Detectar y medir objetos circulares en una imagen

Este ejemplo muestra cómo detectar automáticamente círculos u objetos circulares en una imagen y visualizar los círculos detectados.

Probar este ejemplo

Copy Command

Paso 1: Cargar una imagen

Lea y muestre una imagen de fichas de plástico redondas de varios colores. Además de tener muchos círculos que detectar, esta imagen presenta algunos elementos interesantes desde el punto de vista de la detección de círculos:

1. Hay fichas de distintos colores, que tienen contrastes diferentes con respecto al fondo. Por un lado, las azules y las rojas tienen un contraste fuerte en este fondo. Por el otro lado, algunas de las fichas amarillas no contrastan bien con el fondo.
2. Observe cómo algunas de las fichas están unas sobre otras y cómo otras están cerca y casi se tocan entre ellas. Los límites de objetos que se solapan y la oclusión de objetos suelen ser situaciones difíciles para la detección de objetos.

```
rgb = imread("coloredChips.png");
imshow(rgb)
```

Get

Paso 2: Determinar el intervalo del radio para buscar círculos

Encuentre el intervalo del radio adecuado de los círculos utilizando la función `drawline`. Dibuje una recta sobre el diámetro aproximado de una ficha.

```
d = drawline;
```

Get

La longitud de la ROI de la recta es el diámetro de la ficha. Las fichas comunes tienen diámetros del intervalo de 40 a 50 píxeles.

```
pos = d.Position;
diffPos = diff(pos);
diameter = hypot(diffPos(1),diffPos(2))
```

Get

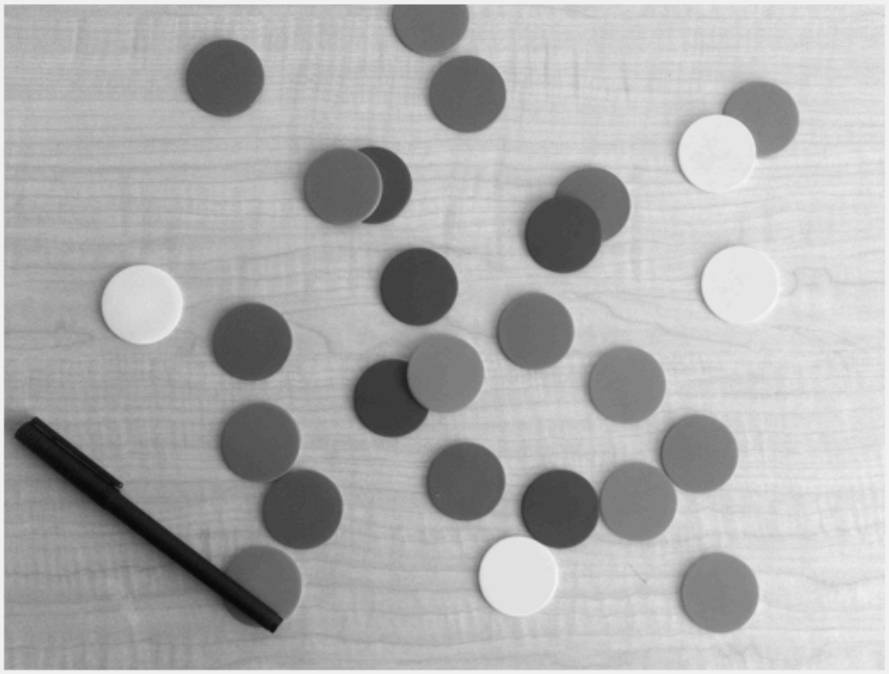
```
diameter = 45
```

Paso 3: Intento inicial de encontrar círculos

La función `imfindcircles` busca círculos con un intervalo de radios. Busque círculos con radios en el intervalo de 20 a 25 píxeles. Antes de eso, una buena práctica consiste en preguntarse si los objetos son más claros o más oscuros que el fondo. Para responder esa pregunta, observe la versión en escala de grises de esta imagen.

```
gray_image = im2gray(rgb);
imshow(gray_image)
```

