

SUSAN Corner Detector

Germán Andrés Jején Cortés

September 2019

1. Introducción

Este documento resume el funcionamiento del detector de bordes y esquinas de SUSAN. Esta nueva aproximación representa una salida representativa de los métodos anteriores.

2. El principio de SUSAN para extracción de características

Si se considera la Figura 1, en donde se muestra un rectángulo oscuro en un fondo blanco,. Una mascara circular (con un pixel central conocido como nucleo) es mostrada en 5 posiciones diferentes. Si el brillo de cada pixel dentro de la mascara es comparado con el brillo del pixel nucleo entonces se define un area de mascara el cual tiene el mismo o similar brillo que el nucleo, esta area es conocida como el USAN "Univalue Segment Assimilating Nucleous". En la Figura 2, cada mascara de la Figura 1 es mostrada con su USAN mostrada en blanco.

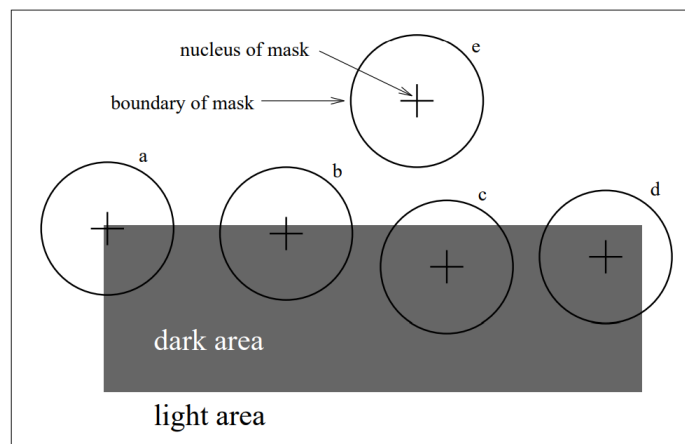


Figura 1: USAN Mascaras

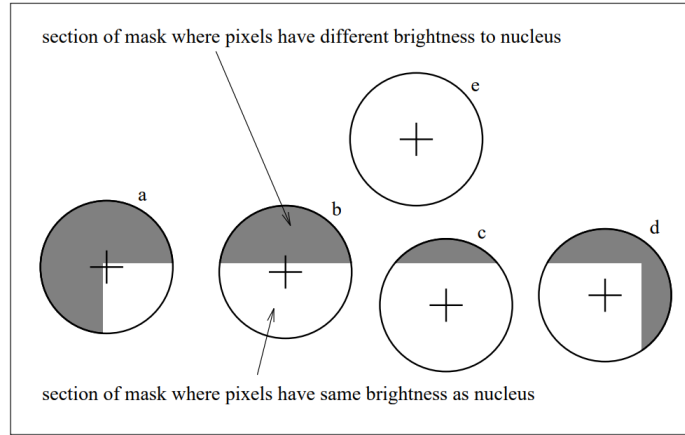


Figura 2: USAN Area Mascaras

El área local o USAN provee bastante información acerca de la estructura de la imagen, a pequeña escala. Esta aproximación de detección de características tiene muchas diferencias respecto a los métodos bien conocidos, la más importante es que no se hace uso de derivadas, por ende no es necesario alguna reducción de ruido.

La mejor métrica en la imágenes es el área USAN acerca de la estructura de la imagen alrededor de un punto en question. Según las Figuras 1 y 2, USAN es máxima cuando el núcleo se encuentra en una región plana y empieza su reducción cuando encuentra un borde y el mínimo cuando existe una esquina. Una mejor percepción de este fenómeno se observa en la Figura 3.

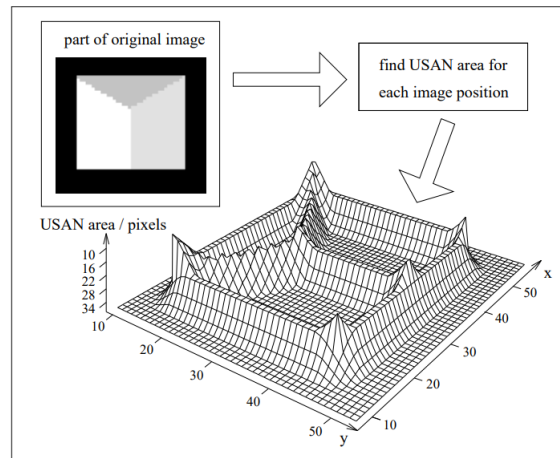


Figura 3: USAN Area 3-D Aproximación

En resumen **Una imagen Procesada genera una salida invertida (Área USAN) que tiene bordes u característica bidimensionales fuertemente mejoradas, con las características bidimensionales mas fuertemente mejoradas que los bordes.**

Esto genera el acronimo para SUSAN (Smallest Univalve Assimilating Nucleus).

El hecho que las mejoras de SUSAN en la detección de bordes y esquinas se explica en el hecho que no hace uso de derivadas, esto genera un muy buen desempeño en presencia de ruido. El efecto integrativo junto a su característica no lineal genera un fuerte rechazo al ruido.

