

Método para identificar la puntuación de dardos lanzados en una diana

M. Ghidini, P. Quindós de la Riva

7 de mayo de 2024

Resumen

El proyecto consiste en la implementación de un método para la segmentación de una diana es sus áreas de puntuación con la correcta asignación de los puntos. Además se intenta reconocer un dardo lanzado en la diana para obtener la puntuación.

Palabras clave: Procesamiento de imágenes digitales, dardos, segmentación, detección de líneas, transformada de Hough.

1. Introducción

Nuestro problema consiste en la localización, en una imagen, de un dardo en una diana para determinar la puntuación.

Tenemos una diana clásica de forma circular y dividida en veinte secciones, que alternan color entre blanco y negro y con dos círculos concéntricos (bullseye). Hay también dos anillos externos, que alternan color verde y rojo, que dividen ulteriormente la diana en zonas de puntos dobles y triples.

Para calcular correctamente los puntos es necesario que el proceso de procesamiento de la imagen determine correctamente la ubicación del dardo en la diana y cuantos puntos corresponden a esta ubicación.

Todo esto se realiza mediante la utilización de técnicas de procesamiento de imágenes digitales. Nuestro trabajo se ha realizado mediante MATLAB.

Nuestro trabajo se fundamenta en el artículo de Jacob D. Delaney, *Method to Identify and Score Darts thrown into Dartboard* [1].

En el resto de esta documentación analizaremos los siguientes pasos:

- 2. Planteamiento Teórico.
- 3. Resolución Práctica.
- 4. Experimentación.
- 5. Manual de usuario.
- 6. Conclusiones.
- 7. Referencias.
- 8. Tabla de tiempos.

2. Planteamiento teórico

En la siguiente tabla se resumen los objetivos planeados inicialmente para la realización del proyecto:

Hito 1	Búsqueda del artículo de investigación del TD Cuestionario sobre el TDA. Planificación inicial y lista de objetivos. Estudio y comprensión del artículo de investigación.
Hito 2	Investigación de la transformada de Hough. Algoritmo de segmentación de la diana. Algoritmo de identificación del dardo. Revisión horas trabajadas.
Hito 3	Experimentar con diferentes perspectivas y mejora del reconocimiento. Funcionalidades adicionales. Realización memoria del trabajo. Revisión horas trabajadas.
Hito 4	Memoria del trabajo. Presentación. Revisión de la documentación. Revisión de la aplicación.
Hito 5 (Mejora)	Mejorar el algoritmo de detección del dardo. Mejorar la documentación. Realizar un ejecutable sencillo sin instalaciones previas. Recuento de horas trabajadas.

Para llevar al cabo dichos objetivos, se ha pensado dividir el problema en dos partes; en la primera vamos a segmentar la imagen de la diana para asignar a cada pixel de la imagen los puntos correspondientes; la segunda tarea es la localización del dardo, en la que se determinan los pixeles de la imagen que corresponden al punto en el que el dardo está en contacto con la diana.

2.1. División de la diana

2.1.1. Separación de la diana del fondo de la imagen

Se empieza convirtiendo la imagen en escala de grises y calculando el umbral, utilizando el método de Otsu. El resultado es una imagen binaria en blanco y negro y se utiliza para eliminar el fondo sobrante y acelerar el procesamiento.

2.1.2. Segmentación de las zonas de multiplicación (x_2 , x_3 , centro y semicentro) por color

En este segundo paso, vamos a crear una máscara para determinar las áreas de la diana en las que se deben multiplicar los puntos. Podemos hacer esto porque sabemos que estas zonas en la diana están coloreadas de verde y rojo, utilizando los canales de intensidad de rojo y verde para crear dicha máscara.

2.1.3. Obtención de todas las zonas de puntuación

Ahora que tenemos la división en las zonas de multiplicación de los puntos, la siguiente tarea es segmentar ulteriormente para conseguir la división radial y asignar los puntos correctos a cada porción.

Para hacer esto, aplicamos el algoritmo de Canny para la detección de los bordes a la imagen de la diana (convertida en escala de grises); se obtiene así una imagen binaria, de la cual vamos a computar la transformada de Hough, que nos sirve para detectar las líneas radiales.

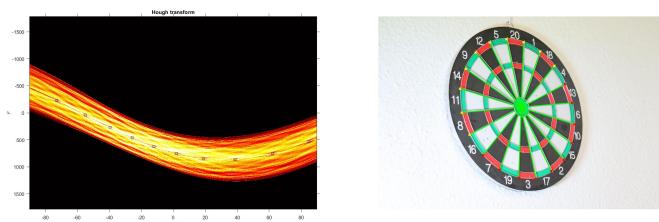


Figura 1: Transformada de Hough para la imagen de la diana

2.1.4. Transformada de Hough

La transformada de Hough constituye un método para comprobar si un conjunto de píxeles se encuentran en una misma línea. Una vez individuadas, esas líneas forman los bordes que nos interesan. Dados n puntos en una imagen, se suponga buscar un subconjunto de esos puntos que se encuentran en líneas rectas. Una posible solución sería buscar primero todas las rectas por cada pareja de puntos y luego todos los subconjuntos de punto que están cerca de dichas rectas. Esto es un trabajo computacionalmente prohibitivo y por eso Hough propuso un método alternativo.