Get



El fondo es bastante claro y la mayoría de las fichas son más oscuras que el fondo. Sin embargo, de forma predeterminada, `imfindcircles` encuentra objetos circulares que son más claros que el fondo. Por lo tanto, especifique el argumento nombre-valor `ObjectPolarity` como `"dark"` en `imfindcircles` para buscar círculos oscuros.

```
[centers,radii] = imfindcircles(rgb,[20 25],ObjectPolarity="dark")
```

Get

```
centers =

     []


radii =

     []
```

Observe que las salidas `centers` y `radii` están vacías, lo que significa que no se ha encontrado ningún círculo. Esto sucede con frecuencia porque `imfindcircles` es un *detector* de círculos y, de forma similar a la mayoría de los detectores, `imfindcircles` tiene un *umbral de detección* interno que determina su sensibilidad. Dicho de forma sencilla, esto significa que la confianza del detector en una determinada detección (de círculos) debe ser mayor que un nivel determinado antes de considerarse una detección *válida*. `imfindcircles` tiene un argumento nombre-valor `Sensitivity` que se puede usar para controlar este umbral interno y, en consecuencia, la sensibilidad del algoritmo. Un valor de `Sensitivity` más alto establece un umbral de detección más bajo y lleva a detectar más círculos. Esto es similar al control de sensibilidad de los detectores de movimiento que se utilizan en sistemas de seguridad doméstica.

Paso 4: Aumentar la sensibilidad de detección

Volviendo a la imagen de las fichas, es posible que, en el nivel de sensibilidad predeterminado, todos los círculos estén por debajo del umbral interno, motivo por el que no se detectó ningún círculo. De forma predeterminada, `Sensitivity`, que es un numero entre 0 y 1, está establecido en 0.85. Aumente `Sensitivity` a 0.9.

```
[centers,radii] = imfindcircles(rgb,[20 25],ObjectPolarity="dark", ...
    Sensitivity=0.9)
```

Get

```
centers = 8×2

  146.1895  198.5824
  328.8132  135.5883
  130.3134   43.8039
  175.2698  297.0583
  312.2831  192.3709
  327.1316  297.0077
  243.9893  166.4538
  271.5873  280.8920

radii = 8×1

   23.1604
   22.5710
   22.9576
   23.7356
   22.9551
   22.9995
   22.9055
   23.0298
```

En esta ocasión, `imfindcircles` encontró algunos círculos; ocho, para ser precisos. `centers` contiene las ubicaciones de los centros de los círculos y `radii` contiene los radios estimados de esos círculos.

Paso 5: Dibujar los círculos en la imagen

La función `viscircles` se puede usar para dibujar círculos en la imagen. Las variables de salida `centers` y `radii` de `imfindcircles` se pueden pasar directamente a `viscircles`.

```
imshow(rgb)
h = viscircles(centers,radii);
```

Get

Los centros de los círculos parecen estar correctamente posicionados y sus radios correspondientes parecen coincidir bien con las fichas reales. Sin embargo, se han omitido bastantes fichas. Pruebe a aumentar `Sensitivity` aún más, a 0.92.

```
[centers,radii] = imfindcircles(rgb,[20 25],ObjectPolarity="dark", ...
    Sensitivity=0.92);

length(centers)
```

```
ans = 16
```

Cuando aumenta `Sensitivity`, se encuentran aún más círculos. Vuelva a representar esos círculos en la imagen.

