

PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO DA UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

DISCIPLINA: CMP 197 - Fundamentos de Visão Computacional

PROFESSOR: DR. Claudio Rosito Jung

ALUNOS: Jerson Geraldo Neto e Otávio Secco

DATA: 20 / 01 / 2023

TRABALHO DE IMPLEMENTAÇÃO 1

O problema proposto leva em consideração a comparação entre imagens digitais e fotografias realizadas de imagens impressas, com o objetivo de mostrar os resultados obtidos. Com esse intuito foi fornecido um *dataset* composto pelas imagens dentro das condições estipuladas.

Para que a comparação pudesse ser realizada de maneira adequada alguns problemas tiveram de ser solucionados inicialmente, quais sejam: dimensionar as fotografias com base em seus valores iniciais; recortar a captura com base em coordenadas de pixel; rotacionar para que ambas estivessem com a mesma orientação; e, comparar os resultados quanti e qualitativamente.

O redimensionamento das imagens foi feito de forma a proporcionar ao usuário a manipulação de maneira mais fácil, assim o tamanho das janelas foi reduzido. Para recortar a imagem (*crop*) foi utilizado o clique com o mouse nos *corners* correspondentes dentro da janela, com isso, as coordenadas foram armazenadas para a posterior transformação.

Os *corners* são determinados através da função `getClicks(event, x, y, flags, params)`, responsável por capturar as coordenadas dos pontos clicados na imagem (`getPoints`) e adicionando em uma lista `click_list`, por meio da função `cv2.setMouseCallback ('output', getClicks)`.

Em seguida usasse a função `orderPoints(pts)` para ordenar os pontos na corretamente (superior esquerdo, superior direito, inferior direito, inferior esquerdo), possibilitando que a transformação de perspectiva possa ser aplicada.

A transformação é realizada por meio da função `fourPointTransform (image, pts)`, iniciando a partir do cálculo da largura e altura da imagem resultante, utilizando as diferenças entre as coordenadas dos pontos clicados.

Com isso, crie uma matriz de destino chamada 'dst', que armazena as coordenadas finais dos quatro pontos da imagem após a transformação de perspectiva, que contém informações sobre como os pixels da imagem original

devem ser mapeados para a imagem transformada.

Obtém-se, então a matriz de quatro pontos de dimensão $N \times 2$, sendo que N representa cada pixel da mesma, onde cada linha representa as coordenadas x e y de um ponto na imagem original e a matriz 'dst' corresponde as coordenadas da imagem transformada. Com isso cria-se uma matriz de equações lineares de maneira a possibilitar que a imagem transformada seja reorientada, rotacionando-a.

Isso é feito por meio da função `cv2.warpPerspective(image, m, (maxWidth, maxHeight))`. Ela é usada para aplicar a transformação de perspectiva na imagem usando a matriz m obtida anteriormente. A imagem resultante é redimensionada para o tamanho especificado em `maxWidth` e `maxHeight`.

Para que as comparações sejam realizadas, a imagem transformada e a digital são mostradas lado a lado. Visivelmente algumas características podem ser percebidas de maneira imediata, como o brilho reduzido e a perda aparente de detalhes que ocorre na imagem transformada. Para demonstrar de maneira mais clara os resultados obtidos, primeiramente, buscou-se utilizar um mapa de contornos simples seguido por um filtro de segmentação. Para esse fim, os filtros utilizados foram os detectores de bordas de Canny e o Sober, e, por fim, um filtro de segmentação básico baseado em Otsu.

O detector de bordas de Canny conseguiu diferenciar as imagens de maneira simples, identificando a ausência de alguns contornos entre ambas as imagens, identificando uma provável perda de informações resultante da transformação. Contornos antes mais nítidos dentro da vegetação, por exemplo, não foram reconhecidos.

No caso do detector de Sobel foi possível visualizar que houve uma maior reunião de pixels na imagem transformada, fruto do efeito “blur” da transformação aplicada. Com isso observasse que os contornos base foram preservados de maneira geral, mas há bordas mais densas que outras, bem como detalhes anteriores (como as nuvens no céu e vegetação) que forma distorcidos.

Quando foi aplicada a segmentação foi ainda mais nítida a interferência ocorrida nos contornos da imagem, haja vista que as imagens possuíam alterações nas bordas, principalmente nas resultantes de sombra.