Consideramos un punto (x_i, y_i) en el plan xy y el ecuación general de una recta en la forma $Y = ax + b$. Por el punto (x_i, Y) pasan un numero infinito de rectas y todas estas rectas cumplen con la ecuación $Y = ax + b$, con la variación de a y b . Entonces podemos escribir la ecuación $b = -xa + y_i$; Consideramos ahora el plan ab (espacio de los parámetros) y podemos obtener la recta que pasa por (x_i, y_i) . También por otro punto (x_j, y_j) pasa una recta en el espacio de parámetros asociada con la recta precedente. De hecho, salvo que las dos rectas sean paralelas, la segunda recta intersecta la recta asociada a (x_i, y_i) en el mismo punto (a', b') , donde a' es la inclinación y b' la intersección de la línea que contiene (x_i, y_i) y también (x_j, y_j) en el mismo punto en el espacio xy .

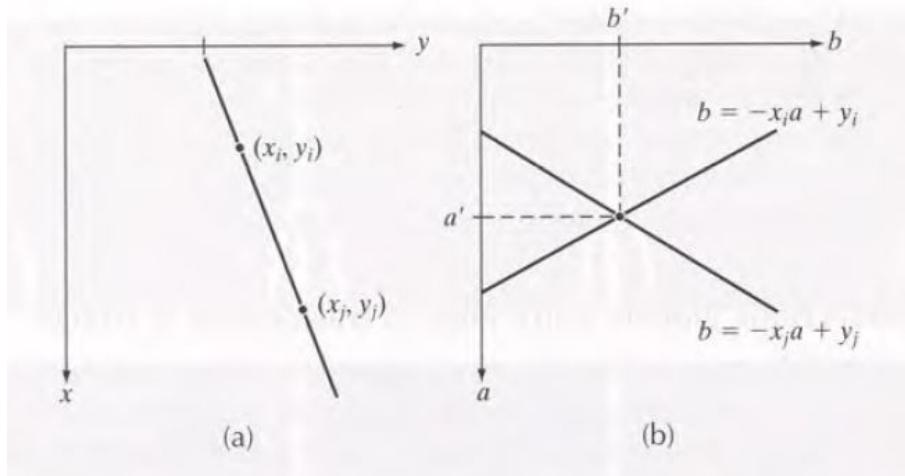


Figura 2: a) plan xy . b) espacio de los parámetros (ab)

Otra forma de considerar esta recta seria utilizando los parámetros theta y rho, definiendo dicha recta en términos de ángulo y radio:

$$x \cos \theta + y \sin \theta = \rho \quad (1)$$

Considerando estos parámetros, el método basado en la Transformada de Hough es el siguiente:

1. Obtener una imagen binaria de los bordes (utilizando por ejemplo un algoritmo de detección de bordes como el de Canny);
2. Individuar las subdivisiones en el espacio $\theta\rho$;
3. Analizar todos los puntos y, cuando se encuentra un punto que pertenece a un borde, iterar sobre todos los posibles valores de θ y ρ ;
4. Definir exactamente las ocurrencias de esos puntos en el espacio $\theta\rho$;
5. Si se encuentran puntos que tienen correspondencia los suma en este espacio.

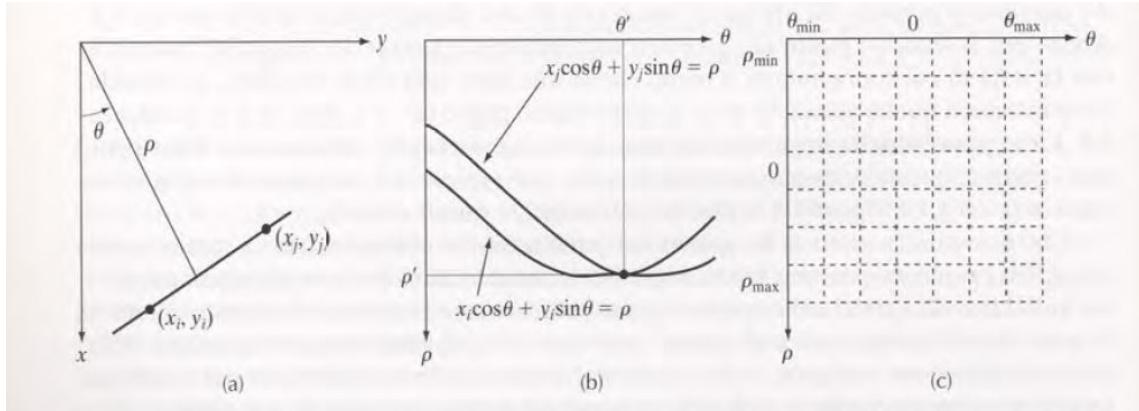


Figura 3: a) recta en el espacio xy con parámetros θ y ρ . b) punto de intersección (θ, ρ) . c) celdas de acumulación en el espacio $\theta\rho$

