



## Trabajo Práctico N.º 3

Materia: **IA4.4 Procesamiento De Imágenes**

Carrera: **Tecnicatura Universitaria en Inteligencia Artificial**

Primer integrante: **Nicolás Noir: N-1273/4**

Segundo integrante: **Ignacio Eloy González: G-5933/1**

Tercer integrante: **Gonzalo Calvo: C-7345/8**

Profesores: **Gonzalo Sad, Juan Manuel Calle, Julián Alvarez**

Fecha: 25/11/2024

**Universidad Nacional de Rosario**  
**Facultad de Ciencias Exactas, Ingeniería y Agrimensura**

# ÍNDICE

<b>PORTADA.....</b>	<b>1</b>
<b>ÍNDICE.....</b>	<b>2</b>
<b>PROBLEMA 1:.....</b>	<b>3</b>
Punto A:.....	3
Punto B:.....	3
Metodología de resolución.....	4
<b>RESULTADOS.....</b>	<b>10</b>
<b>CONCLUSIONES.....</b>	<b>11</b>

## PROBLEMA 1:

En este trabajo práctico, se tiene como objetivo detectar automáticamente cuando se detienen los dados en diferentes videos de tiradas de 5 dados, para así reconocer el valor de cada dado quieto.

### Punto A:

Para este punto decidimos utilizar CIELAB para diferenciar los dados de color rojo, en específico el canal “a”, aplicamos un umbralizado sobre este canal, obteniendo una imagen donde se resalten los dados en el momento donde aparecen. Después decidimos utilizar componentes conectados para encontrar todos los posibles dados, necesitando aplicar filtro de relación de aspecto y área. Luego pensamos que para detectar cuando un dado está quieto, la mejor forma sería observar su posición en el frame anterior y si es igual a la del frame actual, uno podría guardar el número de frame en donde se observa que el dado está quieto. Esto lo implementamos utilizando los centroides de los componentes conectados que ya fueron filtrados, entonces haciendo la distancia euclídea entre las posiciones de los centroides entre el frame actual y el anterior podríamos observar si el dado se sigue moviendo o está quieto, ya que tomamos como supuesto que el dado al girar sobre su eje, su componente conectado no sería siempre el mismo y su centroide iría cambiando, permitiéndonos diferenciarlo de un dado quieto. Una vez encontrados todos los frames donde los 5 dados se mantengan quietos, guardamos su número de frame en una lista y mediante una función elegimos los frames que sean una serie de números consecutivos de más de tres números, es decir, si pensamos que una vez el dado se queda quieto, todos los frames siguientes donde este quieto se van a detectar, esto nos devuelve todos los frames donde el dado esté quieto y no posibles detecciones falsas.

### Punto B:

Ahora necesitamos grabar el video de salida donde se resaltan los dados con un bounding box y su valor es agregado por encima. Para este momento ya contamos con los frames donde los dados están quietos y las posiciones de los centroides de los dados. Lo primero va a ser iterar sobre los frames de nuevo, donde en caso de ser un frame donde no están los dados quietos, se graba sin ningún agregado y en caso contrario, nosotros pensamos que al estar los dados quietos el cálculo de su valor podría ser hecho sobre el primer frame de dado quieto y los siguientes frames se podría aprovechar los valores de los dados encontrados para grabar sin tener que procesar y encontrar los números de los dados en cada frame. Para encontrar estos valores de los dados, utilizamos los centroides de los dados anteriormente encontrados junto con el número de frame y en caso de coincidir el valor del frame con el de los centroides, iterar sobre los centroides, resaltar el dado con un bounding box y agregar el valor del dado por encima. Para esto primero necesitamos preprocesar la imagen para poder encontrar los puntos dentro de los dados, ya que los centroides nos dan las posiciones de los dados, que agregando un padding, seleccionado manualmente después de un análisis del tamaño de los dados, recortamos la imagen y

contando la cantidad de contornos circulares dentro del dado con la ayuda del factor de forma encontramos el valor. El preprocesamiento no va a ser más que llevar la imagen BGR a escala de grises y aplicar un Tophat con un kernel elipse que pensamos sería la mejor opción para resaltar los círculos del dado. Una vez hecho el recorte sobre esta imagen preprocesada, binarizamos la imagen con un threshold seleccionado manualmente y encontramos los contornos dentro de esta imagen, que mediante el factor de forma y obligando a que su perímetro sea mayor que 0, encontramos el valor del dado. Luego hacemos el bounding box del dado con el padding anteriormente mencionado y agregamos el valor del dado por encima de este.