Para verificar os aspectos inerentes à coloração de ambas as imagens foi aplicado um histograma dentro de cada canal RGB que depois foi compilado em um resultado único, tanto para a imagem original quanto para a transformada. O resultado dessa aplicação demonstra de maneira clara a diferença entre ambas,

caracterizada pela redução do número de ondas iniciais, ocasionando a aproximação das cores e a consequente perda de informações que pode ser visualizada na aplicação dos filtros detectores de bordas e na segmentação.

Na abordagem quantitativa dos resultados apresentados no uso da implementação foi utilizado o PSNR (Peak Signal-to-Noise Ratio), essa proporção é usada como uma medida de qualidade entre o original e uma imagem compactada. Quanto maior o PSNR, melhor a qualidade da imagem comprimida ou reconstruída. No caso das imagens transformadas o valor variou sempre no intervalo compreendido entre 10 e 25, o que caracteriza a perda de qualidade ocasionada pela transformação.

O presente trabalho encontra-se disponível no seguinte endereço:
<https://github.com/seccoboy/CMP197-FundamentosdeVisaoComputacional-T1>

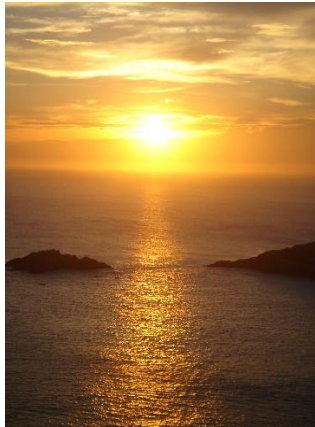
ANEXO A

Figuras: 1 – imagem original; 2 – fotografia; 3 – fotografia transformada

FOTOGRAFIA 1



FOTOGRAFIA 2

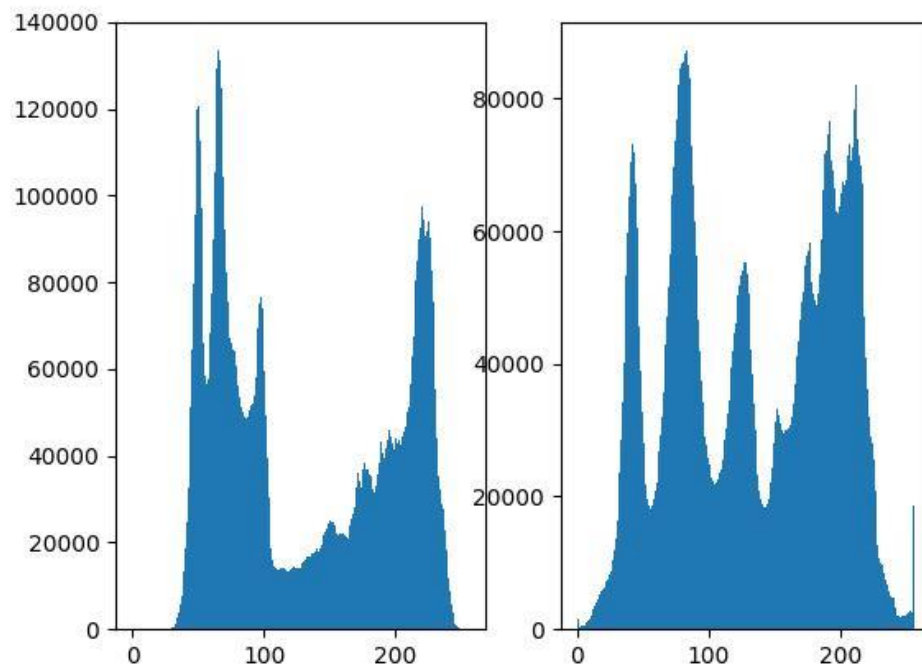
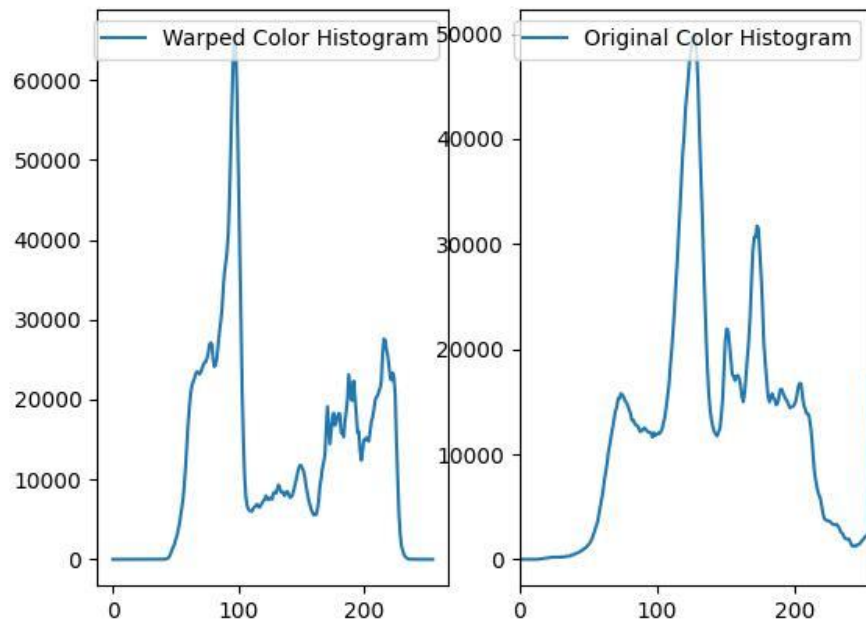


