

Propuestas Proyectos EL7008 Primavera 2015

Proyectos con Alta Prioridad

(mayor alineamiento con curso o mayor relevancia actual)

1. Clasificación de lugares usando bag of words (tutor: Pedro Orellana)

Implementar un sistema de clasificación de lugares usando bag of words y algún descriptor local (SIFT o algún otro a elección). Se debe utilizar la base de datos NYU V1 que dispone de 7 tipos de lugares y 2284 imágenes en RGB y profundidad.

2. Clasificación de imágenes usando deep learning (tutor: Pedro Orellana)

Utilizar el framework de C++ Caffe para clasificar imágenes (lugares, paisajes, vehículos etc..) bajo el paradigma de *deep learning*. Explorar entrenamiento y uso de redes pre-entrenadas.

3. Google Tango (tutor: Matías Mattamala)

El objetivo de este proyecto es aprender a utilizar un dispositivo Google Tango, el cual permite determinar odometría visual entre otras cosas, y documentar su utilización y características para futuros desarrollos. (<https://www.google.com/atap/project-tango/>)

4. Comparación de métodos de odometría/SLAM visual (tutor: Matías Mattamala)

Se busca comparar al menos dos métodos modernos de odometría visual o SLAM visual en robots humanoides y en escenarios desafiantes (movimientos bruscos), en términos de su error de traslación y rotación, recursos utilizados y tiempo de ejecución.

5. Segmentación de nubes de puntos obtenidas usando ORB-SLAM (tutor: Patricio Loncomilla)

En este proyecto se debe usar el algoritmo ORB-SLAM (de código abierto) para obtener una representación del ambiente basada en una nube de puntos rala, obtenida usando una cámara monocular. El objetivo de este proyecto es analizar la nube de puntos con el objetivo de generar clusters de puntos que puedan pertenecer a objetos. Para esto, se detectarán superficies planas en el mapa, y se segmentarán los objetos que estén sobre dichas superficies. También se puede explorar el uso de otras metodologías como LSD-SLAM. Este proyecto debe ser implementado en C o C++.

6. *Object proposals* para detección de objetos (tutor: Patricio Loncomilla)

El reconocimiento de objetos usando ventanas deslizantes requiere una gran capacidad de cómputo, lo cual impide que se pueda usar algoritmos precisos en tiempo real. El enfoque denominado *object proposal* consiste en encontrar regiones de la imagen que puedan corresponder a algún objeto con alta

probabilidad, lo cual permite focalizar la búsqueda sólo en las ventanas indicadas. De este modo, es posible usar algoritmos altamente precisos en tiempo real. Este proyecto consiste en la evaluación de algoritmos de *object proposal* implementados en C o C++.

7. Detección de personas en imágenes RGB y de profundidad (tutor: Patricio Loncomilla)

El objetivo de este proyecto es la detección en tiempo real de personas, usando tanto imágenes RGB como de profundidad. El uso simultáneo de ambas fuentes de información debería permitir la generación de detectores de personas más confiables que los obtenidos usando ambas fuentes de información por separado. Este proyecto debe ser implementado en C o C++.

8. Análisis Semántico de Imágenes de Camino (tutor: Fernando Bernuy)

El objetivo del proyecto es generar una descripción semántica del contenido de imágenes de camino utilizando métodos de segmentación semántica de imágenes e información de contexto. La descripción debe incluir información sobre el camino y su entorno para poder diferenciar entre distintos lugares observados en el camino basado únicamente en información de alto nivel. Para este proyecto se cuenta con una base de datos no etiquetada. El trabajo incluye las siguientes tareas: (i) seleccionar un método de segmentación semántica de imágenes, (ii) el etiquetado de imágenes a utilizar para entrenamiento del método, y (iii) diseño e implementación de la descripción semántica del contenido de las imágenes de camino.

Proyectos con Prioridad Intermedia

9. Software para hacer fotogrametría (tutor: Rodrigo Asenjo)

Existen varios software que hacen fotogrametría, los licenciados son muy caros y los open source no son muy buenos. Se podría partir de la base de un open source y mejorarlo. Será necesario construir un ground truth para validar el software. El software permitirá realizar mediciones de volúmenes, entregue una representación con mallas o nubes de punto y que sea capaz de exportar sus resultados a SolidWorks.

10. Estabilización de video (tutor: Pedro Orellana)

Implementar un sistema de estabilización de video estimando el movimiento entre 2 cuadros utilizando SIFT, flujo óptico u ORB-SLAM.

11. Sistema de realidad aumentada (tutor: Pedro Orellana)

Implementar un sistema para detectar, trackear y estimar pose de un patrón prediseñado para una aplicación de realidad aumentada. Dibujar alguna figura básica (cubo por ejemplo, con OpenGL para facilitar el trabajo). Debe ejecutarse en tiempo real usando una cámara RGB. Se sugiere usar SIFT para reconocer el patrón, aunque se puede usar otro tipo de algoritmos.

12. Tracking de múltiples objetos (tutor: Pedro Orellana)

Implementar un sistema que sea capaz de trackear y detectar múltiples objetos (objetos con formas/patrones/colores identificables) en tiempo real y que pueda lidiar con oclusiones entre los objetos en movimiento y de los objetos por si solos.

13. Reconocimiento de gestos de manos en tiempo real (tutor: Patricio Loncomilla)

El objetivo de esta tarea es el reconocimiento de gestos de manos en tiempo real, para lo cual debe ser programado en C o C++. La estrategia usada para reconocer las manos es libre, pudiendo basarse en detección de piel, ventanas deslizantes, detección de movimiento u otros métodos.

14. Reconocimiento de texto en ambientes reales (tutor: Patricio Loncomilla)

El objetivo de este proyecto es la detección y reconocimiento de texto (OCR) en ambientes reales, usando un código basado en C o C++ (sin usar Matlab). Se puede usar como base el paquete tesseract, o bien desarrollar un reconocedor propio.

15. Detección de puntos faciales en tiempo real (tutor: Patricio Loncomilla)

El objetivo de este proyecto es la detección de puntos faciales (puntos de los ojos, nariz, boca, etc). El proyecto debe estar implementado en C o C++. Se puede usar bibliotecas existentes para apoyar este trabajo.