2.2. Identificación del dardo

Para la identificación del dardo en la diana, hemos empezado implementando el mismo algoritmo propuesto en el artículo [1]. Dicho algoritmo hace la normalización de la diferencia entre la zona clara y la zona oscura por separado, y luego combina los resultados para obtener la imagen del dardo en escala de grises. Dicha imagen es convertida en una imagen binaria, umbralizando con el método de Otsu para crear una máscara binaria del dardo.

Sin embargo, este algoritmo no daba los resultados esperados, entonces se ha decidido implementar un nuevo método para el reconocimiento del dardo en la diana. El nuevo algoritmo extrae el objeto basándose en los canales RGB, en nuestro caso, el azul. En particular, se extrae el canal azul de la imagen con el dardo y se hace una diferencia con la imagen original. Finalmente se convierte la imagen en binaria. Obtenida la región del dardo hay que encontrar los extremos, es decir el punto donde se encuentra la punta del dardo, para asignar correctamente los puntos. Para saber que extremo es donde está la punta nos ayudamos de la orientación del dardo en la imagen. Por supuesto, para que funcione, el dardo tiene que ser pintado completamente de azul.

2.3. Algoritmo de puntuación

Para hacer el algoritmo de puntuación, nos hemos basado en el artículo[1]. Una vez obtenida la división de la diana con la Transformada de Hough, asignamos a cada región la puntuación correspondiente. Luego se toman las coordenadas del pixel donde se encuentra la punta del dardo y se calculan los puntos obtenidos. Para verificar si la puntuación es simple, doble, triple o nula, utilizamos las máscaras creadas anteriormente con el algoritmo de segmentación de la diana.

3. Implementación

El proyecto está desarrollado en MATLAB y se compone de un script principal, `Dardos.m` y de dos funciones, `segmentarDiana.m` y `getScore.m`.

Al ejecutarse del script `Dardos.m`, el programa pide al usuario de cargar la foto de la diana y la foto de la diana con el dardo; esto se hace con la función Matlab `uigetfile`. Luego hay una llamada a la función `segmentarDiana.m`, que se encarga de dividir la diana en las diferentes regiones, devolviendo un conjunto de máscaras. El siguiente paso es dividir la diana radialmente, usando la transformada de Hough. Para hacer eso, se convierte la imagen de la diana en escala de grises, se aplica el algoritmo de Canny para la detección de bordes (función `edge`, especificando '`Canny`' en los parámetros) y en el resultado se aplica la función `hough`. Para individuar los peaks y las líneas obtenidas con la transformada, se utilizan respectivamente `houghpeaks` y `houghlines`. Luego, a cada trozo individual se le asigna la puntuación correspondiente.

Sucesivamente hay que detectar el dardo y se hace como sigue:

1. Se extrae el canal azul de la imagen del dardo y se hace la diferencia entre dicha imagen y la imagen original. Para hacer eso, utilizamos la función `imsubtract` de MATLAB.

2. A la imagen obtenida le aplicamos un filtro de la media, utilizando la función MATLAB `medfilt2`.
3. Luego la convertimos en una imagen binaria y asignamos una etiqueta a cada componente conexa de la imagen (función `bwlabel`).
4. Con la función MATLAB `regionprops` obtenemos los valores de las propiedades para cada componente 8-connected (objeto) en la imagen binaria. Las propiedades que nos interesan son :
 - **BoundingBox**, posición y tamaño del box más pequeño que contiene la región. Esto lo usamos para obtener un recuadro que rodee el dardo (esto solo es para mostrar como se ha detectado el dardo y que sea más fácil de entender)
 - **Centroid**, centro de masa de la región; el primer elemento del array resultante es la coordenada horizontal y el segundo la coordenada vertical del centro de masa. Esto lo usamos para obtener los resultados solo cuando la diana está completamente en paralelo y centrada a la cámara.

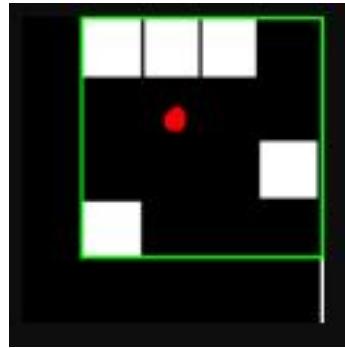


Figura 4: BoundingBox en verde y Centroid en rojo (imagen teórica)

- **Orientation**, indica la orientación del objeto en cuestión. El valor varía entre -90 y 90 (grados). Lo importante para nosotros es indentificar si el dardo tiene orientación positiva o negativa para saber desde que ángulo está tomada la imagen.