FOTOGRAFIA 3

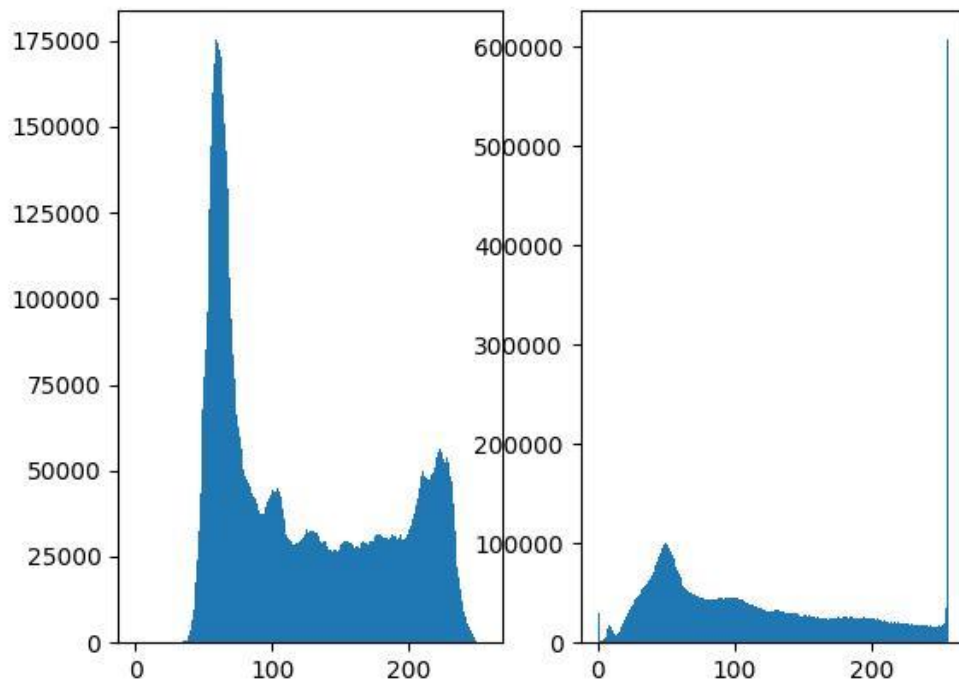
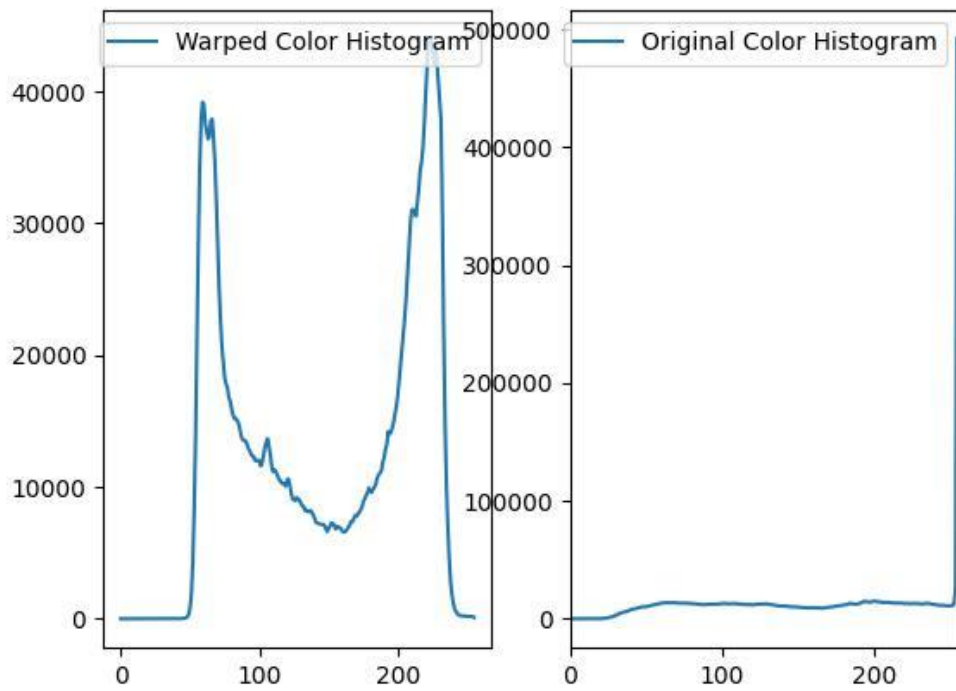


