



Markerless augmented reality application

Realidad Virtual y Aumentada (RVA)

Autor: Antonio Gómez Giménez

Email: i72gogia@uco.es

Córdoba
(2022/2023)



UNIVERSIDAD DE CÓRDOBA

Índice:

1. Descripción del problema:	2
2. Desarrollo de la aplicación:	1
3. Conclusiones:	4
Bibliografía:	6

1. Descripción del problema:

Para la realización de esta práctica se nos planteaba el siguiente problema:

Ser capaces de crear un programa de Realidad Aumentada donde sobre un vídeo de entrada, se añadan elementos virtuales. Dichos elementos virtuales pueden ser:

- Una imagen plana.
- Una reproducción de archivo de vídeo.
- Una vista en tiempo real de una webcam.

También se recalca, la adición de funcionalidades extra como por ejemplo:

- El programa permite usar al menos dos tipos de keypoints (uno real y otro binario).
- El programa permite guardar en vídeo de salida la escena con objetos virtuales.
- El programa permite insertar objetos 3D: sencillos (cubo, pirámide, etc.) o complejos (ogro).

2. Desarrollo de la aplicación:

Tras comentar los objetivos de dicha práctica, se comenzó por analizar y entender el esqueleto de código dado por el profesor. A partir de ese punto, se comenzó el desarrollo para poder alcanzar los objetivos anteriormente especificados.

Para los tres puntos base no ocurrió ningún problema, simplemente se realizó el código que permite realizar dichas funcionalidades. A parte, se consiguió capturar una escena propia con un patrón propio para realizar las pruebas.

Si acaso, se puede destacar que se intentó utilizar un método usado para eliminar falsos emparejamientos. Este método es el de D. Lowe, llamado ratio test, que consiste en la comparación de los dos keypoints, en concreto ver si la distancia de un keypoint se encuentra dentro de un ratio en la distancia del otro keypoint.

Respecto a los puntos opcionales si que hubo más problemas, ya que, simplemente para poder probar un tipo de keypoints de tipo float, es necesario implementar el módulo nonFree, lo que implica instalarse opencv de forma manual con los módulos necesarios.

Respecto a la comparativa de los dos tipos de keypoints, se puede apreciar que cuando se utilizan los keypoints binarios va más rápido computacionalmente que cuando se trabaja con float, esto es lógico, ya que trabajar en punto flotante es mucho más laborioso que trabajar en binario para el procesador.

Respecto a la grabación de la salida de la aplicación, hubo ciertos problemas a la hora de la implementación pero finalmente se solucionaron y funciona correctamente.

Finalmente, respecto a la reconstrucción del objeto 3d en la escena, esto fue algo más complicado, ya que aplicar el ogo implicaba compilar de nuevo opencv para implementar dichos módulos, por ello se optó por crear una casa a base de coordenadas en 3d, dibujando líneas y polígonos.

Antes de realizar el dibujo se realizó un prueba con una imagen, es decir, igual que en el caso sencillo, pero implementar la realización de dicha función pero en coordenadas 3d, ya que si no era capaz de realizar esto, no sería capaz de realizar una casa.

Finalmente se consiguió implementar la imagen en coordenadas 3d en la escena y a futuro, la realización de una casa en 3d.

Cabe destacar que el rendimiento del portátil a la hora de renderizar la casa cae en picado, por ello, se recomienda que se grave con la opción -save para que al terminar el procesamiento se pueda ver de una forma más cómoda.

3. Conclusiones:

Para concluir el trabajo, cabe destacar que se han conseguido cumplir los objetivos y aparte, se ha comprendido el trabajo interno que conlleva la realidad aumentada. Sobre todo cuando nos fijamos en los recursos computacionales y en cómo se ralentiza el portátil usado a la hora de lanzar dicha aplicación.

Bibliografía:

Los recursos usados para la realización de dicha aplicación se han encontrado en la siguiente página web.

- <https://docs.opencv.org/4.x/> (documentación de opencv)