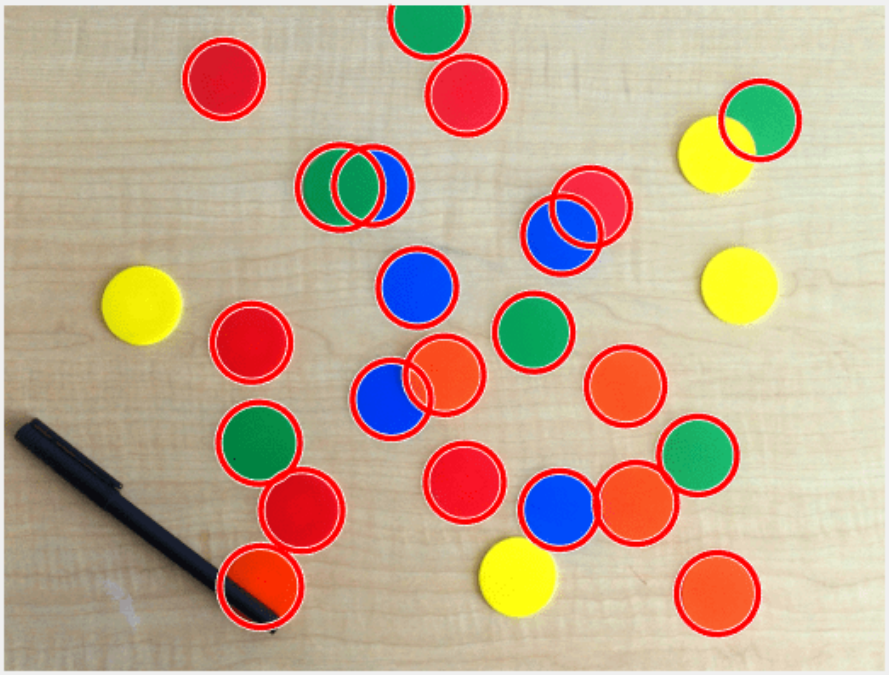
```
delete(h)  % Delete previously drawn circles
h = viscircles(centers,radii);
```

Paso 6: Utilizar el segundo método (de dos fases) para encontrar círculos

Este resultado parece mejor. `imfindcircles` tiene dos métodos diferentes para encontrar círculos. Hasta ahora, se ha utilizado el método predeterminado, conocido como método de *codificación de fase*, para detectar círculos. Hay otro método, conocido generalmente como método de *dos fases*, que está disponible en `imfindcircles`. Utilice el método de dos fases y muestre los resultados.

```
[centers,radii] = imfindcircles(rgb,[20 25],ObjectPolarity="dark", ...
    Sensitivity=0.92,Method="twostage");

delete(h)
h = viscircles(centers,radii);
```

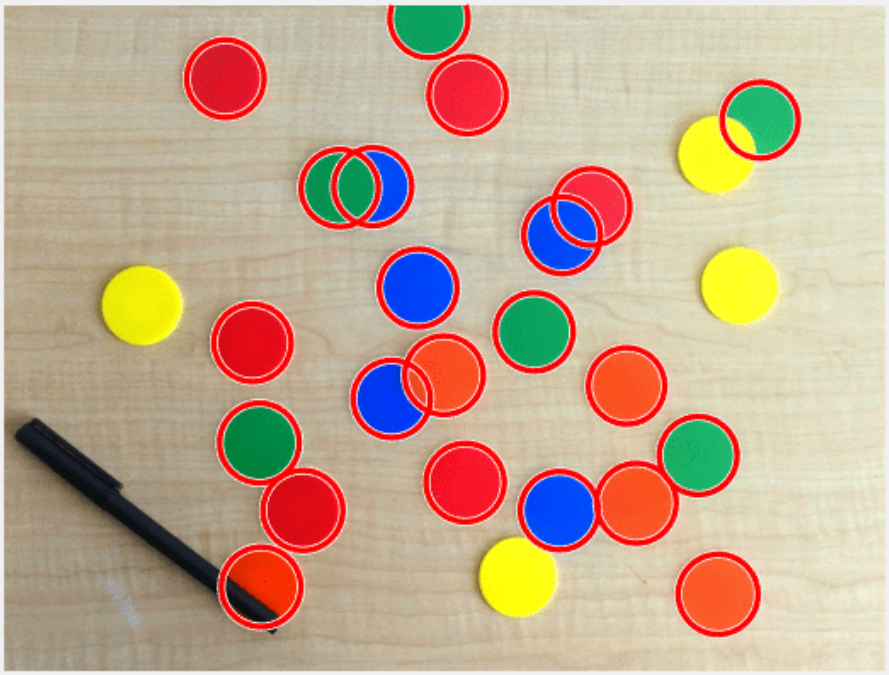


El método de dos fases detecta más círculos con una `Sensitivity` de 0.92. En general, estos dos métodos son complementarios, ya que tienen distintos puntos fuertes. El método de codificación de fase suele ser más rápido y ligeramente más resistente al ruido que el método de dos fases. Sin embargo, también puede necesitar niveles de `Sensitivity` más altos para obtener el mismo número de detecciones que el método de dos fases. Así, el método de codificación de fase también encuentra las mismas fichas si el nivel de `Sensitivity` se eleva más, por ejemplo, a 0.95.

```
[centers,radii] = imfindcircles(rgb,[20 25],ObjectPolarity="dark", ...
          Sensitivity=0.95);

delete(h)
viscircles(centers,radii);
```