La reducción de ruido de SUSAN está relacionada al principio de SUSAN que radica en que USAN es usado para obtener el mejor suavizado local en el contorno.

3. Ejemplo

El siguiente ejemplo hace uso de la parte más sumple de SUSAN creando una mascara circular de 7 pixeles de ancho y 7 de alto. Para la detección de bordes calcula el area de los pixeles similares al pixel nucleo basandose en la siguiente formula.

$$\sum_{i,j} e^{-(\frac{(VV-VN)^2}{10})^6)} \quad (1)$$

Donde VV es el la matriz de intensidad de cada pixel dentro de la mascara y VN es la intensidad del pixel Nucleo.

3.1. Código

```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt
import math
import matplotlib.cm as cm
```

Figura 4: Librerías Necesarias

```
def susan_mask():
    mask=np.ones((7,7))
    mask[0,0]=0
    mask[0,1]=0
    mask[0,5]=0
    mask[0,6]=0
    mask[1,0]=0
    mask[1,6]=0
    mask[5,0]=0
    mask[5,6]=0
    mask[6,0]=0
    mask[6,1]=0
    mask[6,5]=0
    mask[6,6]=0
    return mask
```

Figura 5: Definición de la mascara

```
def plot_image(image,title):
    plt.figure()

    plt.title(title)
    plt.imshow(image,cmap = 'gray')

    plt.show()
```

Figura 6: Función para graficar

```

def susan_corner_detection(img):
    img = img.astype(np.float64)
    g=37/2
    circularMask=susan_mask()
    output=np.zeros(img.shape)

    for i in range(3,img.shape[0]-3):
        for j in range(3,img.shape[1]-3):
            ir=np.array(img[i-3:i+4, j-3:j+4]) # Captura de la zona de interés dentro de la imagen
            ir = ir[circularMask==1] # Captura solo de los píxeles dentro de la máscara
            ir0 = img[i,j] # Pixel núcleo
            a=np.sum(np.exp(-(((ir-ir0)/10)**6))) # Calculo del area SUSAN respecto al núcleo
            if a<=g: # Comparación con límite
                a=g-a
            else:
                a=0
            output[i,j]=a # Si es una esquina guarda un valor Diferente a 0
    return output

```

Figura 7: Detección de esquina

```

img=cv.imread("susan_input1.png",0)
output1 = susan_corner_detection(img)
finaloutput1 = cv.cvtColor(img, cv.COLOR_GRAY2RGB)
finaloutput1[output1 != 0] = [255, 255, 0]
plot_image(finaloutput1, "Output Part1") # good success

```

Figura 8: Ejecución del Código

3.2. Resultado

Para la siguiente imagen de entrada:

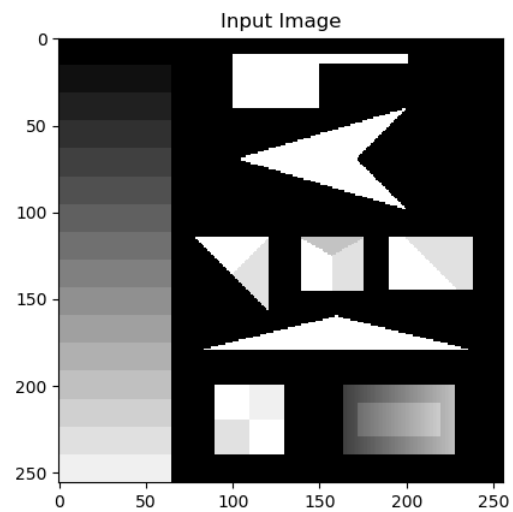


Figura 9: Imagen de Entrada

Se obtiene la siguiente detección

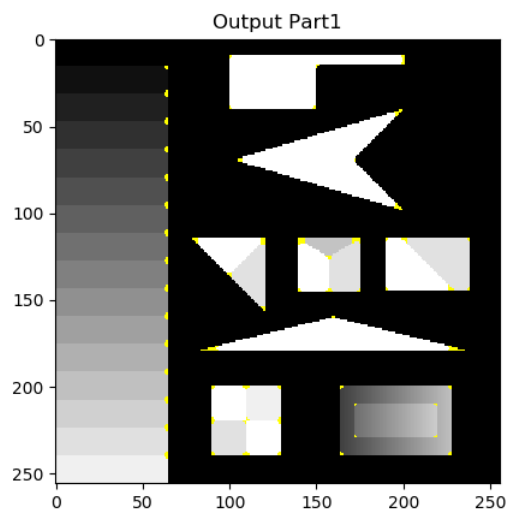


Figura 10: Imagen resultado

4. Conclusión

El algoritmo de SUSAN tiene una visión diferente a los predecesores y un mejor desempeño debido al cambio de paradigma del derivativo al integral. Además que puede ser reforzado haciendo uso de técnicas de suavizado.

5. Referencias

El siguiente documento es completamente basado en el artículo principal de S.M. Smith y J.M. Brady. Disponible en el siguiente enlace

[Link a Artículo original](#)