Figura 5: Orientation (imagen teórica)

- **Extrema**, puntos extremos de la región. En particular, en nuestro caso, el punto extremo será el left-bottom si el valor de **Orientation** es positivo (label 7) y el right-bottom si el valor es negativo (label 3). Esto lo usamos para detectar en que extremo del dardo está la punta del mismo.

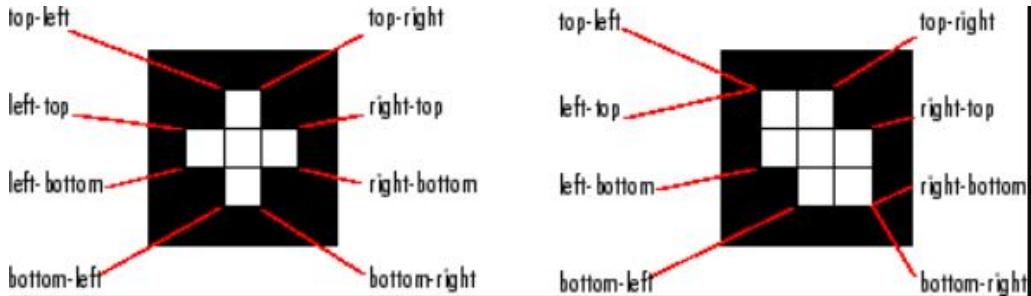


Figura 6: Extrema (imagen teórica)



Figura 7: Aquí podemos observar todos los apartados mencionados antes. La BoundingBox aparece rodeando al dardo, el CentroId (que en la diana centrada a su vez es donde cae el dardo), los puntos obtenidos con Extrema (bien sean con ángulos positivos o negativos obtenidos con Orientation)

Finalmente, para asignar la correcta puntuación, hay una llamada a la función `getScore.m`

3.1. segmentarDiana.m

En esta función se implementa el proceso para dividir la diana en las diferentes regiones. El input de este método es una imagen de tipo rgb.

El primer paso es convertir la imagen en escala de grises con `rgb2gray(I)`.

Luego, para crear la máscara de las zonas caracterizadas por el color rojo y verde (dobles o triples puntos), hacemos lo siguiente:

1. Extraer canal rojo
2. Calcular el umbral con método de Otsu, utilizando la función `graythresh(I)`

3. Binarizar la imagen con `imbinarize(I,T)`, donde T es el umbral calculado precedentemente.
4. Se hace lo mismo con el canal verde
5. Se hace un OR entre las dos máscaras roja y verde.

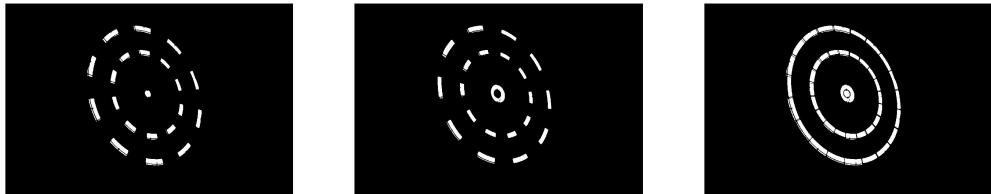


Figura 8: Red mask, green mask y red + green

Luego se aplica una clausura sobre la imagen y se aplica la función `imfill` para rellenar los círculos y obtener el área de la diana en la que se pueden obtener puntos. La imagen complementaria será la zona que vale 0 puntos.



Figura 9: A la izquierda el área de puntos, a la derecha la zona de 0 puntos

Finalmente, para individualizar las zonas de la diana que conceden puntos simples, doble o triples, hacemos una serie de sustracciones entre las máscaras obtenidas anteriormente.

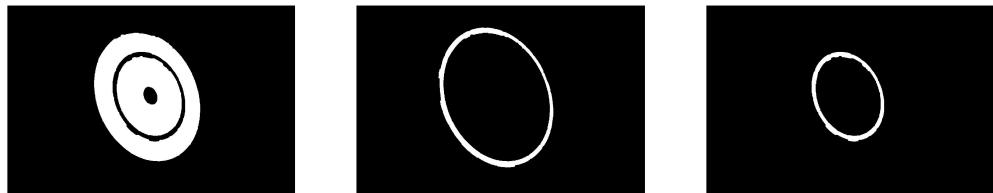


Figura 10: simple, doble, triple



Figura 11: círculo interior y exterior

3.2. `getScore.m`

En esta función se implementa el proceso para determinar la puntuación correcta, basándose en la posición del dardo en la diana.

Tiene como parámetros las coordenadas de la punta del dardo (`x`, `y`), el punto que representa el centro de la diana (`center`), el array de las regiones de puntuación (`region`) y el conjunto de máscaras obtenido con la segmentación de la diana (`masks`).

