Féras GHANDOURI - 11601442
M2 ID3D – 2021/2022

TP1 Analyse d'image

Gradients

Figure 1 Amplitude of gradient in X axis

Figure 2 Amplitude of gradient in Y axis

Figure 3 Amplitude of bi-directional gradient

Figure 4 Amplitude of multi-directional gradient

Gradients are computed using a convolution with the appropriate kernel. The bi-directional gradient is the gradient computed with the X and Y directions, the multi-directional gradient is computed with X, Y, North-West ($3\pi/4$), North-East ($\pi/4$).

Thresholding methods

Global thresholding

Explanation
This method sets to 0 the value of the pixels below a user defined threshold, the pixels above are set to 255.

Result

Figure 5 Global thresholding, with high threshold set to 0.2

Observations

I noticed that this method is not accurate as it misses a lot of areas of where the change of intensity is subtle, resulting in an image with less contours, with the majority contours corresponding to areas of high change in intensity.

Local thresholding

Explanation
This method computes with a convolution the mean of the gradients around a pixel, sets to 0 the value of the pixels below the local mean and 255 for the pixels above. The default kernel size is 3x3, computer the mean considering 8 neighbors.

Result

Figure 6 Local threshold, kernel size 11x11

Observations

I noticed that this method finds a lot more contours but fails to separate the background from the foreground as well as the global thresholding.

Hysteresis thresholding

Explanation

This method applies the same logic as a global threshold for the amplitude values strictly less than the low threshold or strictly higher than the high threshold. There is an extra step for the pixels noted P whose gradient amplitudes are in between the thresholds : it checks if at least one of their neighbors' amplitudes are strictly higher than the high threshold, and if it is, sets the value of P to 255, otherwise, sets it to 0.

The low and high threshold must be chosen carefully, in my implementation, I chose to make the low threshold L_t depend on the high threshold H_t based on a user chosen scalar s, such that $L_t = s * H_t$.

Result

Figure 7 Hysteresis thresholding, Ht = 0.13, s = 0.5 (i.e Lt = 0.065)

I noticed that this method found more contours, by also finding contours in areas of subtle change in intensity in addition to areas of high change. I found that this method solved the issues with global thresholding regarding the areas of subtle changes in intensity, by adding a extra local thresholding step. However, the contours are thick and may not correspond to the reality.

Non-Maximal Suppression

This method removes pixels that have a gradient value lower than the neighboring pixels in the gradient direction. In my implementation, I chose the neighboring pixels based on the angle of the gradient for the current pixel.

Result

Figure 8 Non-max suppression on image of bi-directional amplitude

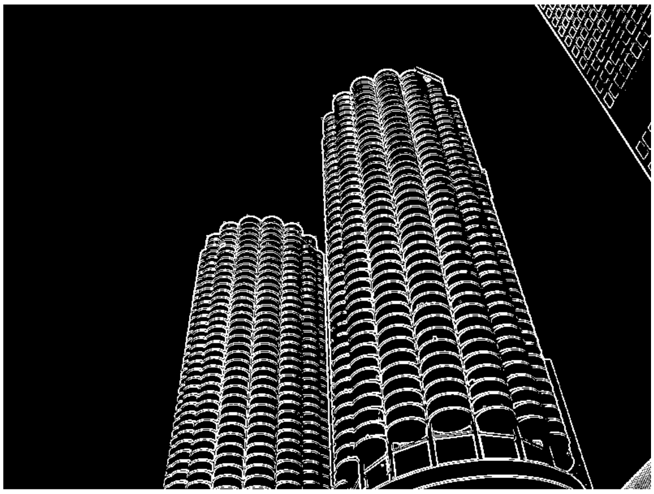
Figure 9 Hysteresis thresholding applied

Observations

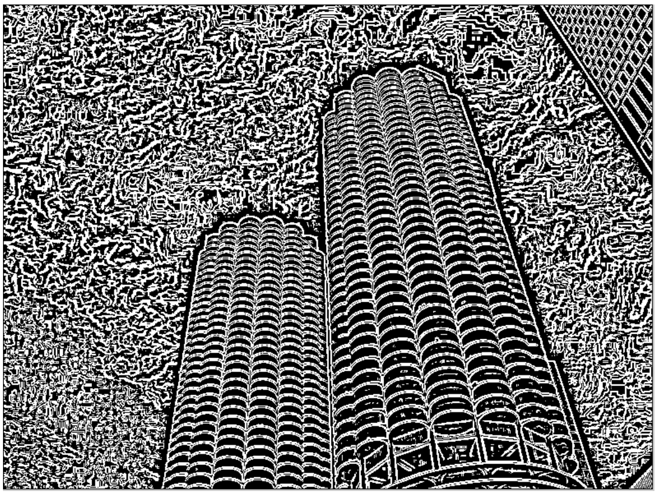
I noticed that this method had the effect of thinning the contours, noticeable on Fig 9. with a hysteresis threshold. This method, combined with hysteresis thresholding produces an image where the contours are thin and accurate.

Annexes

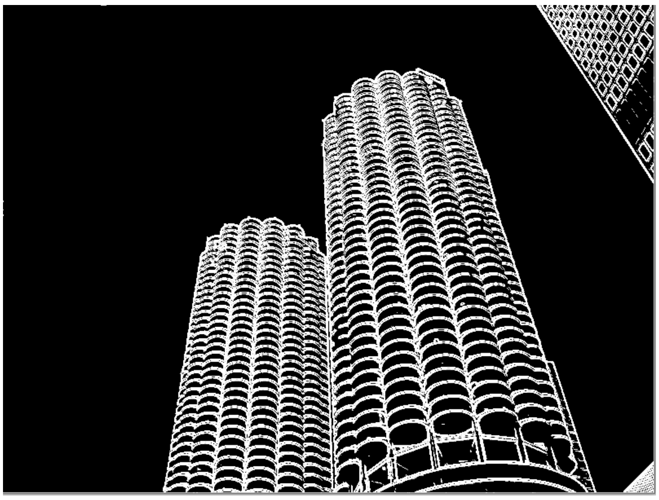
Best of



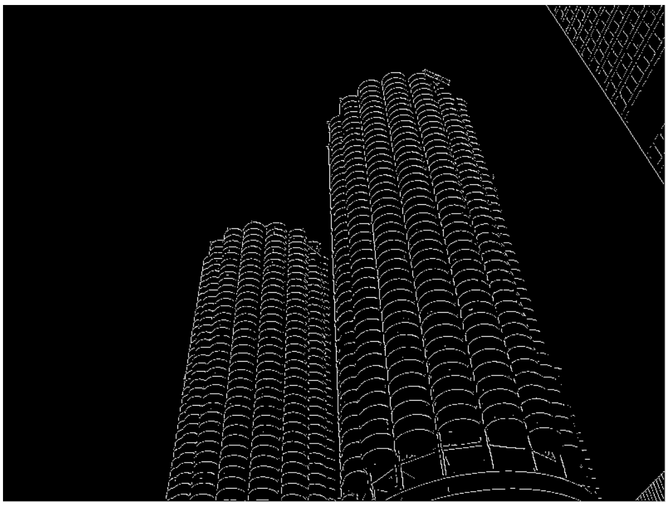
Global thresholding



Local thresholding



Hysteresis thresholding without non-Max suppression



Non-Max suppression, hysteresis threshold