Universidad de Alcalá
Escuela Politécnica Superior

Grado en Ingeniería Informática

Trabajo Fin de Grado

Algoritmos de detección de objetos 3D basados en LiDAR: comparación entre técnicas PCL clásicas y Deep Learning

Autor: Javier de la Peña

Tutores: Luis Miguel Bergasa y Carlos Gómez Huélamo

2021

UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática

Trabajo Fin de Grado

Algoritmos de detección de objetos 3D basados en LiDAR: comparación entre técnicas PCL clásicas y Deep Learning

Autor: Javier de la Peña

Directores: Luis Miguel Bergasa y Carlos Gómez Huélamo

Tribunal:

Presidente: Felipe Espinosa Zapata

Vocal 1º: Fernando Naranjo Vega

Vocal 2º: Rafael Barea Navarro

Calificación: .....

Fecha: .....

A nuestros alumnos pasados, presentes y futuros...

"Empieza haciendo lo necesario, luego haz lo posible y de pronto empezarás a hacer lo imposible."
Francisco de Asís

Agradecimientos

A todos los que la presente vieren y entendieren.

Inicio de las Leyes Orgánicas. Juan Carlos I

Este trabajo es el fruto de muchas horas de trabajo, tanto de los autores últimos de los ficheros de la distribución como de todos los que en mayor o menor medida han participado en él a lo largo de su proceso de gestación.

Mención especial merece Manuel Ocaña, el autor de la primera versión de las plantillas de proyectos fin de carrera y tesis doctorales usadas en el Departamento de Electrónica de la Universidad de Alcalá, con contribuciones de Jesús Nuevo, Pedro Revenga, Fernando Herránz y Noelia Hernández.

En la versión actual, la mayor parte de las definiciones de estilos de partida proceden de la tesis doctoral de Roberto Barra-Chicote, con lo que gracias muy especiales para él.

También damos las gracias a Manuel Villaverde, David Casillas, Jesús Pablo Martínez, José Francisco Velasco Cerpa que nos han proporcionado secciones completas y ejemplos puntuales de sus proyectos fin de carrera.

Finalmente, hay incontables contribuyentes a esta plantilla, la mayoría encontrados gracias a la magia del buscador de Google. Hemos intentado referenciar los más importantes en los fuentes de la plantilla, aunque seguro que hemos omitido alguno. Desde aquí les damos las gracias a todos ellos por compartir su saber con el mundo.

Resumen

Este documento ha sido generado con una plantilla para memorias de trabajos fin de carrera, fin de máster, fin de grado y tesis doctorales. Está especialmente pensado para su uso en la Universidad de Alcalá, pero debería ser fácilmente extensible y adaptable a otros casos de uso. En su contenido se incluyen las instrucciones generales para usarlo, así como algunos ejemplos de elementos que pueden ser de utilidad. Si tenéis problemas, sugerencias o comentarios sobre el mismo, dirigidlas por favor a Javier de la Peña <j.pena@edu.uah.es>.

Palabras clave: Plantillas de trabajos fin de carrera/máster/grado y tesis doctorales, LaTeX, soporte de español e inglés, generación automática.

Abstract

This document has been generated with a template for Bsc and Msc Thesis (trabajos fin de carrera, fin de máster, fin de grado) and PhD. Thesis, specially thought for its use in Universidad de Alcalá, although it should be easily extended and adapted for other use cases. In its content we include general instructions of use, and some example of elements than can be useful. If you have problemas, suggestions or comments on the template, please forward them to Javier de la Peña <j.pena@edu.uah.es>.

Keywords: Bsc., Msc. and PhD. Thesis template, LaTeX, English/Spanish support, automatic generation.

