

Primer Examen Parcial - Visión Computacional

Quistian Navarro, Juan Luis
A341807@alumnos.uaslp.mx

15 de septiembre de 2024

1. Planteamiento del problema

Se cuenta con la imagen "Jorge_Campos.jpg", que ha sido editada para contener ruido que genera un efecto de desenfoque. El objetivo es aplicar una serie de filtros y operaciones proveniente de OpenCV [1] para eliminar el ruido, mejorar la nitidez y rotar la imagen de manera que la silueta de la persona aparezca completamente vertical. Además, se debe recortar la imagen para que solo aparezca la silueta de la persona lo más ceñido posible.

2. Descripción de la solución

El proceso de solución se basa en la aplicación de varios filtros de convolución y Fourier para eliminar el ruido y mejorar la calidad de la imagen. A continuación se detallan los pasos aplicados en el orden correspondiente:

2.1. Filtro Sharpened (Convolución)

El primer filtro aplicado fue un filtro de convolución Sharpened para mejorar la nitidez de los bordes y recuperar detalles de la imagen. Utilizamos un kernel Laplaciano con la función `cv2.filter2D()` de OpenCV.

Kernel aplicado:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Este kernel tiene el efecto de resaltar los bordes y contrastar las áreas circundantes para mejorar la claridad de la imagen.

2.2. Filtro Non-local Means

El siguiente filtro aplicado fue el **Non-local Means** [2], el cual es útil para reducir ruido manteniendo los detalles. Aplicamos este filtro dos veces consecutivas con la función `cv2.fastNlMeansDenoisingColored()`. Este paso fue importante para eliminar los residuos de ruido sin sacrificar la calidad de la imagen.

Parámetros utilizados:

- `h`: 3, que controla la intensidad de la eliminación de ruido.
- `templateWindowSize`: 7, tamaño de la ventana para comparar píxeles.
- `searchWindowSize`: 15, tamaño de la ventana de búsqueda alrededor de los píxeles.

2.3. Filtro de Fourier de paso bajas

Posteriormente se aplicó un filtro de paso bajas [3] utilizando la transformada de Fourier. Este filtro fue necesario para reducir las frecuencias altas que representan el ruido en la imagen. Utilizamos la función `np.fft.fft2()` para calcular la transformada rápida de Fourier, seguido de `np.fft.ifft2()` para la transformada inversa y reconstruir la imagen suavizada.

Los parámetros utilizados fueron:

- `radius`: 120, que define el radio del filtro de pasa bajas.

2.4. Filtro Sharpened adicional

Para aumentar la nitidez tras la eliminación de ruido, aplicamos un filtro Sharpened con un kernel de suave (de convolución), nuevamente usando `cv2.filter2D()`. Repetimos este filtro dos veces para mejorar la nitidez final.

Kernel aplicado fue:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

2.5. Recorte de la imagen

Usamos una función de recorte manual para recortar la imagen a las coordenadas que rodean la silueta de la persona, lo más ceñido posible. La función `crop_image()` realiza el recorte usando las coordenadas especificadas:

- `x=370, y=200, w=320, h=290`

2.6. Rotación de la imagen

Finalmente, rotamos la imagen -35 grados para que la silueta de la persona aparezca en posición completamente vertical. Usamos la función `cv2.warpAffine()` de OpenCV para aplicar la rotación. El ángulo de rotación elegido fue -35 grados para alinear correctamente la figura.

3. Descripción de los resultados

A continuación, se describen los resultados obtenidos en cada paso:

3.1. Filtro Sharpened (primero)

El Filtro Sharpened (figura 1) inicial mejoró significativamente los bordes, haciendo que las partes desenfocadas de la imagen se vuelvan más nítidas y fáciles de identificar.

3.2. Non-local Means

La aplicación de este filtro (figura 2) eliminó el ruido residual de la imagen suavizada. Los detalles fueron preservados y la imagen ahora tiene una mejor relación señal/ruido.

3.3. Filtro de Fourier de pasa bajas

El uso del filtro de pasa bajas (figura 3) en la transformada de Fourier redujo considerablemente el ruido de la imagen. Sin embargo, parte de los detalles más finos se perdieron debido a la suavización.



Figura 1: Sharpened (Kernel medio)



Figura 2: Non local menas

3.4. Filtro Sharpened adicional

La aplicación adicional del Filtro Sharpened dos veces (figura 4) resultó en una imagen más clara y detallada. Los bordes están bien definidos, lo que facilita el reconocimiento de la silueta.

3.5. Recorte de la imagen

El recorte ceñido a la silueta de la persona (figura 5) dejó únicamente al sujeto principal de la imagen, excluyendo cualquier fondo irrelevante.



Figura 3: Low pass filter



Figura 4: Sharpened (kernel suave)

3.6. Rotación de la imagen

La rotación (figura 7) aplicada correctamente posicionó a la persona en una postura completamente vertical, alineando la imagen de acuerdo a los requisitos del examen.

4. Discusión

El proceso de mejora de la imagen a través de la combinación de varios filtros y técnicas permitió alcanzar los objetivos de manera efectiva. Uno de los mayores retos fue equilibrar la eliminación de ruido con la preservación de detalles, pero el uso combinado de los filtros Sharpened, Non-local Means y Fourier proporcionó buenos resultados. La rotación final y el recorte aseguraron que el resultado final cumpliera con las especificaciones del examen.



Figura 5: Recorte imagen final



Figura 6: Recorte imagen original



Figura 7: Rotación

5. Conclusión

El proceso de mejorar una imagen con ruido y desenfoque utilizando OpenCV y técnicas de procesamiento de imágenes como convolución y transformada de Fourier fue satisfactorio. Los resultados muestran que es posible mejorar notablemente la calidad visual de una imagen a través de una adecuada combinación de filtros.

Referencias

- [1] Image filtering. https://docs.opencv.org/4.x/d4/d86/group__imgproc__filter.html. Accessed: 2024-09-15.
- [2] Non-local means denoising. https://docs.opencv.org/3.4/d5/d69/tutorial_py_non_local_means.html. Accessed: 2024-09-15.
- [3] WSTHUB. Python computer vision tutorials — image fourier transform / part 3. <https://wsthub.medium.com/python-computer-vision-tutorials-image-fourier-transform-part-3-e65d10be4492>. Accessed: 2024-09-15.