Primero se encuentra el `hitAngle`, calculando la tangente inversa de $(x - \text{center}, y - \text{center})$ y sucesivamente haciendo el módulo de $((\text{hitAngle} * 180 / \pi) + 360)$ con 360.

Luego, este valor se compara con los valores de los ángulos de cada región de puntuación. Si el valor es mayor del ángulo mínimo y menor o igual del ángulo máximo de una región, entonces el dardo se encuentra en dicha área, que llamamos `hitRegion`; su valor es el valor de la región correspondiente.

El siguiente paso es convertir la `hitRegion` en una máscara (`hitMask`) y hacemos comparaciones con las diferentes máscaras obtenidas con la segmentación de la diana. así podemos comprobar si los puntos son simples, dobles o triples, si el dardo está en el “inner bull” (50 puntos) u “outer bull” (25 puntos), o si se encuentra fuera de la zona de puntuación (0 puntos).

Los puntos obtenidos son guardados en la variable `score`, que se devuelve junto a la `hitMask` al final.

4. Experimentación

Durante la realización del proyecto hemos realizado una gran cantidad de diferentes experimentos buscando mejores resultados. Finalmente hemos conseguido obtener los resultados que deseábamos después de muchas pruebas y de cambiar el algoritmo de detección del dardo.

Como para la realización de todos los experimentos requeríamos de una diana física con la que probar muchos ángulos y perspectivas, empezamos con imágenes de dianas de internet.



Figura 12: imagen diana de internet

Cuando nos llegó la diana realizamos una gran batería de fotos desde diferentes perspectivas y con los dardos en diferentes posiciones. Sin tener en cuenta que los dardos eran del mismo color que la diana (verdes y rojos). Esto hacía que la indentificación de los dardos fuese muy complicada.

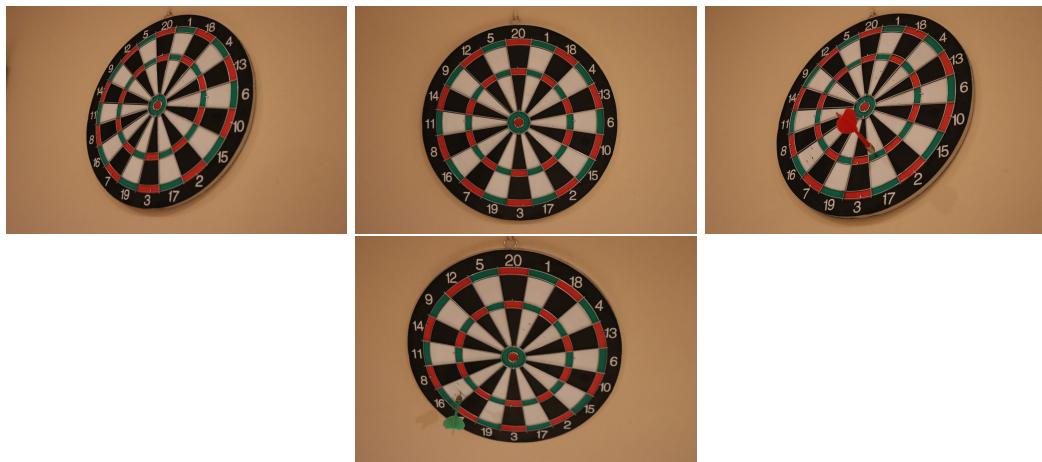


Figura 13: primeras imágenes de la diana

Los resultados obtenidos para estas imágenes fueron:

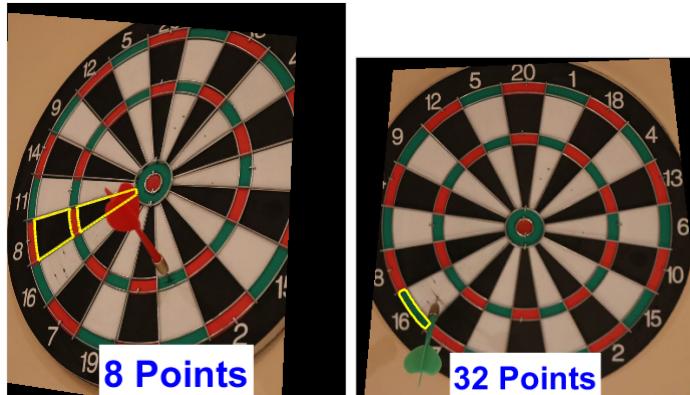


Figura 14: resultados de las primeras imágenes sin pintar los dardos

Como solución a esto decidimos pintar los dardos de color azul. Una vez pintados realizamos otra batería de imágenes.