## Metodología de resolución

Lo primero que hicimos fue ver un frame donde se vean los dados y encontrar los colores del fondo, de los dados y de la mano, ya que nuestro primer objetivo era reconocer los dados para saber cuando se quedaban quietos.

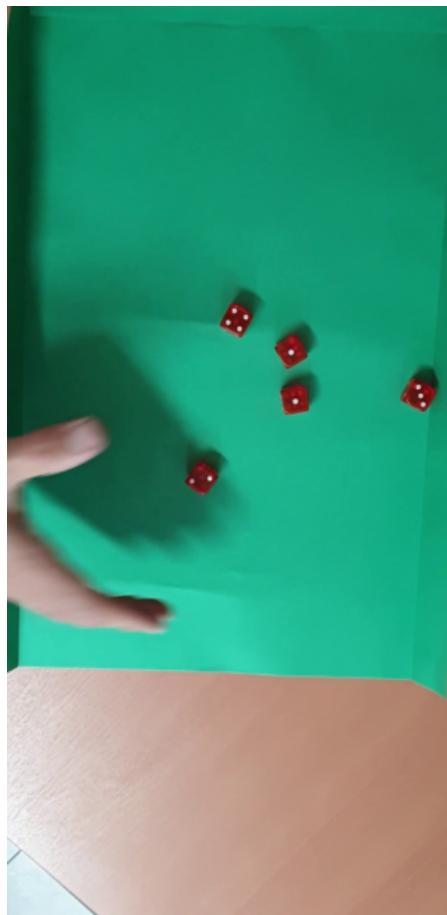


Figura 1.1

Luego pensamos en utilizar otro mapa de colores que no sea RGB, ya que este no facilita la división de colores y otros mapas de colores como HSV, HSL o CIELAB sí. Por lo que decidimos ver los diferentes canales de CIELAB.

