

Parallax-Portal

Implementado bajo el uso de Python, Pygame y OpenGL

Motivación

- Aplicación interactiva, sencilla y fácil de implementar
- No requiere de elementos externos
- Ideal para una experiencia corta
- Posible aplicación en un museo

Detección de caras con Haar Cascade

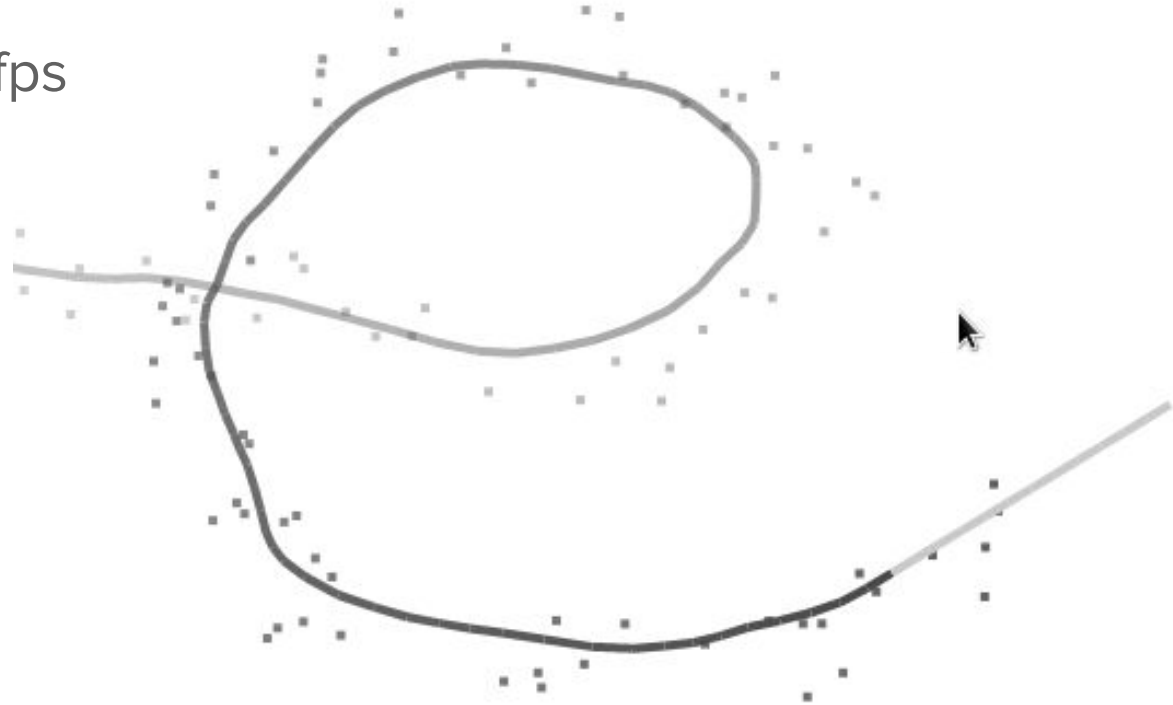
- Algoritmo entrenado
- Barrido a lo largo de la imagen
- Tamaño mínimo y máximo de caras a buscar
- Detección múltiple
- Computacionalmente costoso
- Recorte de la imagen
- Ejemplo



Filtrado mediante filtro de Kalman

- Almacenamiento de posición y velocidad
- Predicción del próximo estado
- Continuación de movimiento ante pérdida de la cara
- Paso de 30 fps a 60 fps
- Detección de salto

Ejemplo



Visualización en 2D

- Pygame
- Efecto de paralaje
- Alterar distancia aparente mediante la sensibilidad del movimiento de la imagen