Figura 15: imágenes de la diana con el dardo azul

Las soluciones obtenidas para estos experimentos fueron:

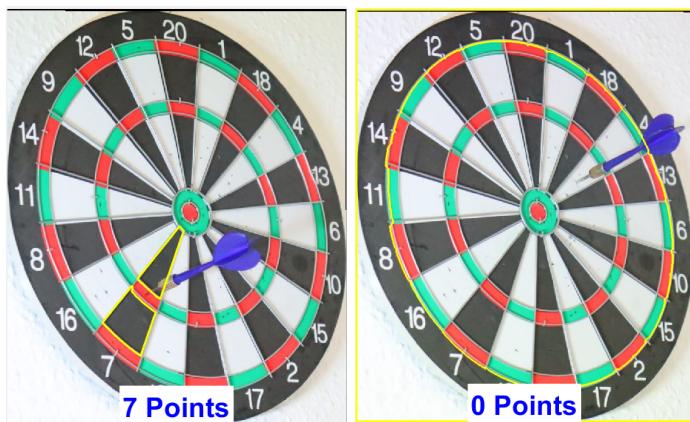


Figura 16: resultados obtenidos para las imágenes de los dardos pintados enteros menos la punta

Los resultados no eran buenos y no sabíamos muy bien a que se debía esto. Pensamos que podía ser porque la punta del dardo no era de color azul y no conseguía detectar bien donde había caido el dardo. Entonces decidimos pintarlo entero.



Figura 17: imágenes de la diana con el dardo entero azul

Los resultados no fueron mucho mejores que los de los experimentos anteriores:

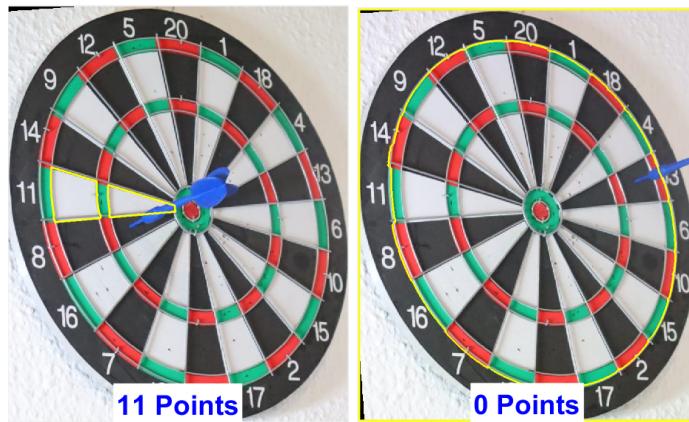


Figura 18: resultados obtenidos para las imágenes de los dardos pintados enteros

Después de que ninguno de los experimentos anteriores funcionase decidimos probar con un nuevo algoritmo de detección del dardo, basado en la detección del color azul. Con este nuevo algoritmo los resultados que obtuvimos fueron finalmente satisfactorios.

