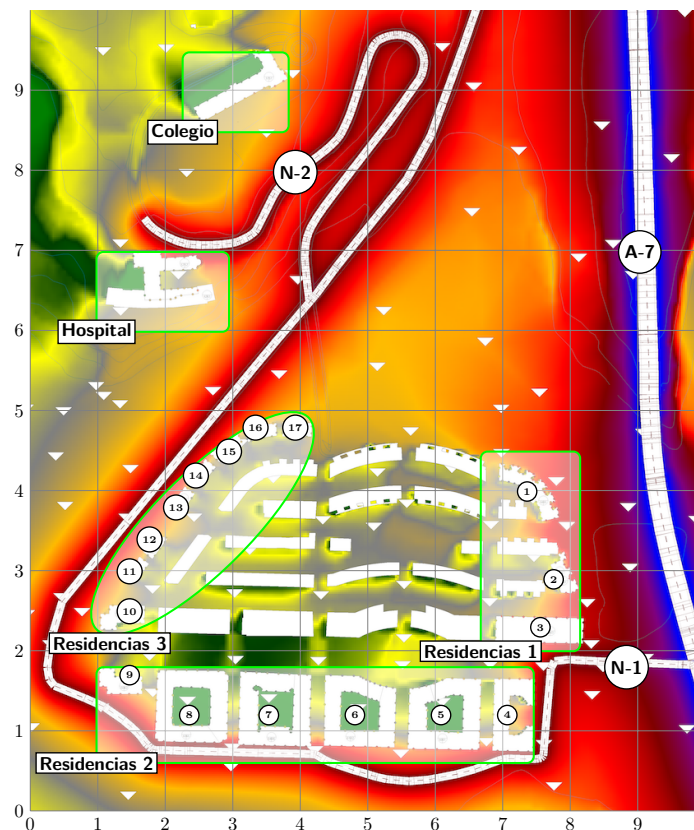
6. Resultados (Con ejemplos de gráficos)

6.1. Diagramas

Gracias al paquete *Tikz* se pueden incluir multitud de medios gráficos, diagramas, capas sobre imágenes, etc. Existen múltiples formas de realizarlo, para ello es recomendable consultar la guía de iniciación disponible aquí: <http://cremeronline.com/LaTeX/minimaltikz.pdf> y también el manual completo disponible aquí: <http://osl.ugr.es/CTAN/graphics/pgf/base/doc/pgfmanual.pdf>.

A continuación se muestran algunos ejemplos. Revisa el archivo .tex para ver cómo se utilizan.

Imagen a la que se le ha añadido cuadros y texto desde latex:



En muchas ocasiones es necesario realizar un diagrama de bloques, más abajo se muestra

un ejemplo de ello. En la red hay multitud de ejemplos que pueden ser fácilmente modificables para un fin concreto, como por ejemplo en esta web: <http://www.texample.net/tikz/examples/tag/block-diagrams/>.

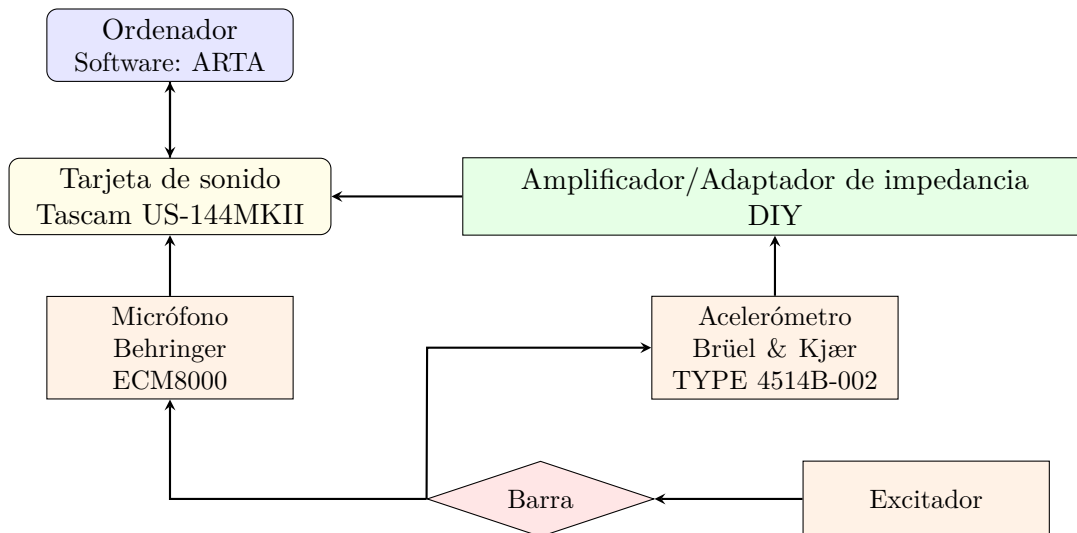


Figura 6.1: Diagrama realizado en latex con Tikz.