Get



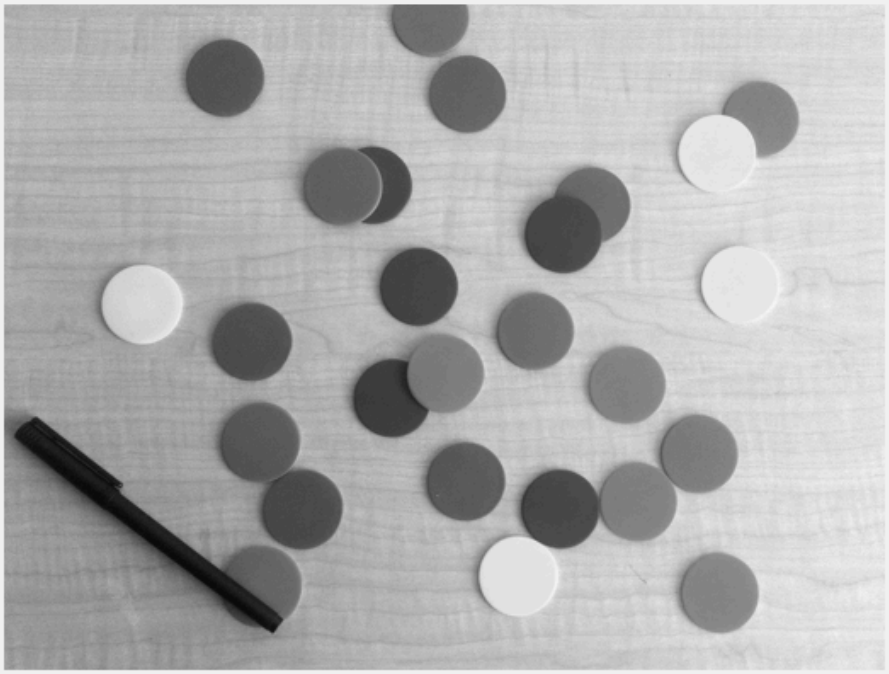
Observe que ambos métodos de `imfindcircles` encuentran con precisión los centros y los radios de las fichas parcialmente visibles (ocultas).

Paso 7: ¿Por qué se siguen omitiendo algunos círculos?

Observando el último resultado, resulta curioso que `imfindcircles` no encuentra las fichas amarillas de la imagen. Las fichas amarillas no tienen un contraste fuerte con el fondo. De hecho, parecen tener intensidades muy similares a las del fondo. ¿Es posible que las fichas amarillas no sean realmente más oscuras que el fondo, como se suponía? Para confirmarlo, muestre de nuevo la versión en escala de grises de esta imagen.

```
imshow(gray_image)
```

Get



Paso 8: Encontrar los círculos claros de la imagen

Las fichas amarillas tienen casi la misma intensidad que el fondo, tal vez sean incluso más claras. Por lo tanto, para detectar las fichas amarillas, cambie `ObjectPolarity` a `"bright"`.

```
[centersBright,radiiBright] = imfindcircles(rgb,[20 25], ...
    ObjectPolarity="bright",Sensitivity=0.92);
```