Histograma comparativo entre as imagens originais (direita) e a transformada
(esquerda)

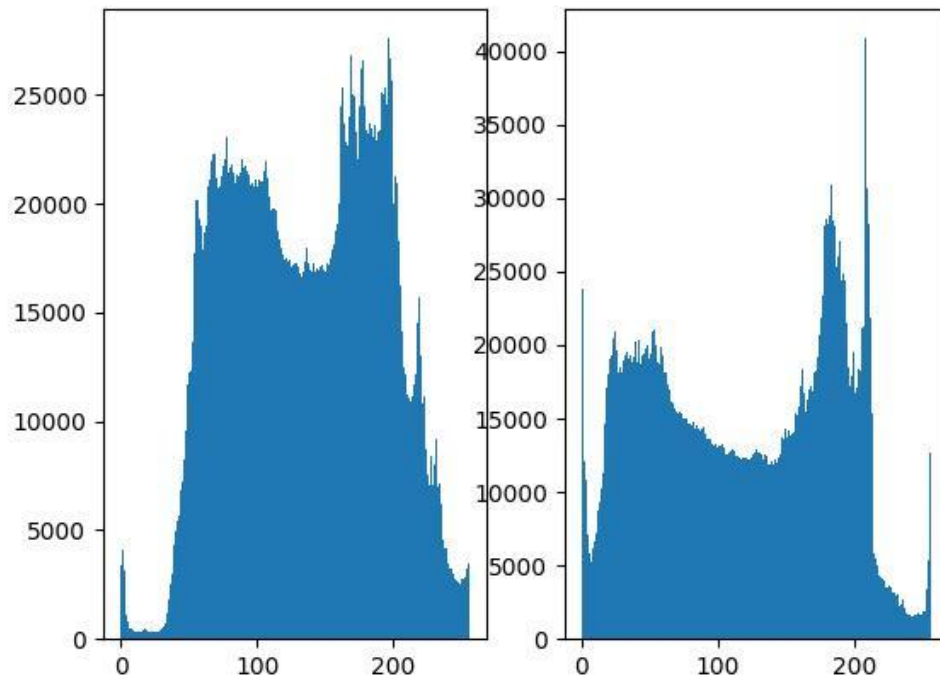
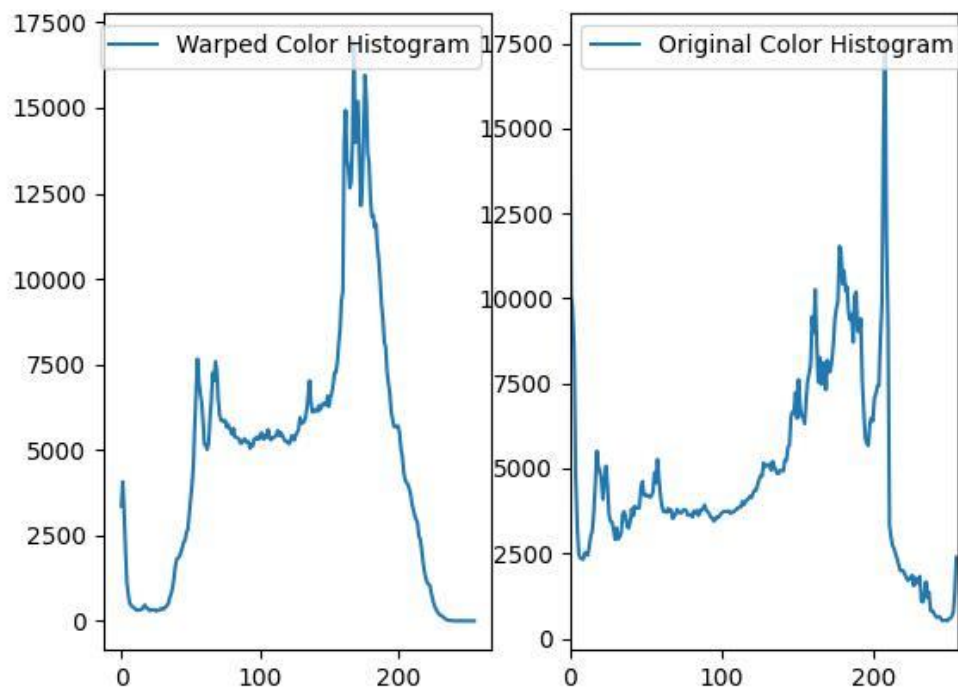
FOTOGRAFIA 1