6.2. Gráficas

Existen múltiples formas de generar gráficas para latex. Hay disponibles herramientas como GeoGebra que dispone de la utilidad para exportar los gráficos en formato Tkiz. También funciones para Matlab que genera las gráficas que muestra habitualmente pero en código para Tkiz.

6.2.1. Línea

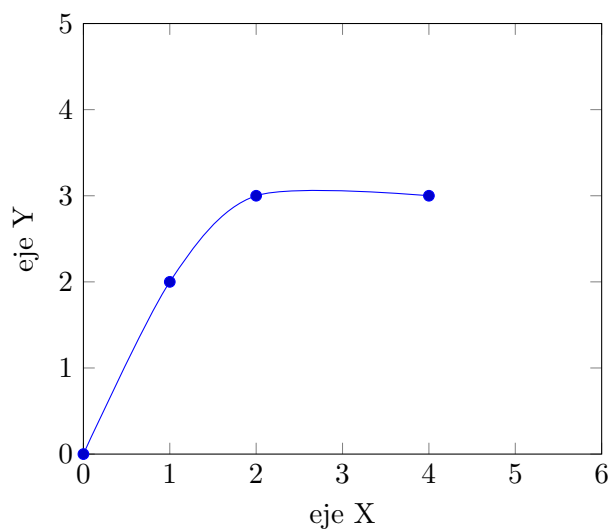
La forma más simple, aunque no sencilla cuando abarca muchos datos es la siguiente:

```

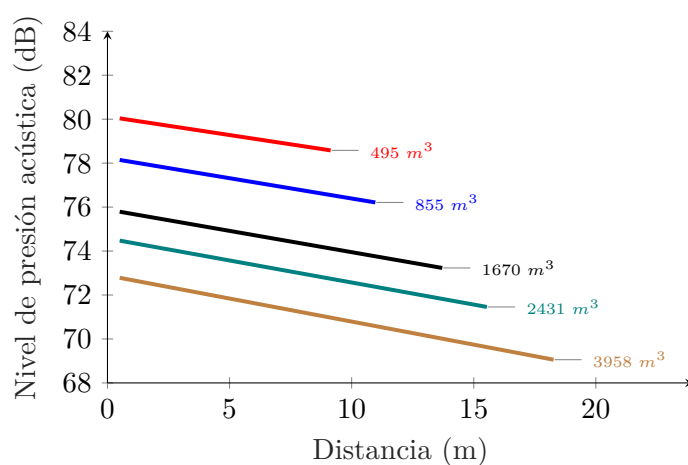
\begin{figure}[ht]
\centering
\begin{tikzpicture}
\begin{axis}
[ymin=0,ymax=5, % Límites del eje y
xmin=0,xmax=6, % Límites del eje x
ylabel= eje Y, % Nombre del eje y
xlabel= eje X] % Nombre del eje x
\addplot+[smooth] coordinates % Une los puntos curva suavizada
{(0,0) (1,2) (2,3 (4,3))}; % Puntos de la gráfica
\end{axis}
\end{tikzpicture}
\caption{Gráfica sencilla.}
\end{figure}

```

El resultado es el siguiente:

**Figura 6.2:** Gráfica sencilla.

Otro ejemplo, en este caso las líneas están calculadas directamente en LaTeX y después cada una tiene una anotación (el código se encuentra en el archivo `archivos/ejemplos/perjudicialesoptiacentro.tex`):

**Figura 6.3:** OP/S003

6.2.2. Barras

Otro ejemplo es la gráfica de barras:

```
\begin{figure}[ht]
\centering
\begin{tikzpicture}
\begin{axis}[
ybar=12pt,
```

```

ymin=0,ymax=150,
xtick=data,
enlarge x limits={abs=2cm},
symbolic x coords={rubio, moreno},
bar width = 20pt,
ylabel= número,
xlabel= color de pelo,
ytick align=outside,
ytick pos=left,
major x tick style = transparent,
legend style={at={(0.04,0.96)},anchor=north west, font=\footnotesize, legend cell align=left,},
]
\addplot[ybar,fill=blue, area legend] coordinates {
(rubio,20)
(moreno,100)};
\addplot[ybar,fill=purple, area legend] coordinates {
(rubio,110)
(moreno,105)};
\legend{Chicos, Chicas}
\end{axis}
\end{tikzpicture}
\caption{Gráfica barras.}
\end{figure}