Índice general

Índice de figuras

Índice de tablas

Índice de listados de código fuente

Índice de algoritmos

Capítulo 1

Introducción

1.1 Sistemas de conducción autónomos

1.2 Sistemas de percepción

1.2.1 Principales sensores para la percepción en vehículos autónomos

1.2.2 Sistemas de detección

1.2.3 Sistemas de seguimiento

1.2.4 Fusión sensorial

Bibliografía

[1] "Información sobre gnu/linux en wikipedia," http://es.wikipedia.org/wiki/GNU/Linux [Último acceso 1/noviembre/2013].

[2] "Página de la aplicación emacs," http://savannah.gnu.org/projects/emacs/ [Último acceso 1/noviembre/2013].

[3] "Página de la aplicación kdevelop," http://www.kdevelop.org [Último acceso 1/noviembre/2013].

[4] L. Lamport, *LaTeX: A Document Preparation System, 2nd edition.* Addison Wesley Professional, 1994.

[5] "Página de la aplicación octave," http://www.octave.org [Último acceso 1/noviembre/2013].

[6] "Página de la aplicación cvs," http://savannah.nongnu.org/projects/cvs/ [Último acceso 1/noviembre/2013].

[7] "Página de la aplicación gcc," http://savannah.gnu.org/projects/gcc/ [Último acceso 1/noviembre/2013].

[8] "Página de la aplicación make," http://savannah.gnu.org/projects/make/ [Último acceso 1/noviembre/2013].

Apéndice A

Manual de usuario

A.1 Introducción

Blah, blah, blah...

A.2 Manual

Pues eso.

A.3 Ejemplos de inclusión de fragmentos de código fuente

Para la inclusión de código fuente se utiliza el paquete `listings`, para el que se han definido algunos estilos de ejemplo que pueden verse en el fichero `config/preamble.tex` y que se usan a continuación.

Así se inserta código fuente, usando el estilo `CppExample` que hemos definido en el preamble, escribiendo el código directamente :

```
#include <stdio.h>

// Esto es una función de prueba
void funcionPrueba(int argumento)
{
  int prueba = 1;

  printf("Esto es una prueba [%d][%d]\n", argumento, prueba);

}
```

O bien insertando directamente código de un fichero externo, como en el ejemplo A.1, usando `\lstinputlisting` y cambiando el estilo a `Cbluebox` (además de usar el entorno `codefloat` para evitar pagebreaks, etc.).

Listado A.1: Ejemplo de código fuente con un `lstinputlisting` dentro de un `codefloat`

```
#include <stdio.h>

// Esto es una función de prueba
void funcionPrueba(int argumento)
{
        int prueba = 1;

  printf("Esto es una prueba [%d][%d]\n", argumento, prueba);

}
```

O por ejemplo en matlab, definiendo settings en lugar de usar estilos definidos:

```
%
% add_simple.m - Simple matlab script to run with condor
%
a = 9;
b = 10;

c = a+b;

fprintf(1, 'La suma de %d y %d es igual a %d\n', a, b, c);
```

O incluso como en el listado A.2, usando un layout más refinado (con los settings de http://www.rafalinux.com/?p=599 en un `lststyle Cnice`).

Listado A.2: Ejemplo de código fuente con estilo `Cnice`, de nuevo con un `lstinputlisting` dentro de un `codefloat`

```
1  #include <stdio.h>
2
3  #define LOOP_TIMES 5
4
5  int main(int argc, char* argv[])
6  {
7    int i;
8
9    for (i = 1; i < LOOP_TIMES; i++)
10     puts("Hola mundo!");
11 }
```

Y podemos reutilizar estilos cambiando algún parámetro, como podemos ver en el listado A.3, en el que hemos vuelto a usar el estilo `Cnice` eliminando la numeración.

Listado A.3: Ejemplo de código fuente con estilo `Cnice`, modificado para que no aparezca la numeración.

```
#include <stdio.h>

#define LOOP_TIMES 5

int main(int argc, char* argv[])
{
  int i;

  for (i = 1; i < LOOP_TIMES; i++)
    puts("Hola mundo!");
}
```

Ahora compila usando `gcc`:

```
$ gcc  -o hello hello.c
```

Y también podemos poner ejemplos de código *coloreado*, como se muestra en el A.4.

