Biometrics lab#4 and lab#5

Paweł Kobojek

December 1, 2017

Contents

1 Administrative stuff

1.1 Report#2 results

Results of your Report#2 were sent to you over email. In case of any questions write me an email (please don't interrupt your work on the classes as we do not have much time).

1.2 Earlier lab finish

I was informed by prof. Saeed that some of you prefer finish your lab earlier in order to focus on your thesis. It is of course fine for me. In such case, I'll send you metarials for the following labs and your task is to implement the methods and prepare a report. I'd like to know who is interested in such way of finishing this lab, especially if it's all of you. **If you don't want to come on the labs on *<2017-11-17 Fri>*, you are not required**

but please send me an email that you are not coming. If there's at least one person who wants to continue our classes in a normal manner I'll prepare the labs which will take place according to normal schedule for them as well.

2 Goal of this lab

Briefly, what we want to do on lab#4 is implement a KMM algorithm (link to the paper: `https://pdfs.semanticscholar.org/88ce/ea7b9db7a318c63908ff61f4e02a6aaabda7.pdf`). On lab#5, we are going to implemented a K3M algorithm (link to the paper: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.390.2963&rep=rep1&type=pdf`). Please test it on any image you can find on the Internet. Fingerprints and letters are especially good for testing this. Rest of this document is more fine-grained description. If you just want to implement it fast, the linked paper is more than enough.

2.1 Report#3

The scope or report#3 is Lab#4 and Lab#5 (KMM and K3M Algorithm). The deadline is *<2019-12-08 Sun 23:59>*. As always, please do describe the algorithm, provide your own thoughts/conclusions and show results on multiple images.

3 Thinning/Skeletonization

3.1 Outline

Thinning or **Skeletonization** is a procedure applied on binarized images. It results in finding a skeleton (ideally, 1 pixel wide) of the object of our concern. For example, if we are working on an OCR system (please note that this is not usually used anymore in modern OCRs), we may want to perform thinning on a letter before further processing (`like here`). Another (also used in practice) application is fingerprint verification and is shown here.

3.2 KMM Algorithm

KMM Algortithm (paper) was designed and implemented by prof. Saeed and his colleagues. It's name corresponds to first letters of authors' names. It is best to read the paper and work with it as it describes the algorithm

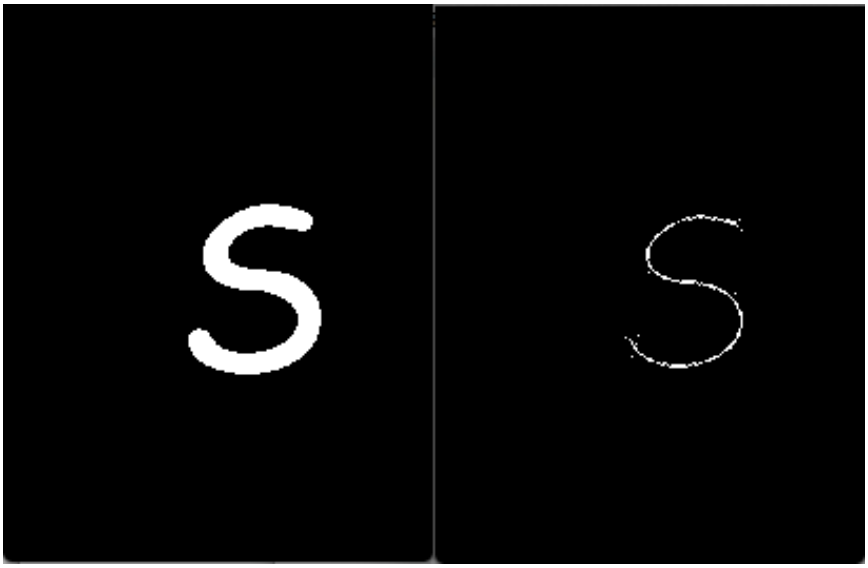

Figure 1: Letter before and after the skeletonization/thinning process.

exhaustively. However, you may find the description in this document as helpful. The images here come from the mentioned paper.

Please note again, that you should be all fine to work only with the linked paper and you may find reading rest of this section as redundant. However, if you are not sure about something in the paper you may take some value out of this redundancy (you can also ask me).

You can find the code block diagram in the last page of the mentioned paper. This may be helpful in coding the algorithm.