FOTOGRAFIA 2

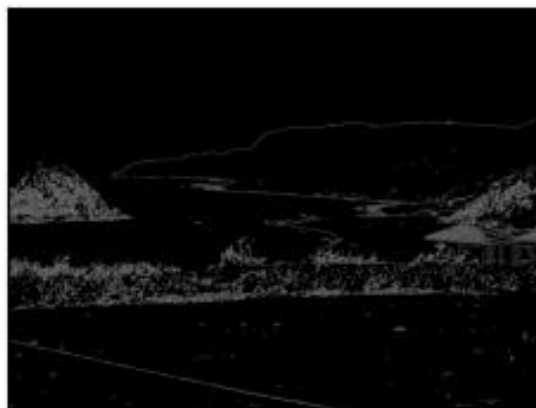


FOTOGRAFIA 3

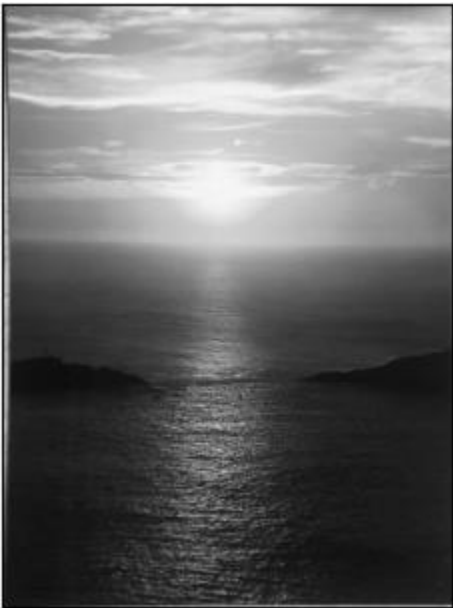


Filtro de bordas de Canny aplicado a imagem original x imagem transformada

FOTOGRAFIA 1



FOTOGRAFIA 2

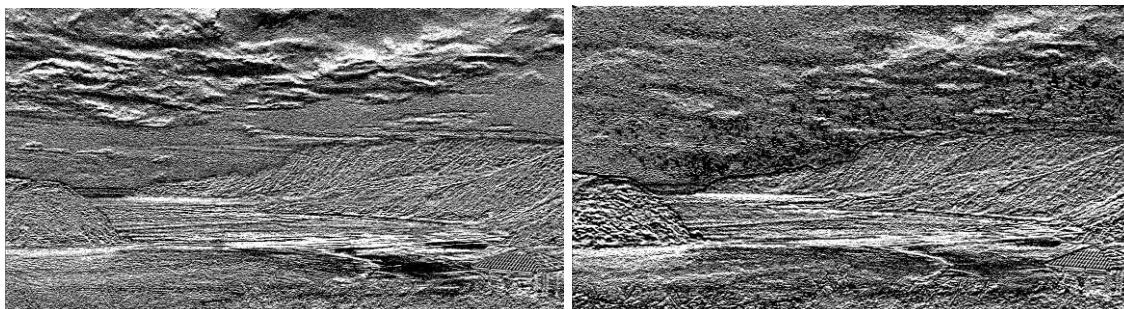


FOTOGRAFIA 3

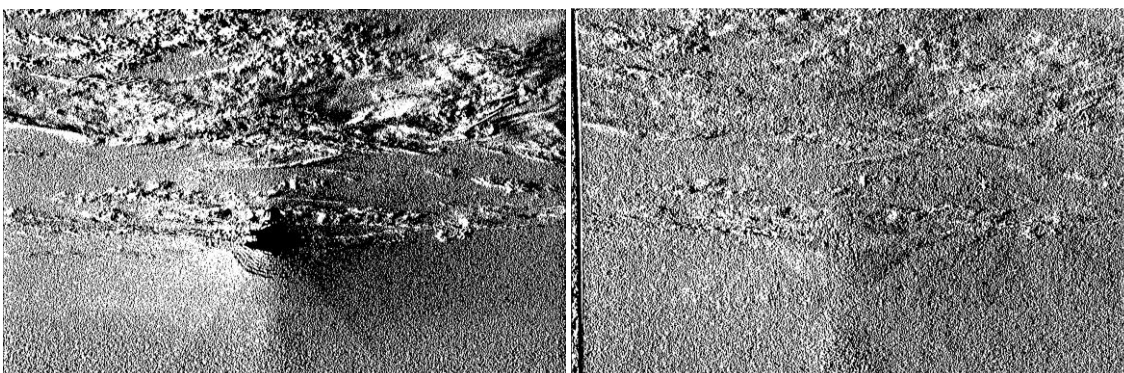


Filtro de bordas de Sobel aplicado a imagem original x imagem transformada
Horizontal