```

El resultado es el siguiente:

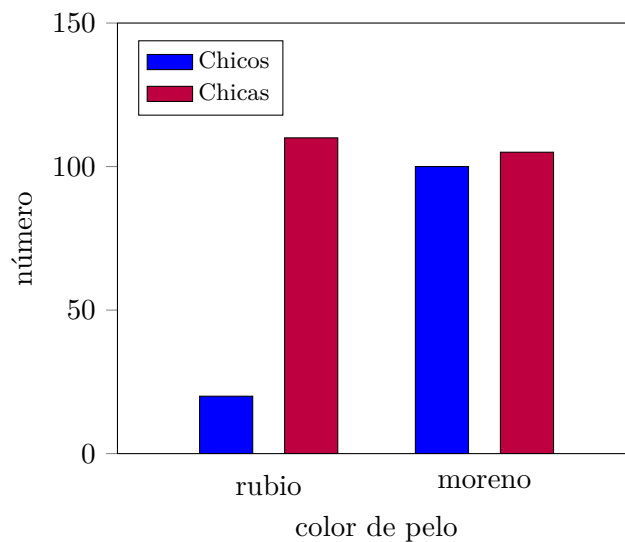


Figura 6.4: Gráfica barras.

6.2.3. Polar

Un ejemplo de gráfica polar semicircular (ver archivo `archivos/ejemplos/polarnorm.tex`):

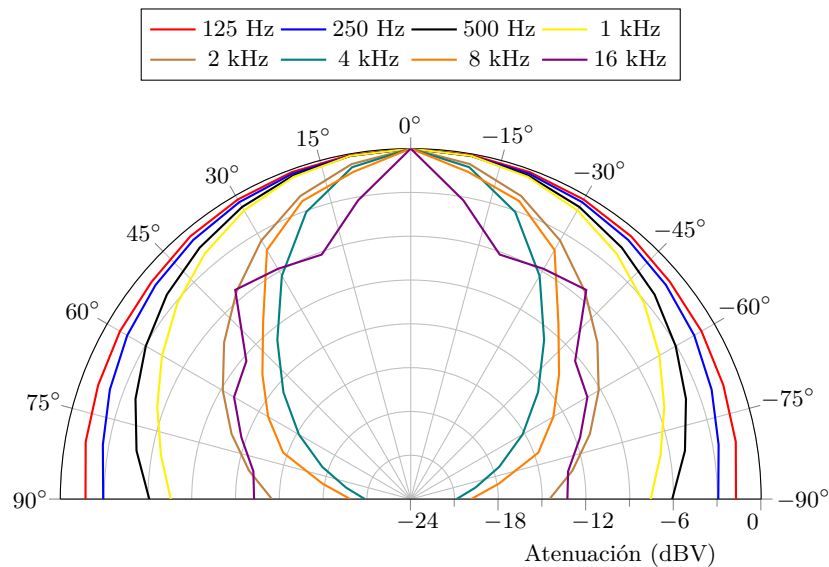


Figura 6.5: Directividad normalizada del altavoz (0 dBV en el eje).

6.3. Importados de MATLAB

Gracias a la herramienta *matlab2tikz* (<https://es.mathworks.com/matlabcentral/fileexchange/22022-matlab2tikz-matlab2tikz>) se pueden exportar las gráficas de cualquier tipo de Matlab a latex. Después de incluir los archivos de *matlab2tikz* se debe escribir una llamada después de crear la figura tal que:

Código 6.1: Ejemplo de llamada a *matlab2tikz*

```
1 fig = plot(x,y);
2 matlab2tikz('figurehandle',fig,'NombreArchivo.tex','height','5cm','width','13.5cm','strict',true,'↔
  ↳ showHiddenStrings',true,'showInfo',false)
```

Y para utilizar el archivo generado por la función en este documento:

```
\begin{figure}[ht]
\centering
{\scalefont{0.8}\input{archivos/ejemplos/ParedFina} }
\caption{Ejemplo de gráfica obtenida con matlab2tikz.}
\end{figure}
```