Get

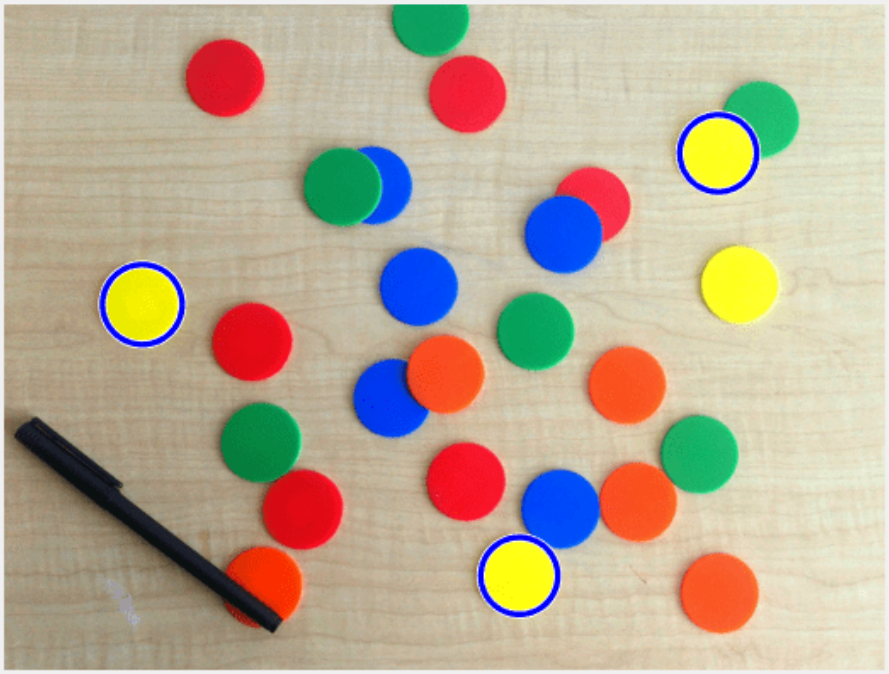
Paso 9: Dibujar los círculos claros con un color diferente

Dibuje los círculos claros en un color diferente cambiando el argumento nombre-valor `Color` en `viscircles`.

```
imshow(rgb)

hBright = viscircles(centersBright,radiiBright,Color="b");
```

Get



Observe que se han encontrado tres de las fichas amarillas faltantes, pero que sigue faltando una ficha amarilla. Estas fichas amarillas resultan difíciles de encontrar porque no destacan tan bien como otras en este fondo.

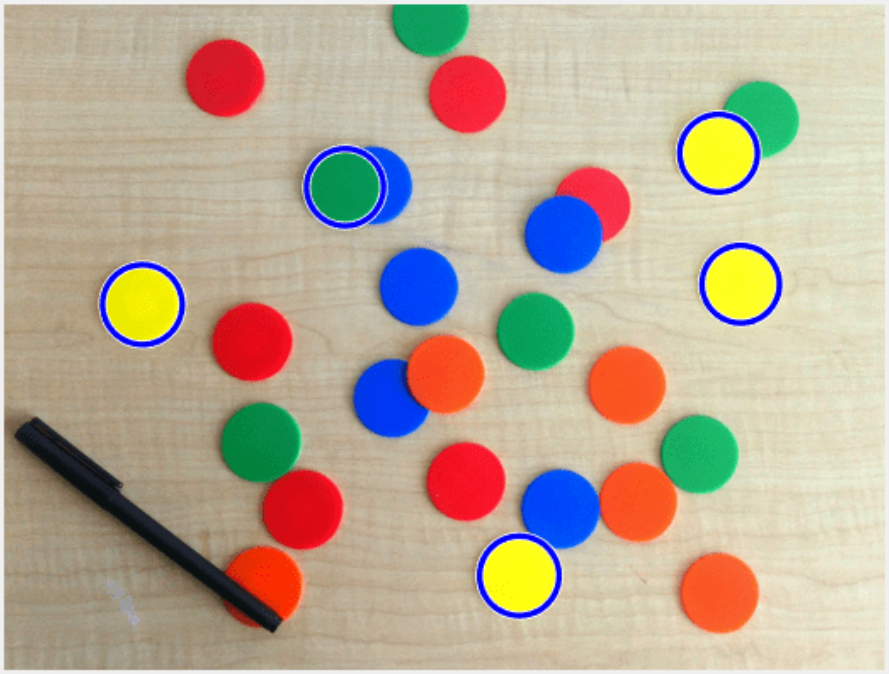
Paso 10: Reducir el valor de `EdgeThreshold`