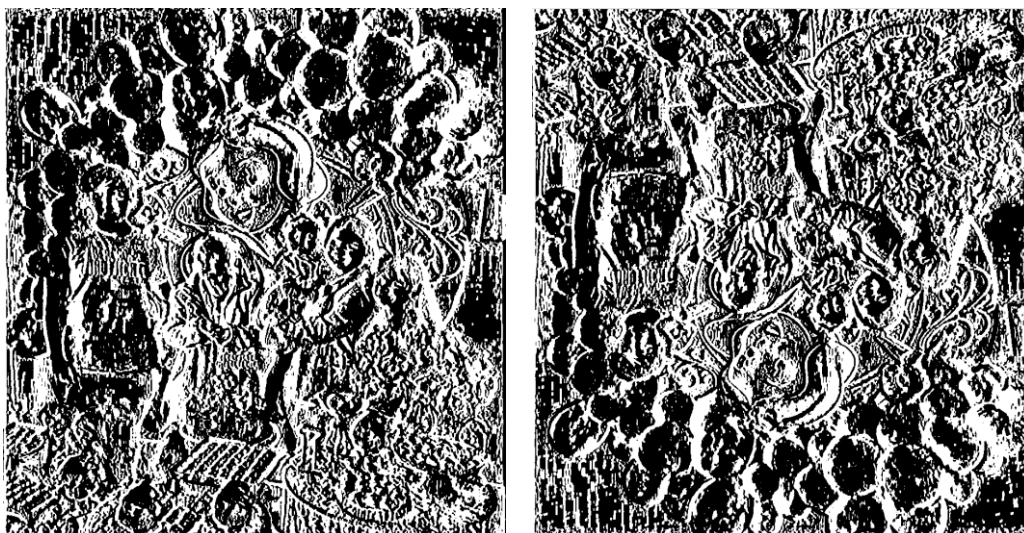
FOTOGRAFIA 1



FOTOGRAFIA 2

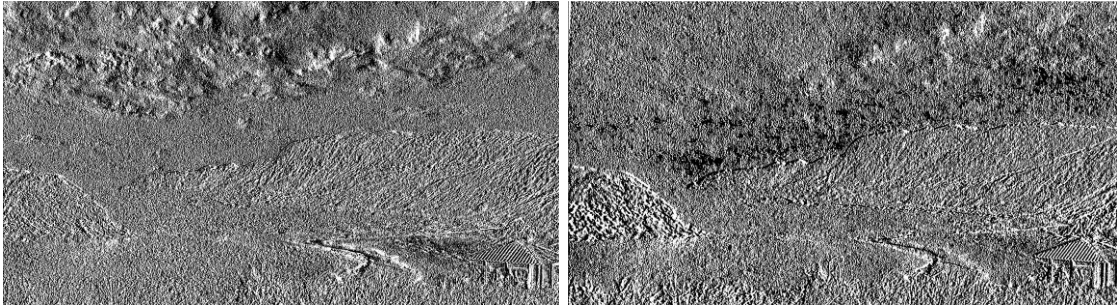


FOTOGRAFIA 3

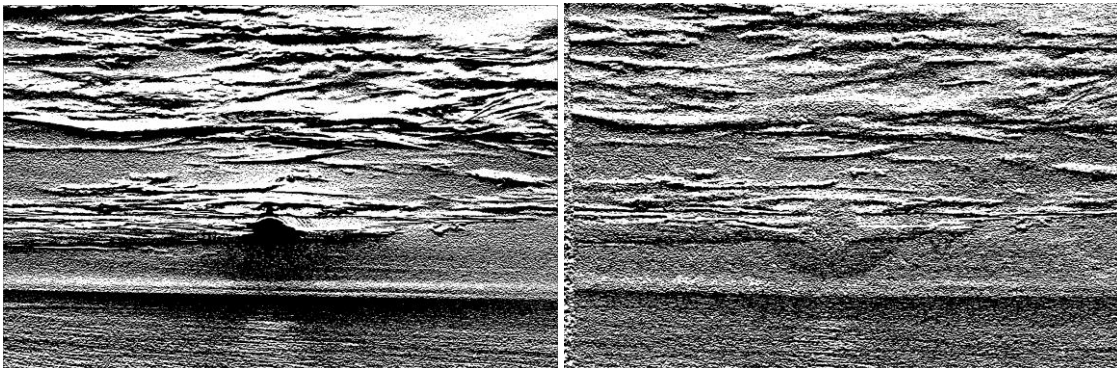


Vertical

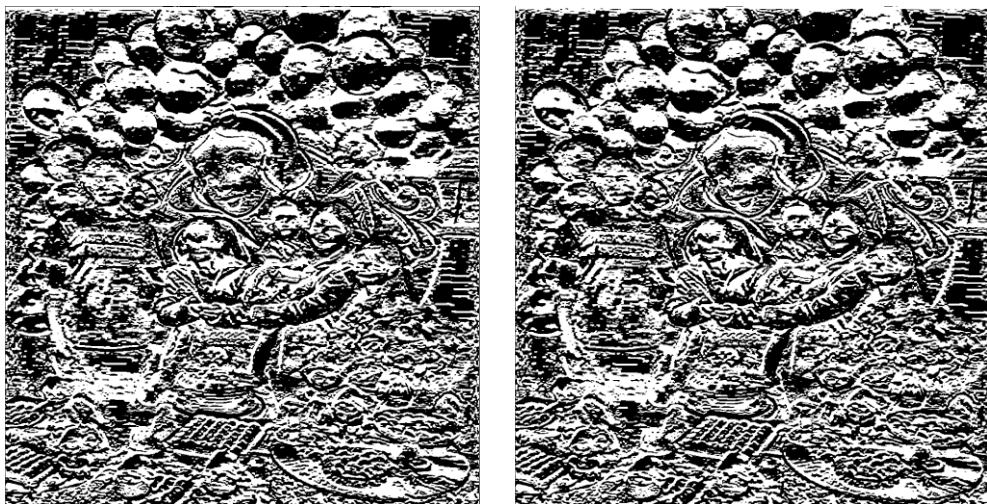