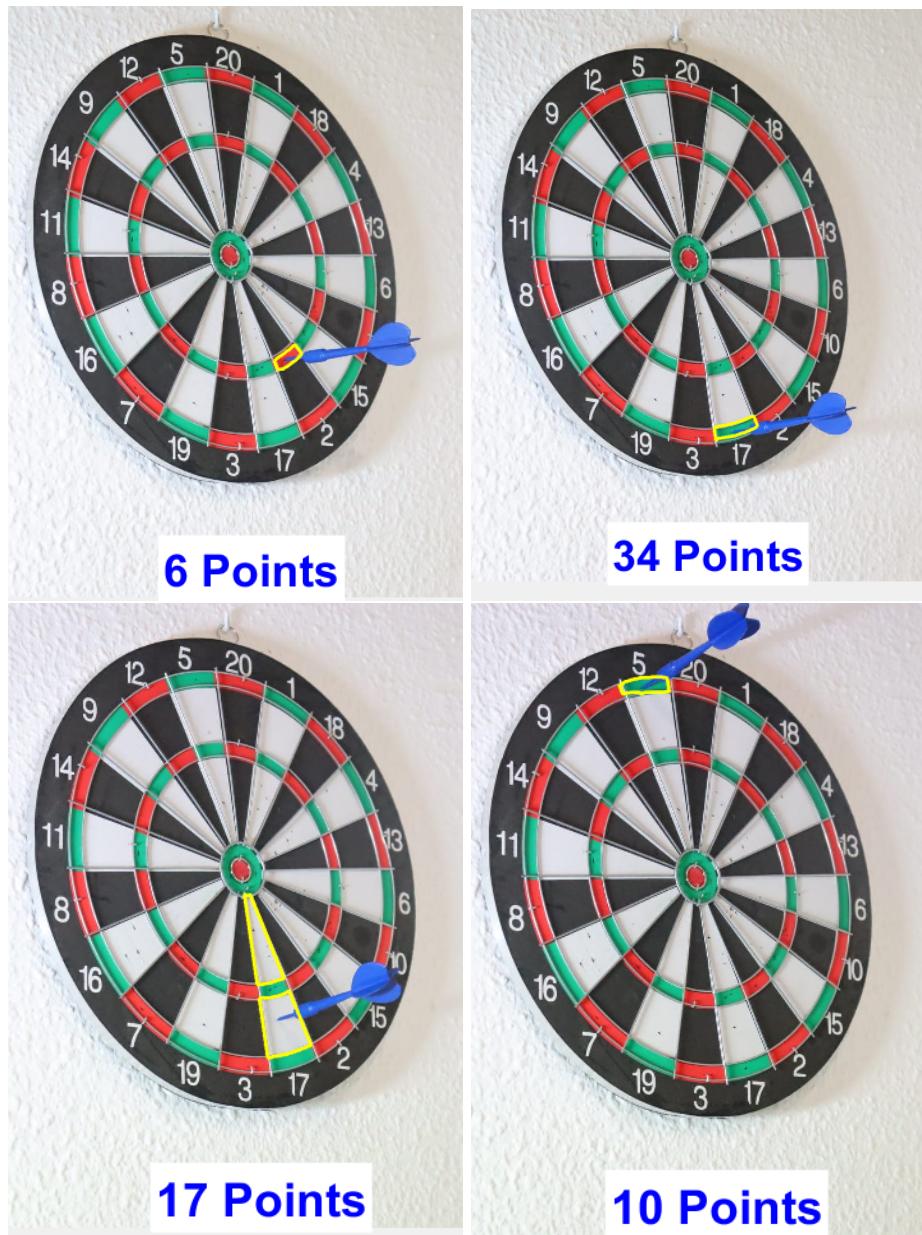


Figura 19: Pruebas con el nuevo algoritmo que detecta el color azul

Como se puede notar, los resultados han sido muy satisfactorios, y el algoritmo siempre detecta la posición y la puntuación correctamente. Pero teniendo ahora este nuevo algoritmo que funciona correctamente decidimos hacer experimentos un poco diferentes para ver como funcionaba desde diferentes ángulos y perspectivas.



Figura 20: Pruebas con el nuevo algoritmo que detecta el color azul mirando desde el lado derecho



Figura 21: Pruebas con el nuevo algoritmo que detecta el color azul mirando desde el frente

5. Manual de usuario

1. Descargar el .zip y descomprimirlo.
2. Una vez descomprimido, veremos que dentro de la carpeta código mejorado existe otra carpeta llamada Dardos.
3. Dentro de Dardos si usted tiene Matlab ya instalado en el ordenador deberá abrir la carpeta "fortestingz" ejecutar el Dardos.exe, mientras que si no tiene Matlab instalado deberá abrir la carpeta "forredistributionz" ejecutar MyAppInstallermcr.exe.
4. Si ya teníamos instalado Matlab el programa se ejecutará correctamente, mientras que si hemos elegido la opción del instalador tendremos que esperar a que se realice una pequeña instalación. NOTA: Cuando elijamos la carpeta de instalación no deben existir caracteres no ingleses en la dirección. ¡También IMPORTANTE marcar la opción Add a shortcut to the desktop! Después de realizar la instalación al abrir el ícono que ha aparecido de la aplicación esta se lanzará.
5. Una vez lanzada la aplicación se abrirá el explorador de archivos donde tendremos que elegir una imagen de una diana sin dardo de las que se adjuntan en la carpeta. Después se volverá a abrir el explorador de archivos y hay que elegir una imagen con dardo.
6. Finalmente aparecerá en pantalla el resultado (no suele tardar mucho pero alomejor la primera vez que se ejecute tarda un poco más).

6. Conclusiones

Las primeras experimentaciones no produjeron los resultados esperados; el primer algoritmo elegido para la detección del dardo no era satisfactorio y todas las pruebas han dado resultados negativos, aunque la solución que normalmente obtuvimos solía estar cerca de donde caía el dardo o alguna parte que lo tapa. Después de esta primera parte de experimentación nuestros objetivos eran:

- 1. Mejorar el algoritmo de detección del dardo.
- 2. Conseguir realizar una cantidad aceptable de experimentos satisfactorios.
- 3. Realizar un análisis de estos experimentos.
- 4. Mejorar la documentación con estos nuevos experimentos.

Después de haber encontrado e implementado el nuevo algoritmo, que detecta el dardo basándose en el hecho y que está pintado de azul, los resultados fueron mucho más satisfactorios y cumplen los objetivos iniciales.

En conclusión podemos decir que estamos aprendiendo mucho durante la realización de este trabajo, y que los errores encontrados al principio nos han ayudado a comprender más en profundidad tanto el funcionamiento de la herramienta MATLAB, como los algoritmos y las técnicas de procesamiento utilizadas.