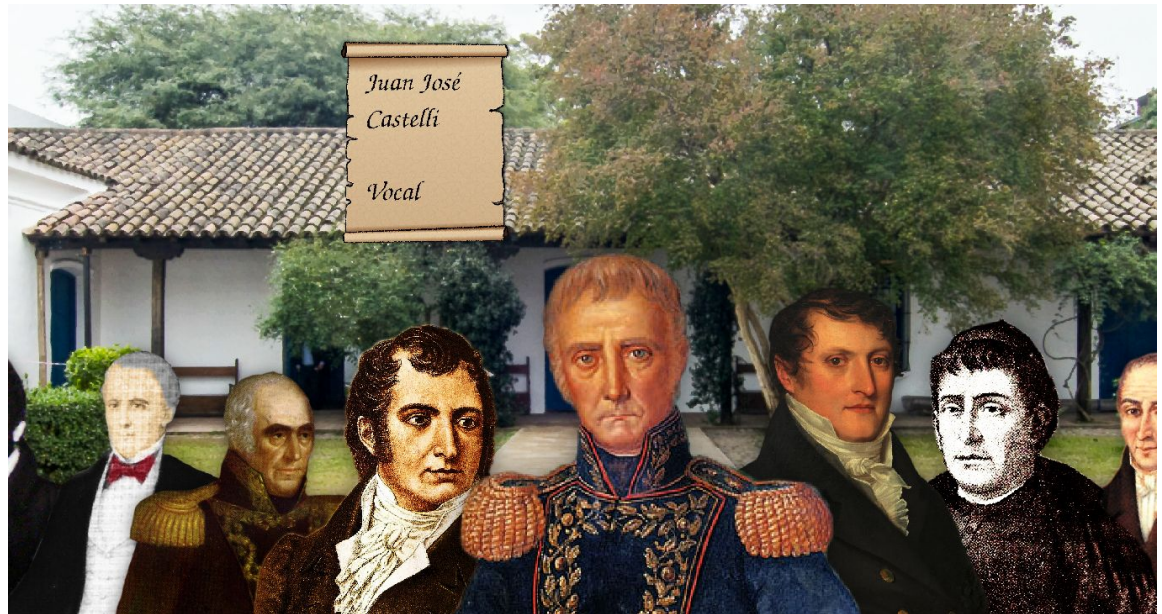


Imagen de fondo basada en foto de «Secretaría de Cultura de la Nación» [CC BY-SA 2.0].

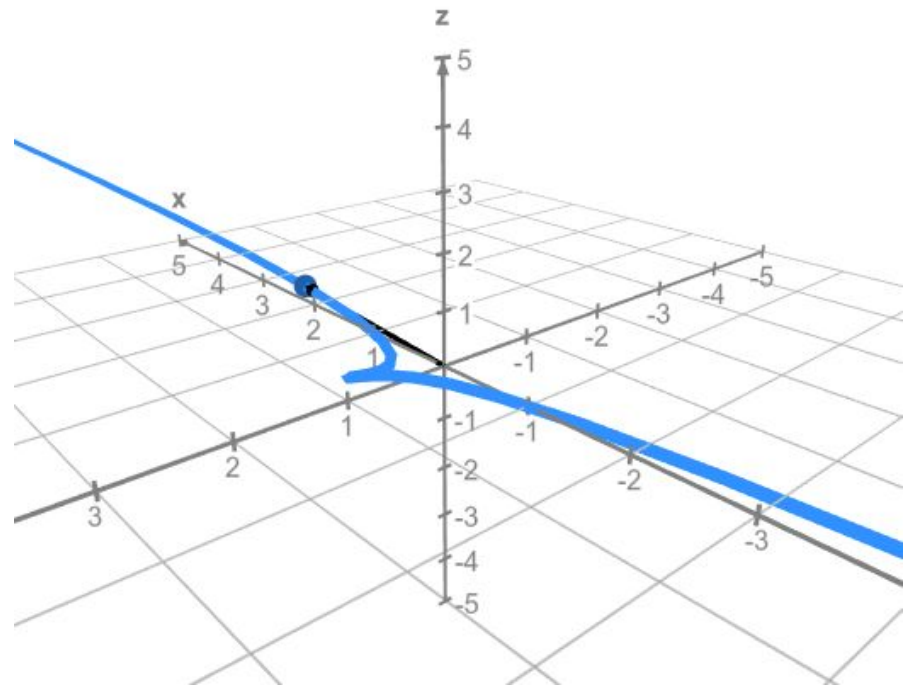
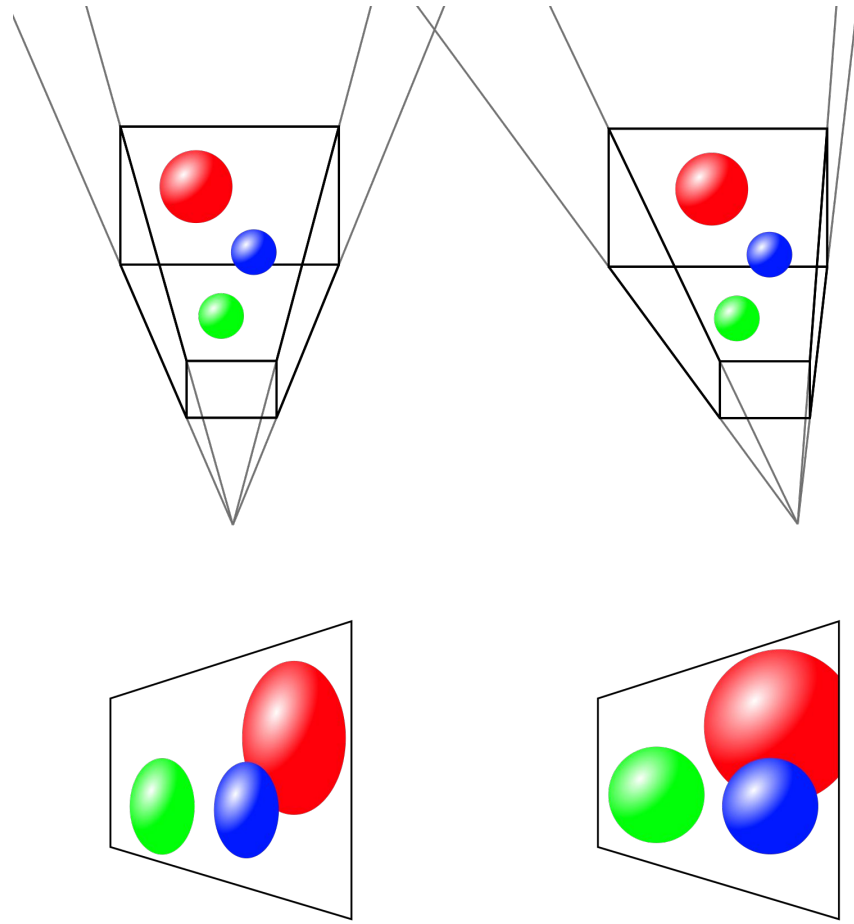
Visualización en 3D