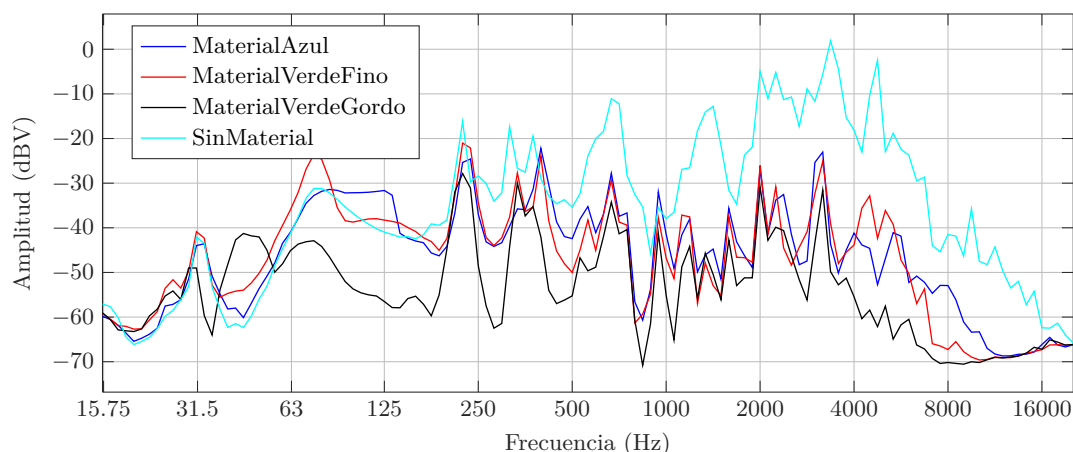


Figura 6.6: Ejemplo de gráfica obtenida con `matlab2tikz`.

Ejemplo de una gráfica 3D generada en Matlab y exportada por `matlab2tikz`:

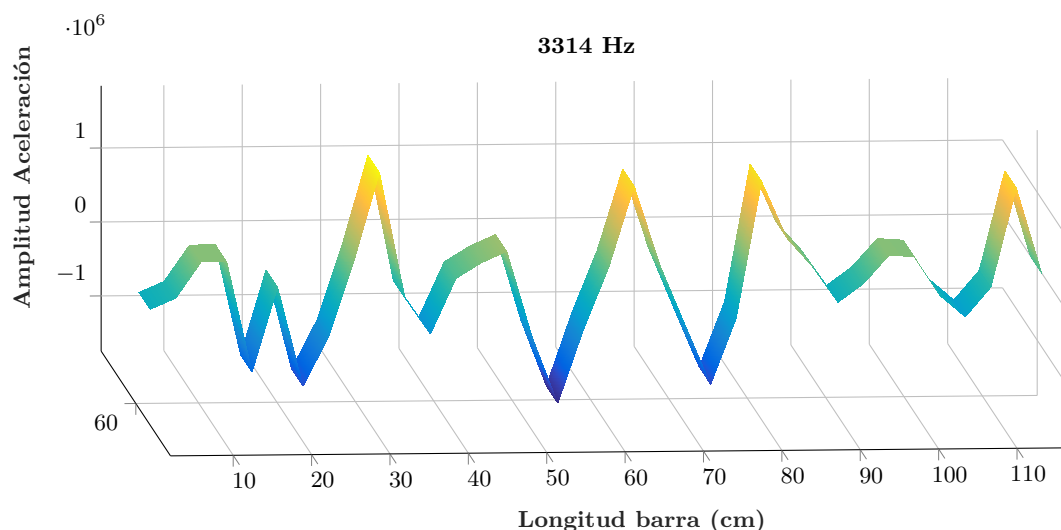


Figura 6.7: Amplitud de la aceleración en el modo número 8.

6.4. Ejemplo avanzado

El potencial del paquete *Tikz* es muy alto, se pueden realizar muchísimas cosas. En la red se facilitan muchos ejemplos para poder ver el funcionamiento y aprender. Existen hilos donde la gente publica sus mejores diseños de *Tikz* como en <https://tex.stackexchange.com/questions/158668/nice-scientific-pictures-show-off> o páginas donde facilitan muchas plantillas como <http://www.texample.net/tikz/examples/all/>.

