

TRABAJO PRÁCTICO N° 3

PROCESAMIENTO DE IMÁGENES

INTEGRANTES:

- ALMADA, Manuela - A-4692/6
- CHIAPPE, Maximiliano - C-7542/6
- LENARDOZZI, Juan Sebastián - L-3398/7

DOCENTES:

- ALLIONE, Joaquín
- CALLE, Juan Manuel
- SAD, Gonzalo

Preparación

Para comenzar el trabajo práctico, descargamos todo el material proporcionado por la cátedra, disponible en el repositorio de Git indicado.

Todos los elementos fueron almacenados dentro de una misma carpeta, la cual incluye el script “**Tp3_PDI**”, encargado de resolver los problemas planteados.

1. Introducción y Objetivo

El objetivo general de los problemas es analizar videos de datos, identificar el frame en el que los datos se encuentran detenidos y determinar el valor de cada uno. Como archivos de entrada se proporcionaron cuatro videos con nombres **tirada_<id>.mp4** que contienen el lanzamiento de cinco dados rojos.

2. Código

Importación de librerías

Se importaron las librerías necesarias, en este caso *cv2*, *matplotlib* y *numpy*.

Detección de quietud de los dados

Esta etapa se realizó mediante la función **obtener_frame_dados_quietos** y **obtener_frame_dados_quietos_especial**.

A. Preprocesamiento y segmentación de los dados

Se reduce el frame a $\frac{1}{3}$ del tamaño original para acelerar el procesamiento.

Se convierte la imagen al espacio HSV para segmentar el color rojo de los dados.

Se aplica un umbralizado local sobre la imagen en escala de grises y se utiliza la función auxiliar **imfillhole_v2** para el relleno del interior de los objetos.



B. Detección de candidatos

Se encuentran los contornos con **cv2.findContours** y se aplican filtros geométricos (por área y relación de aspecto).

C. Detección de quietud

Solo se analiza el movimiento si se detectan 5 dados (a partir de los centroides encontrados).

Se ordenan los centroides por su coordenada en x para asegurar que se compare el mismo dado entre distintos frames.

Se calcula la distancia euclíadiana entre el centroide actual y el anterior para cada dado y se calcula un movimiento promedio.

Si el movimiento promedio es menor al umbral determinado previamente (2 píxeles) se incrementa el contador de quietud.

Para finalizar, se considera que los dados están quietos si el contador alcanza 3 frames seguidos.

obtener_frame_dados_quietos_especial se utiliza para el video de la tirada 2 ya que la función principal no detecta quietud de los dados para este video. Esta función utiliza una estrategia alternativa donde se prioriza el mejor frame basado en un score que maximiza la cantidad de dados detectados y minimiza el movimiento.



Conteo de puntos

Una vez obtenido el frame donde los dados están detenidos, se procede a contar los puntos de cada dado con la función **analizar_valores_dados**.

A. Segmentación y aislamiento de datos

Se realiza nuevamente la segmentación roja, una limpieza con una apertura morfológica y el relleno de huecos para obtener los contornos sólidos.

Se buscan los contornos con **cv2.findContours** y se itera sobre cada dado.

B. Detección de puntos

Se trabaja sobre cada dado. Se aísla la ROI de cada dado y se recortan los bordes internos.

Sobre cada ROI (dado) se aplica Canny para detectar los puntos blancos:

Se aplica un suavizado con gaussianBlur y se utiliza el detector de bordes Canny.

Se filtran los contornos encontrados usando el rango [área_min,

área_max_dinámica]. También se filtra por circularidad (mayor a 0.4).

Se calcula la cantidad de puntos como la cantidad de contornos que pasan los filtros.

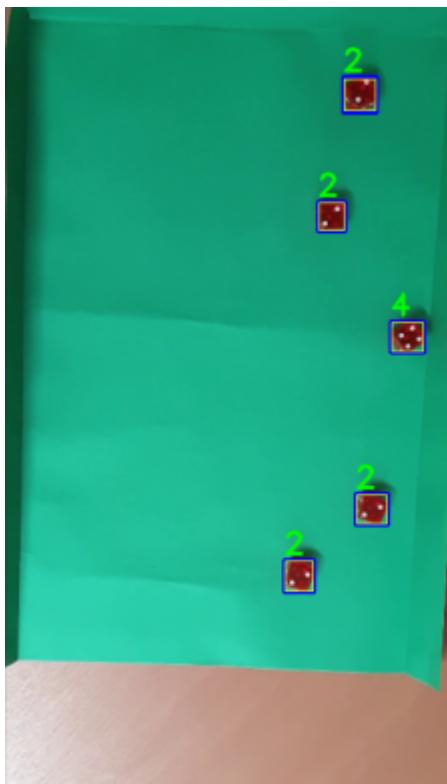
C. Presentación de Resultados

Se genera la salida por consola mostrando el valor individual de cada dado y la suma total.

```
>>> analizar_valores_dados(imagen3)
=====
      RESULTADO DE LA TIRADA
=====
Dado 1: 2 puntos
Dado 2: 2 puntos
Dado 3: 4 puntos
Dado 4: 2 puntos
Dado 5: 2 puntos
TOTAL SUMA: 12
```

Se muestra el frame con un bounding box azul sobre el dado, un bounding box interior amarillo para mostrar la región analizada y el número detectado

en verde.



Generación del video de salida

Se utiliza la función **generar_video_reemplazando_frame**.

Se copia el video original frame a frame. Al llegar al índice del frame quieto, se inserta el frame modificado.

Para que el resultado sea visible el frame modificado se repite 15 veces. Esto pausa la acción permitiendo que se observen mejor el bounding box y el número de cada dado.

Por último se utiliza un bucle para generar los cuatro videos modificados.

Conclusiones

Se logró detectar el momento donde los dados se encontraban quietos y los valores en cada tirada.

El caso de la segunda tirada representó un desafío debido a la detección inestable del frame donde se detenían los dados. Esto fue resuelto aplicando una solución alternativa que prioriza la detección del mejor frame.