- OpenGL y GLM



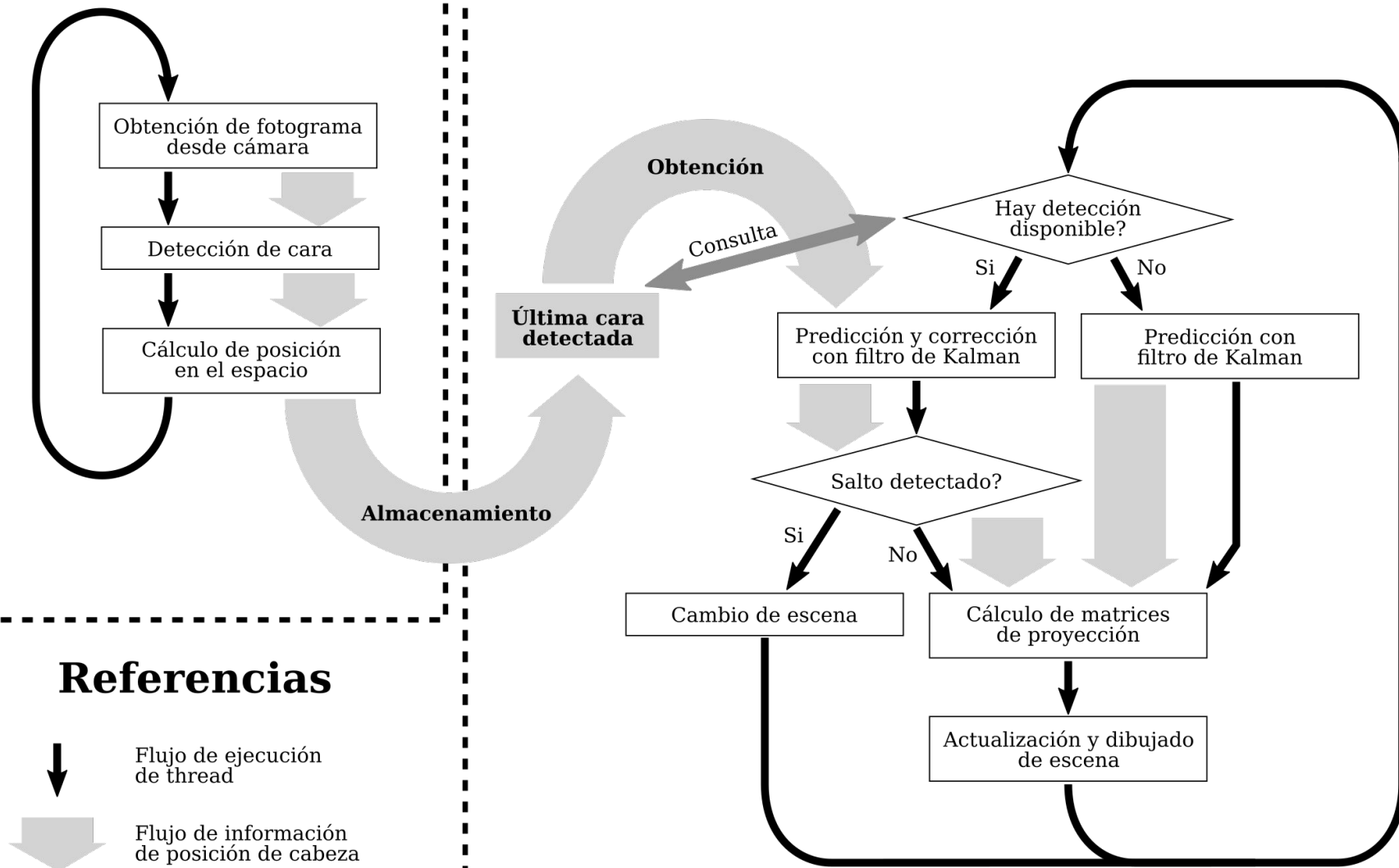
Texturas basadas en imágenes de Sailko [CC BY-SA 4.0], «Darkwood67» (deviantart.com/darkwood67) y Øyvind Holmstad [CC BY-SA 4.0]

Visualización en 3D



Thread de detección

Thread principal



Propuestas de mejoras

- Realizar una escena tridimensional realista por medio de tecnologías aplicadas a los videojuegos como shaders
- Migrar todo el código a 3D mediante el uso de OpenGL
- Utilizar un monitor 3D para mostrar imágenes distintas para cada ojo del usuario
- Mejorar las imágenes, contratando a un artista es posible hacer una experiencia muy interesante y pulida

Muchas Gracias