Un ejemplo de lo que se puede llegar a conseguir es el siguiente:

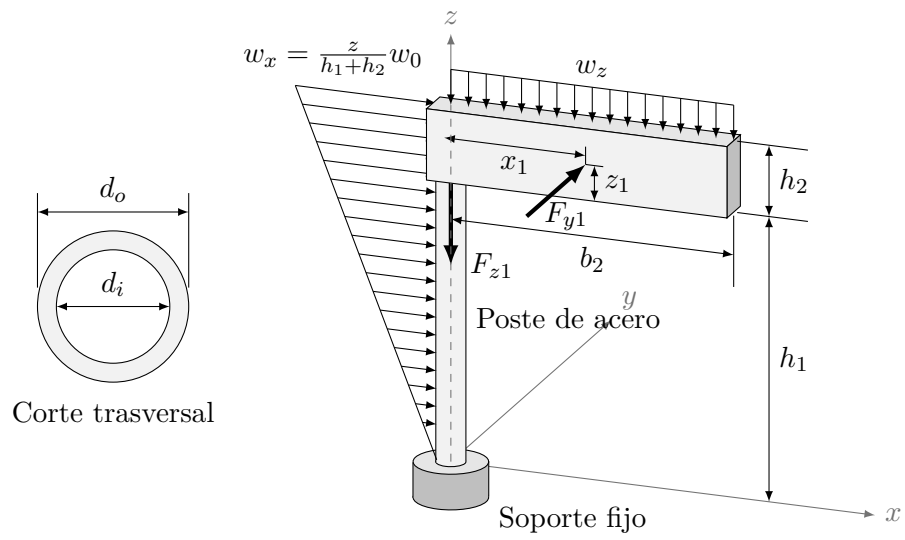


Figura 6.8: Señal realizada con Tikz, sin imágenes.

7. Conclusiones (Con ejemplos de matemáticas)

7.1. Matemáticas

En \LaTeX se pueden mostrar ecuaciones de varias formas, cada una de ellas para un fin concreto.

Antes de ver algunas de estas formas hay que conocer cómo se escriben fórmulas matemáticas en \LaTeX . Una fuente de información completa es la siguiente: <https://en.wikibooks.org/wiki/LaTeX/Mathematics>. También existen herramientas online que permiten realizar ecuaciones mediante interfaz gráfica como <http://www.hostmath.com/>, <https://www.mathcha.io/editor> o <https://www.latex4technics.com/>

Para mostrar una ecuación numerada se debe utilizar:

```
\begin{equation}
\nabla\times{\mathbf H}=\left[\frac{1}{r}\frac{\partial}{\partial r}(rH_\theta)-\frac{1}{r}\frac{\partial H_r}{\partial\theta}\right]\hat{\mathbf z}
\label{ecuacion}
\end{equation}
```

$$\nabla \times \mathbf{H} = \left[\frac{1}{r} \frac{\partial}{\partial r} (r H_\theta) - \frac{1}{r} \frac{\partial H_r}{\partial \theta} \right] \hat{\mathbf{z}} \quad (7.1)$$

Si es necesario agrupar varias ecuaciones en un mismo índice se puede escribir del siguiente modo:

```
\begin{subequations}
\begin{eqnarray}
{\mathbf E}&=&E_z(r,\theta)\hat{\mathbf z}\label{ecu1} \\
{\mathbf H}&=&H_r(r,\theta)\hat{\mathbf r}+H_\theta(r,\theta)\hat{\mathbf\theta}\label{ecu2}
\end{eqnarray}
\end{subequations}
% Se incluye '&' entre la igualdad para centrar las ecuaciones desde el '='.
```

$$\mathbf{E} = E_z(r, \theta) \hat{\mathbf{z}} \quad (7.2a)$$

$$\mathbf{H} = H_r(r, \theta) \hat{\mathbf{r}} + H_\theta(r, \theta) \hat{\boldsymbol{\theta}} \quad (7.2b)$$

Otras dos formas que son las habituales en muchos lugares para incluir ecuaciones son:

Ejemplo de fórmula en línea con el texto `\int_{a}^{b} f(x)dx = F(b) - F(a)`, esta ecuación quedará dentro \leftrightarrow del texto.

Esta otra, al utilizar dos `'$'`, se generará en una línea nueva `\int_{a}^{b} f(x)dx = F(b) - F(a)`

Ejemplo de fórmula en línea con el texto $\int_a^b f(x)dx = F(b) - F(a)$, esta ecuación quedará dentro del texto.

