

Práctica de Técnicas de Inteligencia Artificial.

Primera práctica: Visión Artificial

Miguel Cazorla, Vicente Morell y Diego Viejo
Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante

Curso 2012 - 2013

1. Descripción de la práctica

La práctica de visión artificial para el curso 2012/13 consistirá en el desarrollo de un sistema de creación automática de imágenes panorámicas. Existen en la actualidad una gran variedad de programas que, a partir de varias fotografías que proporciona el usuario, obtienen una imagen panorámica en la que se mezclan todas las imágenes de entrada (<http://hugin.sourceforge.net>). Muchos de estos programas requieren la interacción del usuario para establecer las partes comunes que comparten las imágenes de entrada con el fin de obtener el panorama final. El objetivo de esta práctica consiste en reducir al mínimo, mediante el uso de técnicas de visión artificial, la intervención del usuario para crear dichos panoramas. Tomando como única información de entrada las imágenes, podremos crear un programa que encuentre automáticamente aquellas partes comunes en las imágenes y, a partir de esta información, elabore el panorama final, todo ello de manera automática.

Básicamente podemos resumir el proceso de creación de panoramas en dos pasos fundamentales (aunque los programas para la creación de panoramas realizan alguno más con el fin de ofrecer un resultado más “profesional”). El primero de estos pasos consiste en encontrar las partes comunes que comparten las imágenes de entrada (aquí es donde los programas comerciales suelen requerir más intervención del usuario). La idea consiste en encontrar un conjunto de puntos característicos compartidos por cada par de imágenes de entrada. Una vez encontrados, estos puntos se utilizan para calcular la transformación que mejor alinea las imágenes. En la figura 1 se puede observar una visión general del proceso.

En lugar de pedir al usuario que introduzca los puntos característicos que comparten las imágenes, el primer paso a resolver en la práctica consistirá en encontrar estos puntos automáticamente. Para ello existen multitud de metodologías. Nosotros vamos a apoyarnos en una de las funciones que tenemos implementada en JavaVis. La función FNitzberg encuentra puntos característicos (puntos esquina) en una imagen. No parece muy difícil asumir que si dos imágenes comparten una zona en común, los puntos característicos que encontrará FNitzberg en estas zonas

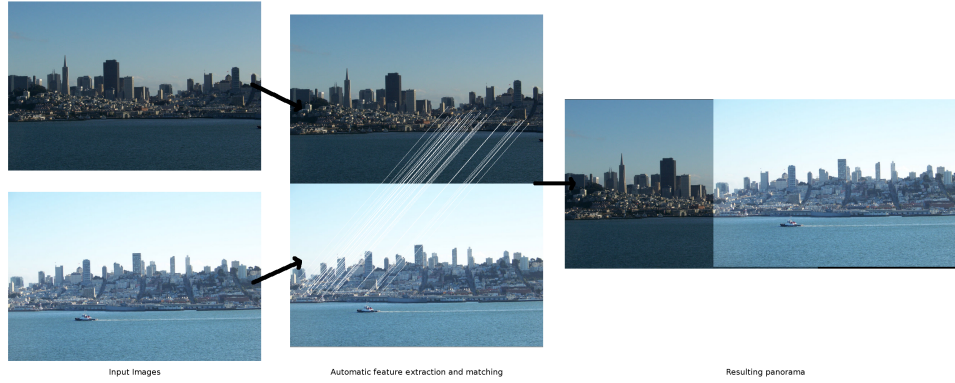


Figura 1: Proceso básico para la creación de un panorama.

comunes estarán situados en los mismos sitios. Nuestra tarea entonces será buscar cuales de los puntos característicos de cada imagen son realmente el mismo punto. El test más sencillo que podemos realizar consiste en comparar las zonas adyacentes a cada uno de los puntos. El método de correlación cruzada nos servirá para calcular un valor de 'parecido' entre dos zonas de las imágenes. Extrayendo una zona cuadrada (ventana) alrededor de los puntos característicos que estemos considerando, el valor de la correlación cruzada se calcula según la siguiente expresión:

$$CC = \frac{\sum_i (I_i^1 - m^1) * (I_i^2 - m^2)}{\sqrt{\sum_i (I_i^1 - m^1)^2} * \sqrt{\sum_i (I_i^2 - m^2)^2}}, \quad (1)$$

donde m^1 y m^2 son la media de las intensidades en niveles de gris (I) de las ventanas utilizadas. Esta expresión será máxima cuando las dos zonas sean iguales. De esta manera podemos buscar, para cada punto característico de una de las imágenes, cuál es el punto característico más parecido en la otra imagen. Para evitar falsos positivos, una técnica muy utilizada es aceptar como válido el emparejamiento solamente si el valor de correlación cruzada más parecido (CC_1) es considerablemente mejor que el del segundo más parecido (CC_2), es decir:

$$CC_1 * \lambda > CC_2 \quad (2)$$

donde $0 < \lambda < 1,0$ es un factor con el que especificamos cuanto mejor debe ser CC_1 para considerar el emparejamiento como válido.