Hay otro parámetro en `imfindcircles` que puede resultar útil en este caso, como `EdgeThreshold`. Para encontrar círculos, `imfindcircles` solo utiliza los píxeles de borde de la imagen. Estos píxeles de borde son, en esencia, píxeles con un valor de gradiente alto. El argumento nombre-valor `EdgeThreshold` controla cuán *alto* debe ser el valor de gradiente en un píxel para poder considerarse un píxel de borde e incluirse en el cálculo. Un valor alto (más cercano a 1) para este parámetro solo permitirá que se incluyan los bordes sólidos (valores de gradiente más altos), mientras que un valor bajo (más cercano a 0) es más permisivo e incluye incluso los bordes más débiles (valores de gradiente más bajos) en el cálculo. En el caso de la ficha amarilla que falta, puesto que el contraste es bajo, se espera que algunos de los píxeles de límites (en la circunferencia de la ficha) tengan valores de gradiente bajos. Por lo tanto, reduzca el valor `EdgeThreshold` para asegurarse de que la mayoría de los píxeles de borde de la ficha amarilla se incluye en el cálculo.

```
[centersBright,radiiBright,metricBright] = imfindcircles(rgb,[20 25], ...
    ObjectPolarity="bright",Sensitivity=0.92,EdgeThreshold=0.1);

delete(hBright)
hBright = viscircles(centersBright,radiiBright,Color="b");
```

Get

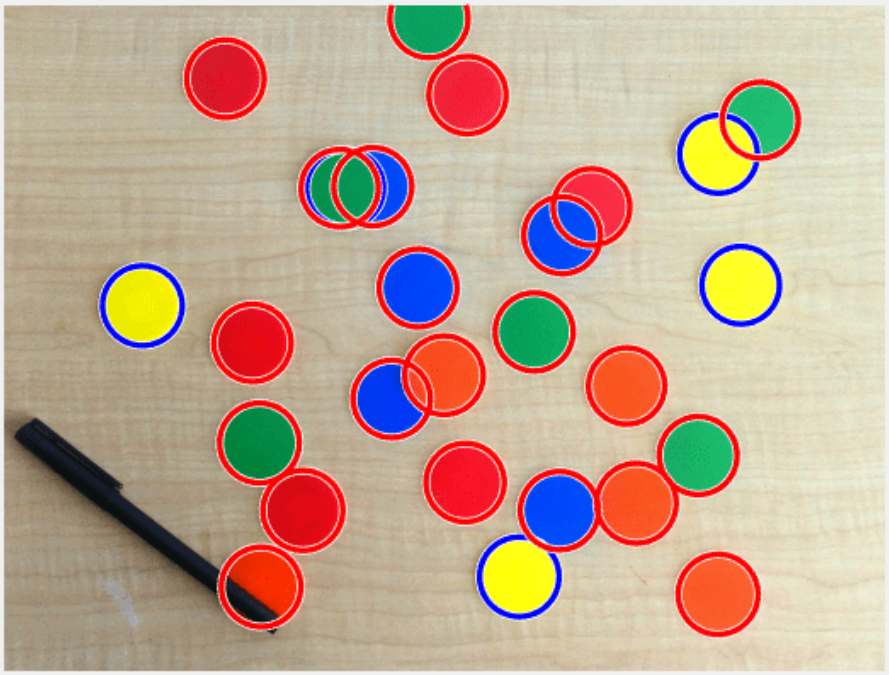


Paso 11: Dibujar juntos los círculos oscuros y claros

Ahora, `imfindcircles` encuentra todas las fichas amarillas y una verde. Dibuje estas fichas en azul, junto con las otras fichas que se encontraron antes (con `ObjectPolarity` establecido en `"dark"`) en rojo.

```
h = viscircles(centers,radii);
```

Get



Se detectan todos los círculos. Una última indicación: debe observarse que, al cambiar los parámetros para que sean más agresivos en la detección, se pueden encontrar más círculos, pero también aumenta la probabilidad de detectar círculos falsos. Hay una compensación entre el número de círculos reales que se puede encontrar (tasa de detección) y el número de círculos falsos que se encuentra con ellos (tasa de falsas alarmas).

Disfrute encontrando círculos.

Consulte también

`imfindcircles` | `circles2mask` | `viscircles`

Temas relacionados

- Identificar objetos redondos
- Medir el radio de un rollo de cinta