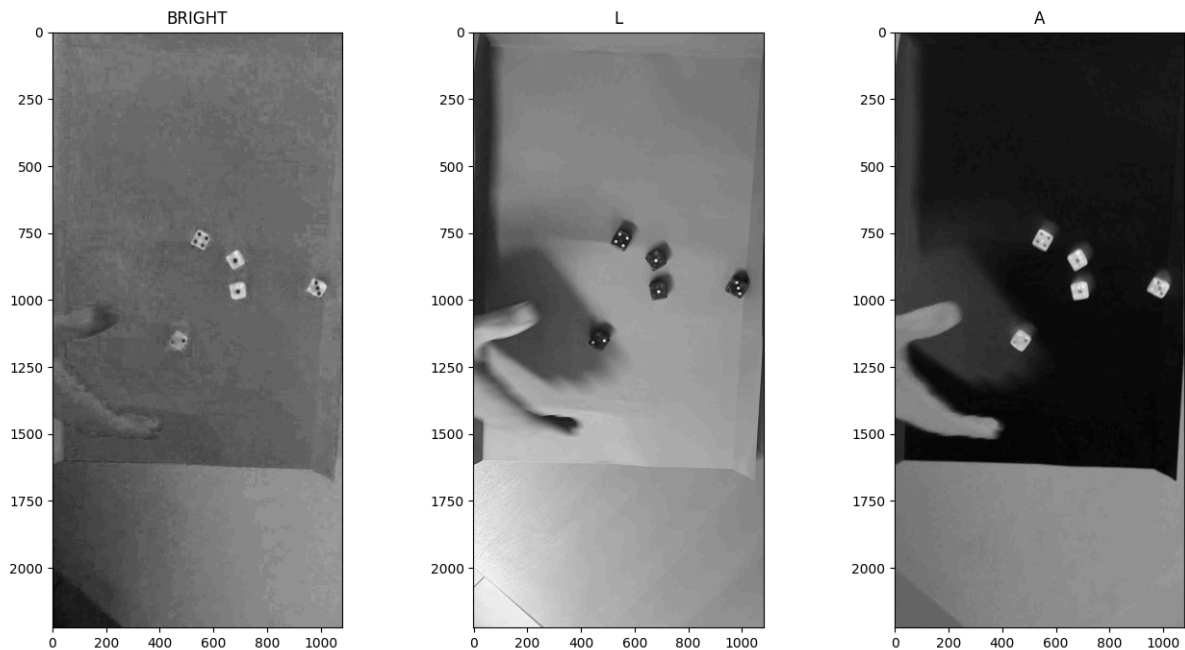


Figura 1.2

Después de comparar los tres canales de CIELAB observamos que el canal "a" era el mejor para este procedimiento, ya que los dados y el fondo eran de valores muy diferentes, permitiéndonos utilizar un umbral que los diferencie fácilmente.

Luego, con los dados ya diferenciados del fondo, decidimos aplicar un umbral para así luego poder detectarlos más fácilmente, como se ve en la Figura 1.3.

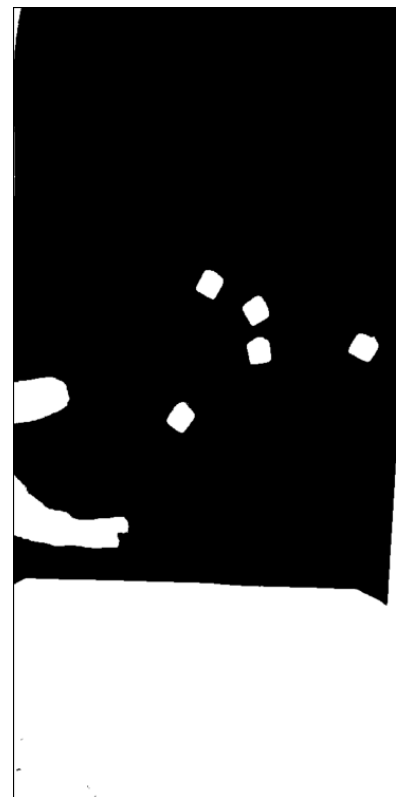


Figura 1.3

El siguiente paso fue buscar los dados utilizando componentes conectadas, ya que podríamos obtener los centroides, y los diferentes objetos de la imagen, con lo que luego iteramos en las diferentes formas, buscando aquellas que tengan una relación de aspecto cuadrada, ya que componentes conectadas también nos da sus stats, de donde podemos saber su alto y su ancho. Además de esa relación, también buscamos que las áreas de los dados tengan un valor acorde al tamaño que observamos, para así quitar posible ruido.

Gracias a este filtro, obtenemos los dados, y nos guardamos los centroides de cada dado, ya que esto nos interesa para saber en qué frame del video, los dados no se mueven, o sea, los centroides no cambian de posición.



Figura 1.4

Para saber en qué momento los dados se detenían, pensamos en comprar la posición de los centroides de cada dado en el frame actual y el frame anterior. Primero, pensamos en poder guardarnos la sucesión de frames donde los dados estaban quietos así poder remarcarlos y buscar sus puntos, hasta qué o se vuelvan a mover, o como se ve en los videos, otro objeto tape alguno de los dados, ya que consideramos que si uno de los dados no se ve, no estaría cumpliendo con la consigna de mostrar los 5 dados.

Utilizamos distancia euclídea entre los dados de los frames anteriores y los actuales. Guardándonos cuando los dados están quietos.