Una vez que tengamos la lista de emparejamientos válidos, pasaremos a crear la imagen panorámica resultante. Los vectores que unen los puntos emparejados nos servirán para calcular la traslación que finalmente nos permitirá alinear las imágenes. El vector de traslación final se puede calcular como una media de los vectores anteriormente citados. Con esta información podemos calcular el tamaño de la imagen resultante y copiar las imágenes originales en las posiciones correspondientes.

Finalmente, como consideraciones generales para la implementación de la práctica, crearemos una función JavaVis llamada FPanorama que estará localizada en el grupo Applic del menú funciones de JavaVis. La función podrá tener todos los métodos auxiliares que sean necesarios para la correcta estructuración y claridad del código. La función manejará, al menos, tres parámetros de entrada. El primero se corresponderá con el parámetro que utiliza la función FNitzberg, el segundo será el tamaño de ventana que tomaremos para calcular la correlación cruzada entre dos puntos característicos y el tercero será el factor λ para la eliminación de falsos positivos. Además, el alumno podrá incorporar más parámetros a la función si lo considera necesario, justificando razonadamente su uso. También tendrá que tener en cuenta que esta función ha de trabajar con una secuencia de imágenes. Se valorará que en el resultado se incluyan, además del panorama creado, imágenes que muestren los pasos intermedios.

2. Parte experimental

Aparte del proceso de implementación de la función FPanorama, el alumno deberá desarrollar un proceso de experimentación. En primer lugar, se tendrá que realizar el ajuste de los distintos parámetros que utiliza la función. A continuación, los experimentos realizados por el alumno deberán buscar los límites de la función para, razonadamente, explicar bajo que condiciones el funcionamiento de su función no está garantizado. En caso de que se haya implementado una parte opcional, el alumno deberá demostrar mediante experimentos cómo la parte opcional incorporada supone una mejora sobre los límites originales de la función.

3. Documentación

La práctica irá acompañada de la correspondiente documentación. En dicho documento, el alumno explicará y justificará razonadamente los detalles de diseño de la función implementada, experimentación, y resultados.

4. Parte optativa

Como parte optativa de la práctica se deja al alumno la posibilidad de mejorar este proceso básico de creación de panorama. Como guía orientativa, el proceso puede ser mejorado en tres aspectos fundamentales: la búsqueda y emparejamiento de características; el cálculo de la transformación que alinea las imágenes en función de los emparejamientos; y la transformación de las imágenes de entrada para que no se aprecien en el panorama final los cortes en los límites de las imágenes.

5. Normas de entrega y evaluación de la práctica

La práctica es individual. Cualquier indicio de plagio supondrá el suspenso de la parte de prácticas y, por consiguiente, de la presente convocatoria de la asignatura. Se avisa al alumno que existe la posibilidad de utilizar sistemas automáticos de detección de copia.

La práctica se valorará de la siguiente manera:

- 25 % por código y funcionamiento. Se probarán varios ejemplos y se valorará el funcionamiento final. Si la práctica no cumple con un mínimo de funcionamiento, se suspenderá por completo sin tener en cuenta los apartados siguientes. En cuanto al código, deberá estar debidamente estructurado e indentado y haciendo un uso adecuado del lenguaje de programación. El código deberá estar debidamente comentado (incluso para la generación de javadoc).
- 25 % por la experimentación.
- 30 % por documentación. La documentación se podrá entregar en cualquier formato. Será responsabilidad del alumno asegurar que la documentación pueda ser leída en cualquier sistema operativo.
- 20 % por la parte optativa.

Fechas de entrega de la práctica:

- El plazo de entrega de la práctica finaliza a las 24 horas del miércoles 24 de Abril del 2013.
- La entrega se realizará a través del sitio web moodle de la asignatura.