Rojo

Verde

Azul

Arquitecturas Paralelas

-Seguimiento de Imagen 2.0-

-Raul Villar Ramos-

Grupo 6 2ª Entrega.

Información y Recursos:

Conocimientos adquiridos de programación en cursos anteriores y la documentación aportada en Faitic (Practicas)

Manual OpenCv: http://docs.opencv.org/opencv2refman.pdf

Entorno usado: CodeBlocks / Windows 7 - 64 bits

Diseño usado y como llegué a mi solución:

Después de analizar los requisitos de la práctica, me di cuenta que para buscar una determinada imagen en una determinada imagen, las imágenes no siempre son 100% idénticas entonces decidí rehacer todo mi programa y hacer lo siguiente:

Crear una máscara de la imagen objetivo y compararla con toda la imagen de fondo, y nos quedamos con la diferencia mas baja de todas.

Para una lectura mas cómoda y una organización del código, traté de hacer casi todo con funciones y un procedimiento.

Una de ellas será una función inline de tipo uchar nombrada getXY que la usaré para crear punteros a las imágenes.

Paso 1:

Una vez iniciado el programa e introducir las imágenes hace lo siguiente:

Crear mascara de la mariquita y guardarla.

Lo hago a través de la función Fmascara obtengo la máscara y la almaceno directamente sin retornar nada.

Paso 2:

Buscamos por primera vez el objetivo, en mi caso obtuve que empieza en

x – 295

y- 216

Ahora invoco a getDiferencia que es una función que funciona de la siguiente manera:

Primero creo una copia del fondo para recorrerlo con la máscara.

Después con un bucle me dedico a montar la máscara de la mariquita con el fondo,

Una vez termino de fundirlas activo mi función calcular Diferencia que compara mi montaje con ese frame y me dice la diferencia actual.

Paso 3:

Inicio mi bucle principal,

Suponiendo que partimos de la posición anterior de la mariquita, busco en un área de desde -10 hasta +10 pixeles tanto de horizontal como vertical y si en algún momento la diferencia es inferior a 100.000 suponemos que la encontramos.

Rompemos el bucle y invoco a mi función de dibujar PintaCuadrado ,

en la que uso una función propia de open cv cvRectangle

El bucle se repite mientras queden fotogramas.

EJ:

Mientras que (Frame=cvQueryFrame(VideoFondo))

Mientras el video contenga fotogramas lo iremos reccoriendo usando:

IplImage* Frame = cvQueryFrame(VideoFondo)

Lo que hace esta llamada es guardar en Frame el siguiente fotograma si existe, en caso de no existir termina el bucle.

Una vez acabado el bucle esperamos una tecla y liberamos memoria y eliminamos la ventana.

Notas:

He usado paralelismo en la función CalcularDiferencia, dado que es el trabajo mas pesado y repetitivo.

Una breve explicación de ella:

Creo 2 bloques de 128 bits para las comparaciones.

Le digo que si la mariquita es mas larga que el fondo salga y retorne -1 porque sería error.

Empieza el bucle con paralelismo:

Cargo las imágenes en su bloque,

Realizo la suma absoluta (_mm_sad_epu8(blo1,bloqu2);

Voy guardando el resultado y una vez terminada la suma total es retornada;

Escola Superior de Enxeñeria Informática. 2015

Traté de ordenar bien el código y comentar todo lo posible además de usar nombres de fácil reconocimiento para una compresión más cómoda.

También traté de segmentar el código en funciones para hacerlo mas versátil.

Raúl Villar Ramos