7. Autoevaluación

Criterios	Matilde Ghidini	Pablo Quindós de la Riva
Comprensión y dominio del tema	0.75	0.75
Exposición didáctica	0.65	0.65
Integración del equipo	0.40	0.40
Objetivos	0.65	0.65
Aspectos didácticos	0.65	0.65
Trabajo reutilizable	0.75	0.75
Experimentación y conclusiones	0.75	0.75
Contenidos	0.75	0.75
Divulgación de los contenidos	0.65	0.65
Organización de la documentación	0.70	0.70
Bibliografía y recursos	0.40	0.40

8. Tabla de tiempos

Fecha de la actividad	Tiempo (h)	Miembros	Actividad realizada
28/03	2h	Matilde	Busqueda articulo
28/03	2h	Pablo	Busqueda articulo y propuesta TD
29/03	2h	Pablo	Busqueda articulo y propuesta TD
29/03	2h	Matilde	Busqueda articulo y propuesta TD
4/04	2h	Matilde	Cuestionario TDA
4/04	2h	Pablo	Cuestionario TDA
5/04	1h	Matilde	Cuestionario TDA
5/04	2h	Pablo	Cuestionario TDA
5/04	1h	Matilde	Planificacion trabajo
14/04	1h	Matilde	Creacion proyecto MATLAB y configuarcion entorno
15/04	4h	Matilde	Estudio del articulo
15/04	4h	Pablo	Estudio del articulo
18/04	2h	Matilde	Estudio teórico sobre la transformada de Hough
18/04	2h	Pablo	Estudio teórico sobre la transformada de Hough
25/04	2h	Matilde	Algoritmo de segmentacion de la diana
25/04	2h	Pablo	Algoritmo de segmentación de la diana
26/04	2h	Matilde	Pruebas del algoritmo de segmentacion de la diana
26/04	2h	Pablo	Pruebas del algoritmo de segmentación de la diana
9/05	1h	Matilde	Estudio del articulo sobre el algoritmo de puntuacion
9/05	4h	Matilde	Algoritmo de puntuacion
17/05	2h	Pablo	Pruebas del algoritmo de segmentación de la diana
23/05	1h	Matilde	Revision de objetivos
23/05	1h	Pablo	Revisión de objetivos
24/05	2h	Matilde	Documentacion
24/05	2h	Pablo	Documentacion
24/05	1h 30min	Pablo	Realización primera bateria de fotos
25/05	2h	Matilde	Documentacion
25/05	2h	Pablo	Documentacion
27/05	30min	Matilde	Tutoria con la profesora
27/05	30min	Pablo	Tutoria con la profesora
27/05	30min	Matilde	Revision horas de trabajo y reorganizacion
27/05	30min	Pablo	Revisión horas de trabajo y reorganización
27/05	2h	Matilde	Documentacion
27/05	2h	Pablo	Detección del dardo

Fecha de la actividad	Tiempo (h)	Miembros	Actividad realizada
28/05 28/05 28/05 28/05	2h 1h 30min 4h 2h	Matilde Pablo Matilde Pablo	Algoritmo detección del dardo Realización de la segunda batería de fotos Pruebas algoritmo detección del dardo Pruebas algoritmo detección del dardo
29/05 29/05 29/05	2h 2h 2h	Matilde Matilde Pablo	Documentación Pruebas y experimentación Pruebas y experimentación
30/05 30/05 30/05 30/05 30/05 30/05	3h 2h 5h 30min 1h 30min 2h	Matilde Pablo Matilde Pablo Pablo Pablo	Documentación Documentación Presentación Presentación Realización de la tercera batería de fotos Pruebas y experimentación
31/05 31/05 31/05 31/05	4h 3h 30min 3h 3h	Matilde Pablo Matilde Pablo	Documentación Documentación Revisión final Revisión final
22/06	3h	Pablo	Desarrollo del nuevo algoritmo de detección del dardo
24/06 24/06 24/06 24/06	4h 3h 5h 3h	Matilde Matilde Pablo Pablo	Estudio del nuevo algoritmo de detección del dardo del dardo Documentación Desarrollo del nuevo algoritmo de detección del dardo Documentación
27/06 27/06 27/06	3h 1h 4h	Matilde Pablo Pablo	Documentación Más fotos Desarrollo del nuevo algoritmo de detección del dardo
28/06 28/06	3h 2h	Matilde Pablo	Documentación Documentación
29/06	3h	Pablo	Documentación
30/06 30/06 30/06	3h 2h 2h	Matilde Pablo Pablo	Documentación Sacar el ejecutable y probar que todo funcione correctamente Documentación

Tiempo total Matilde: 73 horas

Tiempo total Pablo: 74,5 horas

Referencias

- [1] Delaney, Jacob D. *Method to Identify and Score Darts thrown into Dartboard*, 2015.
- [2] R.C. González, R.E. Woods, *Digital Image Processing*, 4th Edition, Pearson, 2018.
- [3] MATLAB documentation, <https://www.mathworks.com/help/matlab/>.
- [4] How to detect blue spot in an image, <https://it.mathworks.com/matlabcentral/answers/488890-how-to-detect-blue-spot-in-an-image>.