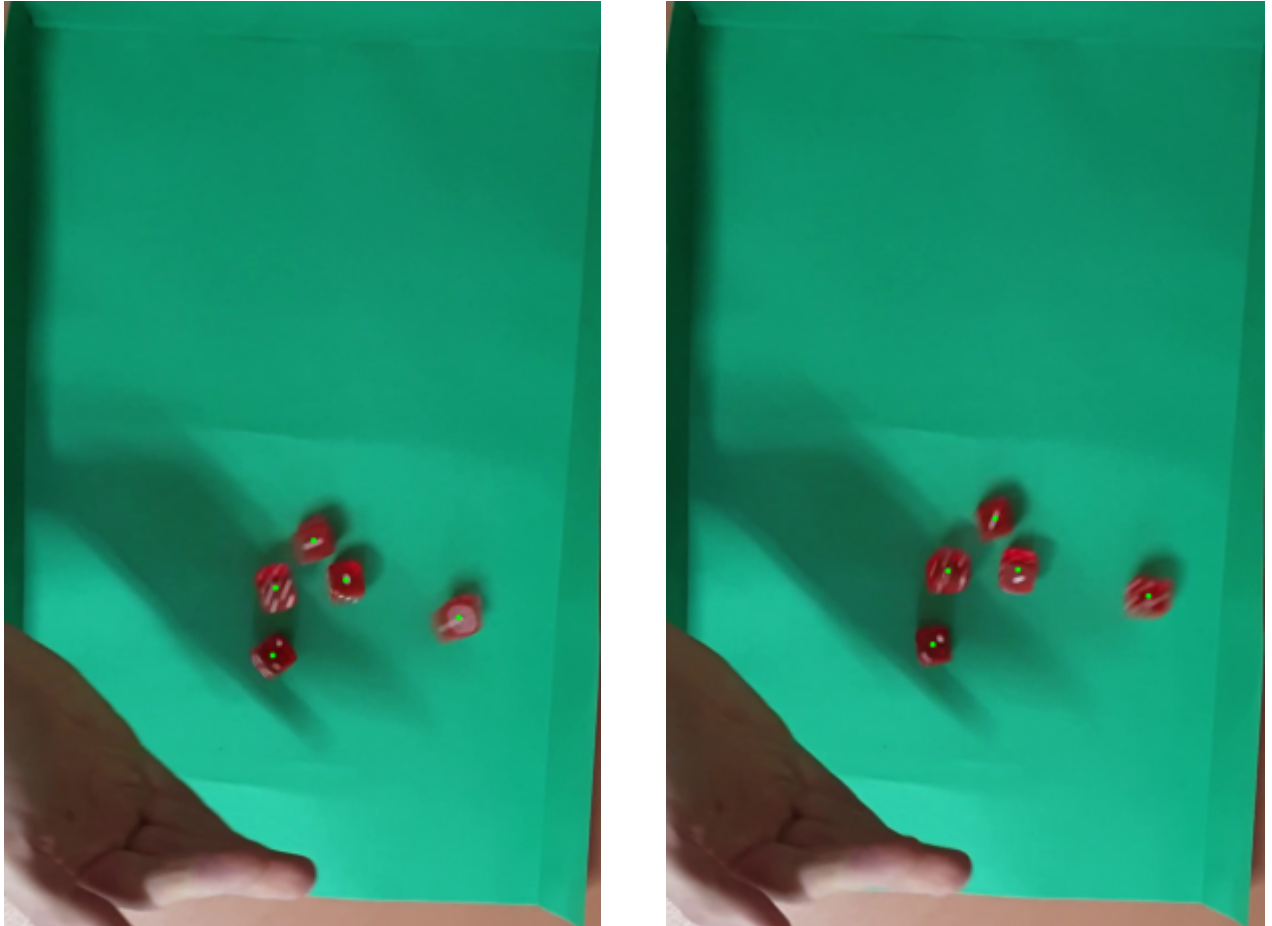


Figura 1.5

Ahora, que ya tenemos el índice de los frames donde los dados están quietos, iteramos sobre todos los frames y sus índices, buscando el primer frame de la lista de frames con dados quietos, para así, poder realizarle diferentes transformaciones para encontrar sus puntos. Los índices de frames quietos encontrados para poder ser utilizados, se llevaron a una lista y si estos son una serie de más de 3 números consecutivos, los contamos como frames de dados quietos.

Para encontrar los valores de los dados quietos, aplicamos un blur a la imagen original, para así poder detectar mejor la formas de los dados, todo esto ya habiendo pasado a escala de grises. Luego probamos de utilizar tophat sobre cada dado, lo que nos dio un resultado pobre, ya que los puntos de las caras laterales no podían ser borradas. Por eso replanteamos dónde realizar las operaciones, y probamos en realizarla a la imagen entera, lo que nos dio resultados mejores.