Listado A.4: Ejemplo con colores usando el estilo `Ccolor`

```
#include <stdio.h>

#define LOOP_TIMES 5

int main(int argc, char* argv[])
{
  int i;

  for (i = 1; i < LOOP_TIMES; i++)
    puts("Hola mundo!");
}
```

Finalmente aquí tenéis un ejemplo de código shell, usando el estilo `BashInputStyle`:

```
#!/bin/sh

HOSTS_ALL="gc000 gc001 gc002 gc003 gc004 gc005 gc006 gc007"

for h in $HOSTS_ALL
do
  echo "Running [$*] in $h..."
  echo -n "  "
  ssh root@$h $*
done
```

A.4 Ejemplos de inclusión de algoritmos

En la versión actual (abril de 2014), empezamos a usar el paquete `algorithm2e` para incluir algoritmos, y hay ajustes específicos y dependientes de este paquete tanto en `config/preamble.tex` como en `cover/extralistings.tex` (editadlos según vuestras necesidades).

Hay otras opciones disponibles (por ejemplo las descritas en `http://en.wikibooks.org/wiki/LaTeX/Algorithm`), y podemos abordarlas, pero por el momento nos quedamos con `algorithm2e`.

Incluimos dos ejemplos directamente del manual: uno sencillo en el algoritmo A.1, y otro un poco más complicado en el algoritmo A.2.

Data: this text
Result: how to write algorithm with LaTeX2e
initialization;
while *not at end of this document* **do**
  read current;
  if *understand* **then**
    go to next section;
    current section becomes this one;
  else
    go back to the beginning of current section;

Algoritmo A.1: How to write algorithms

Data: $G = (X, U)$ such that G^{tc} is an order.
Result: $G' = (X, V)$ with $V \subseteq U$ such that G'^{tc} is an interval order.
begin
  $V \longleftarrow U$
  $S \longleftarrow \emptyset$
  for $x \in X$ **do**
    $NbSuccInS(x) \longleftarrow 0$
    $NbPredInMin(x) \longleftarrow 0$
    $NbPredNotInMin(x) \longleftarrow |ImPred(x)|$
  for $x \in X$ **do**
    if $NbPredInMin(x) = 0$ **and** $NbPredNotInMin(x) = 0$ **then**
      $AppendToMin(x)$
1 **while** $S \neq \emptyset$ **do**
REM remove x from the list of T of maximal index
2 **while** $|S \cap ImSucc(x)| \neq |S|$ **do**
      for $y \in S - ImSucc(x)$ **do**
        { remove from V all the arcs zy : }
        for $z \in ImPred(y) \cap Min$ **do**
          remove the arc zy from V
          $NbSuccInS(z) \longleftarrow NbSuccInS(z) - 1$
          move z in T to the list preceding its present list
          {i.e. If $z \in T[k]$, move z from $T[k]$ to $T[k-1]$}
        $NbPredInMin(y) \longleftarrow 0$
        $NbPredNotInMin(y) \longleftarrow 0$
        $S \longleftarrow S - \{y\}$
        $AppendToMin(y)$
    $RemoveFromMin(x)$

Algoritmo A.2: IntervalRestriction

Apéndice B

Herramientas y recursos

Las herramientas necesarias para la elaboración del proyecto han sido:

- PC compatible
- Sistema operativo GNU/Linux [1]
- Entorno de desarrollo Emacs [2]
- Entorno de desarrollo KDevelop [3]
- Procesador de textos LaTeX[4]
- Lenguaje de procesamiento matemático Octave [5]
- Control de versiones CVS [6]
- Compilador C/C++ gcc [7]
- Gestor de compilaciones make [8]

Universidad de Alcalá
Escuela Politécnica Superior

ESCUELA POLITECNICA
SUPERIOR

Universidad
de Alcalá