Esta otra, al utilizar dos `'$'`, se generará en una línea nueva

$$\int_a^b f(x)dx = F(b) - F(a)$$

También se puede añadir información adicional a una ecuación con la función *condiciones* creada para esta plantilla:

```
\begin{equation}
\underset{z=z_0}{\mathrm{Res}}(f(z))=\frac{1}{(m-1)!}\lim_{z\rightarrow z_0}\left[\frac{d^{m-1}}{dz^{m-1}}\left[(z-z_0)^m f(z)\right]\right]
\end{equation}

\begin{condiciones}[donde:]
% Excepto 'Descripción y valor' el resto no es necesario el símbolo $para texto matemático.
% Item & Relación & Descripción o valor
m & \rightarrow & Es la multiplicidad del polo $z_0$ \\
z_0 & \rightarrow & Es la parte que se iguala a 0 con el polo. \\
f(z) & \rightarrow & Es la función contenida en la integral.
\end{condiciones}
```

$$\mathrm{Res}(f(z)) = \frac{1}{(m-1)!} \lim_{z \rightarrow z_0} \left[\frac{d^{m-1}}{dz^{m-1}} [(z - z_0)^m f(z)] \right] \quad (7.3)$$

donde: $m \rightarrow$ Es la multiplicidad del polo z_0

$z_0 \rightarrow$ Es la parte que se iguala a 0 con el polo.

$f(z) \rightarrow$ Es la función contenida en la integral.

Si lo que deseas es una ecuación alineada a la izquierda o derecha puedes hacerlo con lo siguiente (el `'&'` simple es utilizado para alinear las ecuaciones desde ese punto, los iguales):

```
% Alineado a la izquierda al incluir al final el doble '&&'
\begin{flalign}
y_{h_1} &= \begin{bmatrix} 6\cos(\sqrt{6}x) \\ -\sqrt{6}\sin(\sqrt{6}x) \end{bmatrix} e^x && \\
y_{h_2} &= \begin{bmatrix} 6\sin(\sqrt{6}x) \\ \sqrt{6}\cos(\sqrt{6}x) \end{bmatrix} e^x && \\
\end{flalign}

% Alineado a la derecha al incluir al inicio el doble '&&'
\begin{flalign}
&& y_{h_1} = \begin{bmatrix} 6\cos(\sqrt{6}x) \\ -\sqrt{6}\sin(\sqrt{6}x) \end{bmatrix} e^x \\
&& y_{h_2} = \begin{bmatrix} 6\sin(\sqrt{6}x) \\ \sqrt{6}\cos(\sqrt{6}x) \end{bmatrix} e^x \\
\end{flalign}
```


$$y_{h_1} = \begin{bmatrix} 6 \cos(\sqrt{6}x) \\ -\sqrt{6} \sin(\sqrt{6}x) \end{bmatrix} e^x \quad (7.4)$$

$$y_{h_2} = \begin{bmatrix} 6 \sin(\sqrt{6}x) \\ \sqrt{6} \cos(\sqrt{6}x) \end{bmatrix} e^x \quad (7.5)$$

$$y_{h_1} = \begin{bmatrix} 6 \cos(\sqrt{6}x) \\ -\sqrt{6} \sin(\sqrt{6}x) \end{bmatrix} e^x \quad (7.6)$$

$$y_{h_2} = \begin{bmatrix} 6 \sin(\sqrt{6}x) \\ \sqrt{6} \cos(\sqrt{6}x) \end{bmatrix} e^x \quad (7.7)$$

Tanto con la función utilizada en (7.1,7.3), como en (7.2a,7.2b) y en las anteriores, si se les incluye un '*' después de 'equation', 'subequation' o 'flalign', se elimina la numeración de las ecuaciones pero manteniendo el resto de características.