FOTOGRAFIA 1



FOTOGRAFIA 2

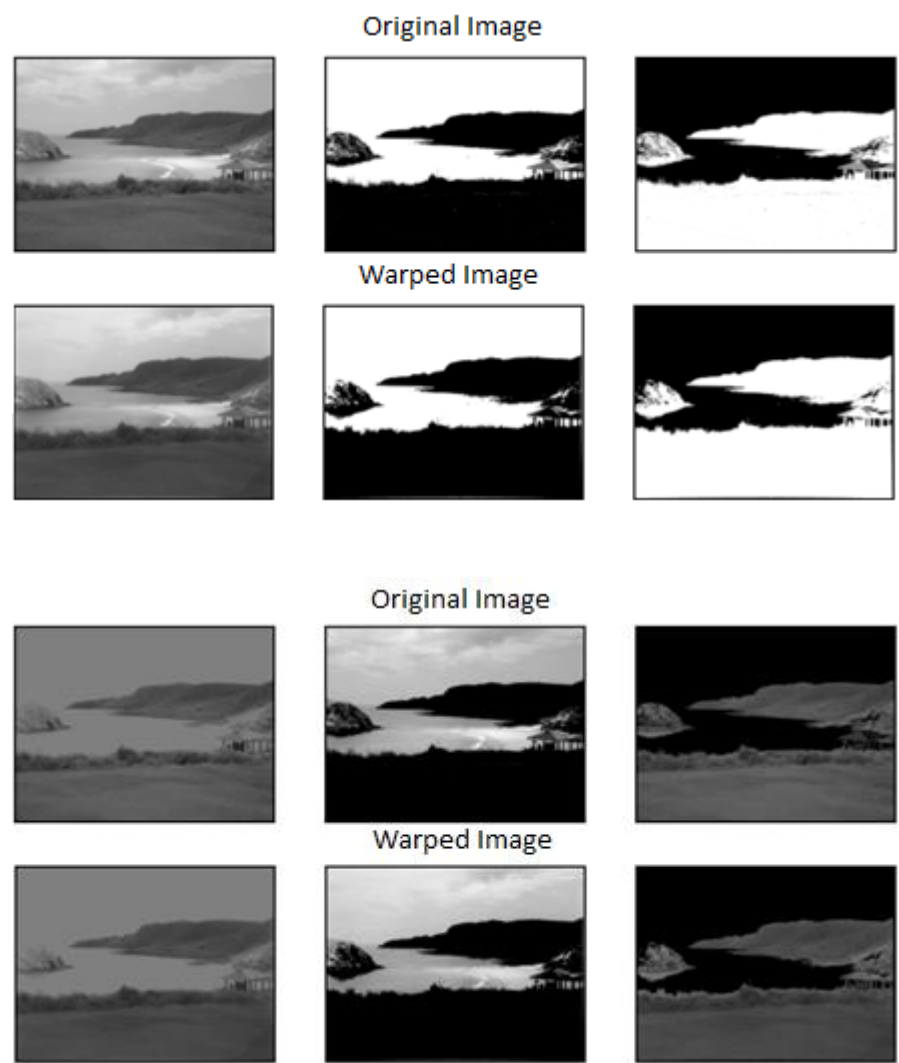


FOTOGRAFIA 3



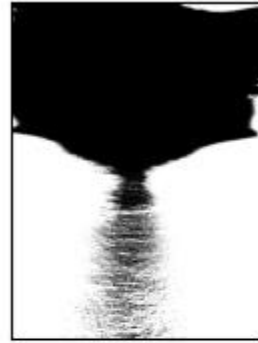
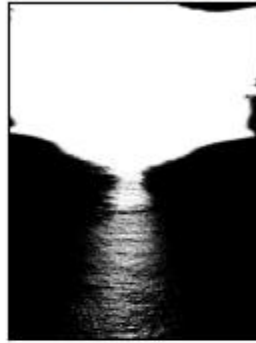
Segmentação aplicada a imagem original x imagem transformada

FOTOGRAFIA 1



FOTOGRAFIA 2

Original Image



Warped Image



Original Image



Warped Image



FOTOGRAFIA 3

Original Image



Warped Image



Original Image



Warped Image