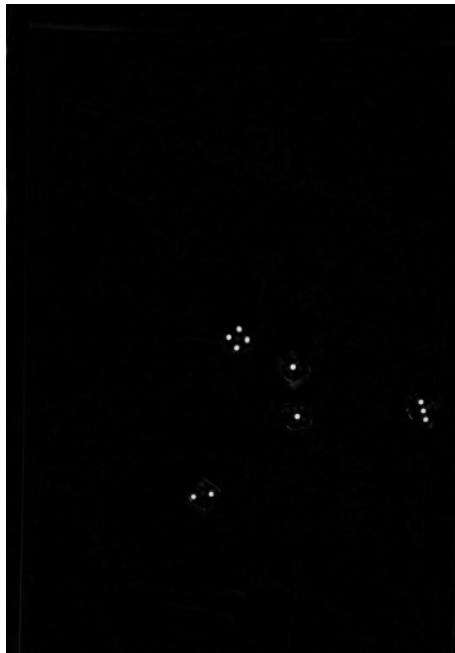


Figura 1.6

Luego, recortamos cada dado, agregando un margen desde su centroide, hacia cada una de sus diferentes direcciones. Luego, lo umbralizamos para así poder encontrar los círculos de las caras superiores de los dados. Para encontrar estos círculos utilizamos contornos



Figura 1.7

La manera de detección de puntos fue igual a la del trabajo práctico anterior, fue diferenciando sus factores de forma, ya que los puntos dentro del dado son circulares, agregando en cada iteración el valor correspondiente a cada dado

Al final, realizamos un cuadrado alrededor de cada dado y el valor de su cara superior, como fue pedido en las consignas.



### Trabajo Práctico N.º 3

Para el resto de frames, únicamente dibujamos el recuadro y su valor, ya que al estar quietos, no hace falta ajustar la posición del cuadrado que le agregamos y tampoco realizar el proceso de detección ni umbralizado ni nada de lo que se había realizado anteriormente.

Todo este proceso de la detección de los valores de los datos quietos, se realizó en paralelo con el grabado del video de salida, el cual si no era uno de los frames en donde había datos quietos, el frame se grababa por default y en caso de ser uno de los frames de datos quietos, se realizaba lo anteriormente mencionado.

## RESULTADOS



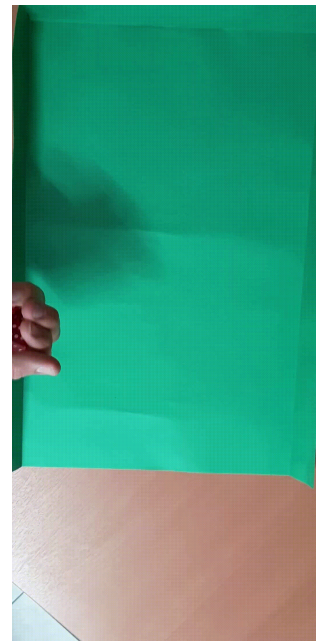
tirada\_output\_1.gif



tirada\_output\_2.gif



tirada\_output\_3.gif



tirada\_output\_4.gif

## CONCLUSIONES

Para dar como finalizado este trabajo práctico nos gustaría indagar en los diferentes planteamientos y problemas que tuvimos a lo largo del desarrollo del mismo. Empezando por qué método utilizar para poder encontrar en qué frame los datos se quedan quietos, primero probando con "Harris Corner Detection", que aunque fue fácil de implementar, daba problema al momento de diferenciar cuando un dato se quedaba quieto, ya que utiliza escalas numéricas muy bajas, por lo que cualquier fluctuación en los cambios de colores, daba valores muy distintos, por lo que nuestra otra opción fue utilizar la opción correcta que fue buscar los centroides de las formas. Pensamos en utilizar los centroides, ya que estamos más familiarizados con este método, y además nos dio mejores resultados.

Al momento de procesar la imagen, como ya se mencionó anteriormente, cometimos el error de realizar el proceso a cada dato, que además de que este proceso sería más costoso computacionalmente, encima daba malos resultados. La decisión de utilizar tophat como método fue gracias a la utilización de la misma en el trabajo práctico anterior, donde también tuvimos problemas con los objetos más claros.

Finalizando con la conclusión, no tuvimos demasiados problemas con este trabajo práctico, gracias a todo el conocimiento obtenido en los anteriores trabajos, siendo este el que menor cantidad de tiempo nos tomó.