Bibliografía

- Bradski, G., y Kaehler, A. (2013). Learning opencv: Computer vision in c++ with the opencv library. En (2nd ed., p. 373). O'Reilly Media, Inc.
- Brown, D. (2002, 12). Close-range camera calibration. En (Vol. 37).
- Camera parameters (internal, external). (2014). En K. Ikeuchi (Ed.), *Computer vision: A reference guide* (pp. 81–81). Boston, MA: Springer US. Descargado de https://doi.org/10.1007/978-0-387-31439-6_100019 doi: 10.1007/978-0-387-31439-6_100019
- Chen, S., Li, Y. F., Zhang, J., y Wang, W. (2008a). Active sensor planning – the state-of-the-art. En S. Chen, Y. F. Li, J. Zhang, y W. Wang (Eds.), *Active sensor planning for multiview vision tasks* (pp. 39–66). Berlin, Heidelberg: Springer Berlin Heidelberg. Descargado de https://doi.org/10.1007/978-3-540-77072-5_3 doi: 10.1007/978-3-540-77072-5_3
- Chen, S., Li, Y. F., Zhang, J., y Wang, W. (2008b). Active vision sensors. En S. Chen, Y. F. Li, J. Zhang, y W. Wang (Eds.), *Active sensor planning for multiview vision tasks* (pp. 11–38). Berlin, Heidelberg: Springer Berlin Heidelberg. Descargado de https://doi.org/10.1007/978-3-540-77072-5_2 doi: 10.1007/978-3-540-77072-5_2
- Derpanis, K. G. (2005). Overview of the ransac algorithm..
- Fischler, M. A., y Bolles, R. C. (1981, junio). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. En (Vol. 24, p. 381–395). New York, NY, USA: Association for Computing Machinery. Descargado de <https://doi.org/10.1145/358669.358692> doi: 10.1145/358669.358692
- Fuster-Guilló, A., Jorge Azorín-López, M. S.-C., Castillo-Zaragoza, J. M., Garcia-D’Urso, N., y Fisher, R. B. (2020). RGB-D based framework to Acquire, Visualize and Measure the Human Body for Dietetic Treatments..
- Ganapathi, V., Plagemann, C., Koller, D., y Thrun, S. (2010, 06). Real time motion capture using a single time-of-flight camera. En (p. 755–762). (<https://doi.org/10.1109/CVPR.2010.5540141>)
- Herrera C., D., Kannala, J., y Heikkilä, J. (2012). Joint depth and color camera calibration with distortion correction. En (Vol. 34, p. 2058–2064).
- Herrera C., D., Kannala, J., y Heikkilä, J. (2011). Accurate and practical calibration of a depth and color camera pair. En P. Real, D. Diaz-Pernil, H. Molina-Abril, A. Berciano, y W. Kropatsch (Eds.), *Computer analysis of images and patterns* (pp. 437–445). Berlin, Heidelberg: Springer Berlin Heidelberg.

- Kazmi, W., Foix, S., Alenyà, G., y Andersen, H. J. (2014). Indoor and outdoor depth imaging of leaves with time-of-flight and stereo vision sensors: Analysis and comparison. En (Vol. 88, p. 128 - 146). (<https://doi.org/10.1016/j.isprsjprs.2013.11.012>)
- Keselman, L., Woodfill, J. I., Grunnet-Jepsen, A., y Bhowmik, A. (2017). Intel(r) realsense(tm) stereoscopic depth cameras. En *2017 ieee conference on computer vision and pattern recognition workshops (cvprw)* (p. 1267-1276).
- Koch, B. (November 2010). Status and future of laser scanning, synthetic aperture radar and hyperspectral remote sensing data for forest biomass assessment. En (Vol. 65, p. 581-590). (<https://doi.org/10.1016/j.isprsjprs.2010.09.001>)
- Li, S., Wang, J., Liang, Z., y Su, L. (2016). Tree point clouds registration using an improved icp algorithm based on kd-tree. En *2016 ieee international geoscience and remote sensing symposium (igarss)* (p. 4545-4548).
- Mikhelson, I. V., Lee, P. G., Sahakian, A. V., Wu, Y., y Katsaggelos, A. K. (2014). Automatic, fast, online calibration between depth and color cameras. En (Vol. 25, p. 218 - 226). Descargado de <http://www.sciencedirect.com/science/article/pii/S1047320313000503> (Visual Understanding and Applications with RGB-D Cameras) doi: <https://doi.org/10.1016/j.jvcir.2013.03.010>
- O' Riordan, A., Newe, T., Toal, D., y Dooly, G. (2018, 12). Stereo vision sensing: Review of existing systems.. (<https://doi.org/10.1109/ICSensT.2018.8603605>)
- Procházková, J., y Martišek, D. (2018, 04). Notes on iterative closest point algorithm..
- Raposo, C., Barreto, J. P., y Nunes, U. (2013, 06). Fast and accurate calibration of a kinect sensor. En (p. 342-349). doi: 10.1109/3DV.2013.52
- Ruan, M., y Huber, D. (2014). Calibration of 3d sensors using a spherical target. En *2014 2nd international conference on 3d vision* (Vol. 1, p. 187-193).
- Rusinkiewicz, S., y Levoy, M. (2001). Efficient variants of the icp algorithm. En *Proceedings third international conference on 3-d digital imaging and modeling* (p. 145-152).
- Sankey, T. T. S. D. D. M. B. S. B. (June 2017). UAV lidar and hyperspectral fusion for forest monitoring in the southwestern USA.. (<https://doi.org/10.1016/j.rse.2017.04.007>)
- Sansoni G., D. F., Trebeschi M. (2009). State-of-the-art and applications of 3d imaging sensors in industry. En (pp. 568-601).
- Saval-Calvo, M., Azorin-Lopez, J., Fuster-Guillo, A., y Mora-Mora, H. (2015). -mar: Multiplane 3d marker based registration for depth-sensing cameras. En (Vol. 42, p. 9353 - 9365). Descargado de <http://www.sciencedirect.com/science/article/pii/S0957417415005503> doi: <https://doi.org/10.1016/j.eswa.2015.08.011>
-

