

Visión por Computador

Proyectos Finales

Curso 2017

Condiciones de realización:

- 1.- Número de alumnos por proyecto: 2 como máximo
 - 2.- Antes del día 22 de diciembre todos los alumnos tendrán que haber elegido un posible proyecto. Los cambios de proyecto deberán ser aprobados por el profesor.
 - 3.- Una vez seleccionado el proyecto, se enviará un correo al profesor indicando nombre de los alumnos participantes y proyecto elegido.
 - 4.- La nota de la evaluación será por proyecto
 - 5.- Se valorará el enfoque, las técnicas usadas, calidad de la implementación, estudio experimental, resultados obtenidos, etc. También se valorarán todas las contribuciones innovadoras que hagan del proyecto una herramienta útil a otras personas con escasos o nulos conocimientos de visión por computador.
 - 6.- Pueden usarse todos los recursos de OpenCV excepto los módulos de Stitching, Face recognition y aquellos otros que resuelvan problemas completos de aplicación. Sí podrán usarse, si es necesario, librerías que implementen el manejo de estructuras de datos no definidas en OpenCV, tales como listas, colas y grafos. Cualquier otro software que se desee usar deberá ser consultado y autorizado previamente.
- Todos los proyectos deberán entregar una memoria y el código fuente. En la memoria debe contener como mínimo:
- a) Clara definición del problema a resolver y el enfoque elegido para ello.
Comentarios sobre las bondades del enfoque y de la implementación desarrollada.
 - b) Valoraciones sobre la implementación a partir de los resultados experimentales de evaluación obtenidos usando una base de datos de imágenes
 - c) Posibles propuestas de mejora de la técnica.
- 7.- Los proyectos deberán ser entregados antes del día 15 de Enero en la plataforma de la asignatura

A continuación se relacionan los temas que podrán ser objeto de proyecto. En clase o en las tutorías se irá contestando a todas las preguntas que haya sobre los mismos.

Temas objeto de posibles de proyecto: (valoración máxima de puntos al lado)

1. Implementar el algoritmo de Seam Carving (20 puntos)
(<http://www.win.tue.nl/~wstahw/edu/2IV00/seamcarving.pdf>
<https://code.google.com/p/seam-carving-gui/>)
2. Implementa el algoritmo de segmentación Intelligent Scissor (20 puntos)
(<http://courses.cs.washington.edu/courses/cse455/02wi/readings/mort-sigg95.pdf>)
3. Realizar una aplicación, bajo la plataforma OpenCV que implemente de forma exacta y eficiente el algoritmo de Canny. Todos y cada uno de los pasos del algoritmo deberán ser implementados. La aplicación tomará como entrada una imagen, los valores de la histéresis y el valor de sigma. La salida deberá ser el mapa de contornos. (20 puntos)
(ANULADO)
4. (NUEVO) Implementar y experimentar con el descriptor de formas ShapeContext (25 puntos) https://link.springer.com/chapter/10.1007/0-8176-4481-4_4 (pag 81) y <https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/shape/papers/mori-belongie-malik-pami05.pdf>
5. Implementar una herramienta que permita recortar objetos de una imagen y pegarlos en otra imagen sin que aparentemente se note en la imagen final que los objetos han sido pegados. (usando técnicas estudiadas en la asignatura) (20 puntos)
6. Implementar en todos sus detalles el algoritmo de estimación de una homografía usando el error de Sampson. Leer el libro de Hartley&Zisserman (pg 98)para aprender el algoritmo. No se permite el uso de ningún código externo. Todos los algoritmos usados deben ser de implementación propia salvo los algoritmos de optimización. (30 puntos)
7. Implementar la creación de panoramas lineales con proyección rectangular y mezcla de colores usando el algoritmo de Burt-Adelson (15 puntos)
(http://www.cs.princeton.edu/courses/archive/fall05/cos429/papers/burt_adelson.pdf)
8. Implementar la creación de panoramas lineales con proyección en superficies cilíndricas o esféricas y mezcla de colores usando el algoritmo de Burt-Adelson. (25 puntos)
9. Implementar la creación automática de panoramas (registrado geométrico y fotométrico) a partir de una bolsa de imágenes: a) selección de agrupaciones de imágenes relacionadas, b) creación de los panoramas definidos por cada agrupación (www.cs.bath.ac.uk/brown/papers/ijcv2007.pdf) (20 puntos)

10. Implementar una herramienta que haciendo uso de modelos de bolsas de palabras sea capaz de recuperar con la mayor eficacia posible las instancias de un objeto dado. Usar The Oxford Building Dataset (<http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/>) (30 puntos)
11. Implementar una herramienta que haciendo uso de modelos de bolsas de palabras sea capaz de recuperar las instancias de un objeto dado en los frames de una secuencia de vídeo. Añadir un experimento de validación que permita evaluar la calidad de los resultados obtenidos (Leer 14.3 de Szeliski y <http://www.robots.ox.ac.uk/~vgg/publications/papers/sivic03.pdf>) (30 puntos)
12. Implementar el algoritmo del descriptor HOG (Histograma de Gradientes) y aplicarlo a la detección de peatones en imágenes.
(<http://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf> ,
<http://pascal.inrialpes.fr/data/human/>) (30 puntos)
13. Diseñar y entrenar una arquitectura CNN que permita distinguir perros de gatos (usar <http://www.robots.ox.ac.uk/~vgg/data/pets/>
<http://vision.stanford.edu/aditya86/ImageNetDogs/>
https://archive.org/details/CAT_DATASET) (30 puntos)
14. GENERICO: Usar una de las bases de datos dadas en <http://www.robots.ox.ac.uk/~vgg/data/data-cats.html> para construir un problema de reconocimiento de objetos. Definir el problema en términos de identificar el modelo de entrenamiento a usar, los descriptores a extraer, el modelo de test y como se evalúan los resultados. Presentar los resultados alcanzados (30 puntos)

Cualquier otro proyecto puede ser aceptado, tras previa consulta y aprobación.

Se entenderá copia el uso de código extraído de internet o de cualquier otra fuente que no hay sido previamente autorizado