- Schätz, M., Procházka, A., Kuchyňka, J., y Vyšata, O. (2020, 03). Sleep apnea detection with polysomnography and depth sensors. En (Vol. 20, p. 1360). doi: 10.3390/s20051360
- Smisek, J., Jancosek, M., y Pajdla, T. (2013, 01). 3d with kinect. En (p. 3-25). doi: 10.1007/978-1-4471-4640-7_1
- Su, P.-C., Shen, J., Xu, W., Cheung, S.-C. S., y Luo, Y. (2018). A fast and robust extrinsic calibration for rgb-d camera networks †. En (Vol. 18).
- Tech4Diet. (2019). Project tin2017-89069-r spanish state research agency (aei). 4d modelling and visualization of the human body to improve adherence to dietetic-nutritional intervention of obesity.. (<http://tech4d.dtic.ua.es/> (accessed Nov. 2019))
- Tech4Diet, M. . (2019). Project tin2017-89069-r spanish state research agency (aei). 4d modelling and visualization of the human body to improve adherence to dietetic-nutritional intervention of obesity.. (<http://tech4d.dtic.ua.es/media/> (accessed Nov. 2019))
- Tran, V., y Lin, H.-Y. (2018, 11). A structured light rgb-d camera system for accurate depth measurement. En (Vol. 2018, p. 1-7). (<https://doi.org/10.1155/2018/8659847>)
- Villena-Martínez, A. A.-L. J. S.-C. M. M.-P. J. G.-R. J. G.-G. A., V.; Fuster-Guilló. (2017). A Quantitative Comparison of Calibration Methods for RGB-D Sensors Using Different Technologies. Sensors..
- Wang, J., Shi, F., Zhang, J., y Liu, Y. (2008). A new calibration model of camera lens distortion. En (Vol. 41, p. 607 - 615). Descargado de <http://www.sciencedirect.com/science/article/pii/S0031320307003020> doi: <https://doi.org/10.1016/j.patcog.2007.06.012>
- Zhang, C., y Zhang, Z. (2011). Calibration between depth and color sensors for commodity depth cameras. En *2011 ieee international conference on multimedia and expo* (p. 1-6).
- Zhang, Z. (2016). Camera calibration: a personal retrospective. Machine Vision and Applications. En (p. 963-965). (<https://doi.org/10.1007/s00138-016-0809-z>)
- Zhengyou Zhang. (1999). Flexible camera calibration by viewing a plane from unknown orientations. En *Proceedings of the seventh ieee international conference on computer vision* (Vol. 1, p. 666-673 vol.1).
-

A. Anexo I

Aquí vendría el anexo I

B. Páginas horizontales

Aquí se muestra cómo incluir páginas en horizontal.
Esta página está en vertical

Esta página está en horizontal

Esta página también está en horizontal

Esta página está de nuevo en vertical

C. Importar PDF

A continuación se muestra una página importada de un PDF externo. Observar los comentarios en el código de este anexo para más información. También puedes leer el manual con todas las opciones en <http://osl.ugr.es/CTAN/macros/latex/contrib/pdfpages/pdfpages.pdf>.

Alicante 15 DE MARZO DE 2007

Expediente número

Referencia del petionario **AYUNTAMIENTO DE ALICANTE**
Departamento de Medio Ambiente
C/San Nicolás, nº 2, 4º
03001 ALICANTE
Contacto: Juan Luís Beresaluze

DOCUMENTO DE SÍNTESIS

***ELABORACIÓN DEL MAPA ACÚSTICO MUNICIPAL DE LA CIUDAD DE
ALICANTE***

Fecha de realización del estudio: MAYO 2005 – MARZO 2007