This algorithm works on binarized images. We assume that black pixels refer to object and white to the background. We want to skeletonize the object. We start by marking each black pixel with 1 (link to image). The 1's which stick to the background are marked as 2s or 3s in case they are in elbow corners (link to image). In other words, we set 1 to 3 if and only if it only sticks to the background in corners and not in top/right/bottom/left. We then set all contour pixels (contour = sticking to the background) to 4 if they have 2, 3 or 4 sticking neighbors (link to image). After that, we delete all 4s (set their value to 0 - background). (link to image) We then calculate the weights for each pixel marked as 2, 3 or 4 by applying

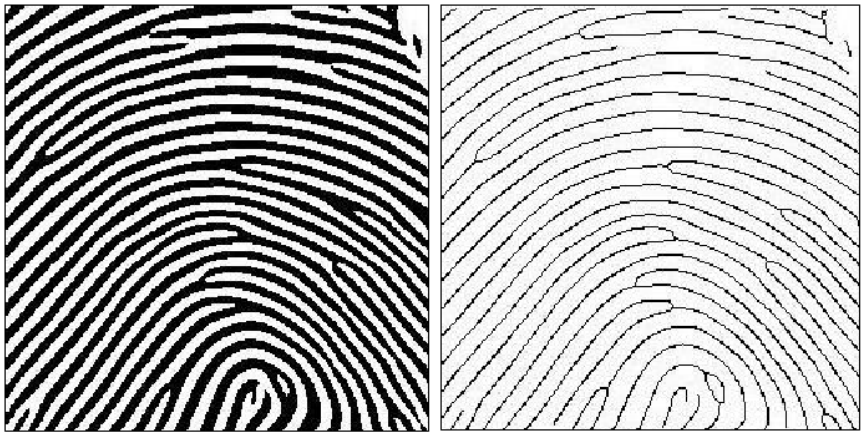

Figure 2: From: `http://web.media.mit.edu/~msaveski/projects/2010_fingerprint-verification.html`

a 2 dimensional mask to it. $\begin{bmatrix} 128 & 1 & 2 \\ 64 & x & 4 \\ 32 & 16 & 8 \end{bmatrix}$ This matrix gives us weights. We sum all the weights which correspond to the processed pixel's (X in the matrix) neighboring pixels which belong to the object (which means it's not background - it's value is non zero). Having calculated that, we can now check if the value we obtained is in the Deletion Array. The Deletion Array contains multiple, hardcoded values which encode patterns of pixel's neighborhood which should mark the pixel for deletion. You can find the array values in the paper (you should copy those values). If the processed pixel does not belong to the Deletion Array, then set its value to 1 (to prevent next iterations from processing it). The last step (or multiple steps, as it may be an iterative process) is removing the 2s and 3s if such deletion won't interrupt the skeleton (link to image).

3.3 K3M Algorithm

K3M Algorithm (paper) is the next iteration of KMM algorithm. The K3M algorithm is well described in the section 3 of the linked paper. The algorithm consists of two parts: iterative part and thinning-to-one-pixel-width-skeleton part. The iterative part on the other hand, consists of 7 (0 to 6) phases. Phase 0 regards marking certain image pixels as border pixels for further processing, phases 1-5 goal is deletion of certain border pixels and phase 6 is unmarking the remaining border pixels. Each phase has a corresponding

lookup array which lets us make a decision whether to mark the pixel as border (in case of phase 0)/delete the pixel (in case of phases 1-5). The second part of the algorithm (thinning to one pixel witdh skeleton) also has a corresponding lookup table. All the metnioned lookup tables are analogous to the lookup table from KMM algorithm. One has to compute the weights of the processed pixel and check if it resides in the lookup table in order to make a phase/part-specific decision (which means whether to mark the pixel as border/delete the pixel).

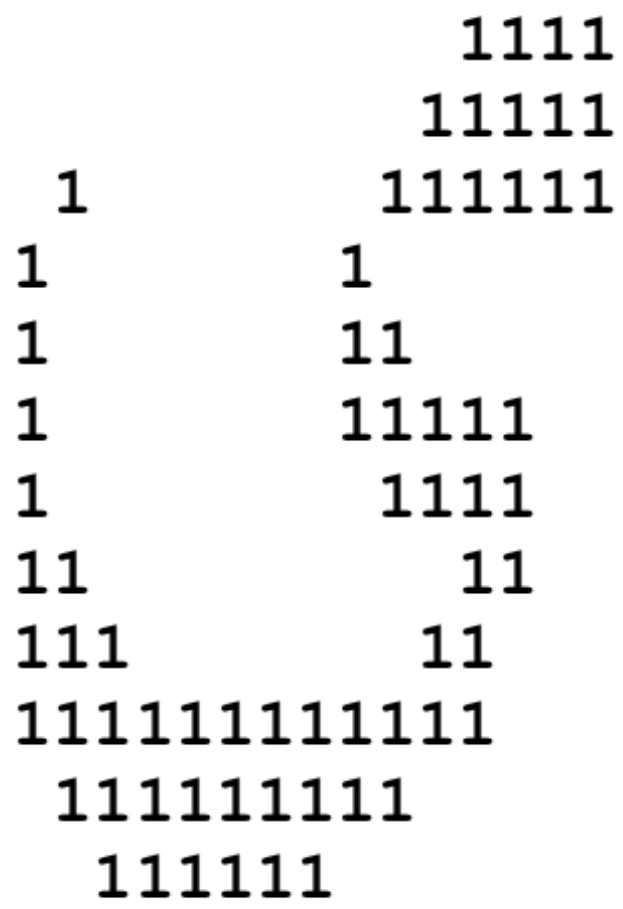

Figure 3: KMM Step 1

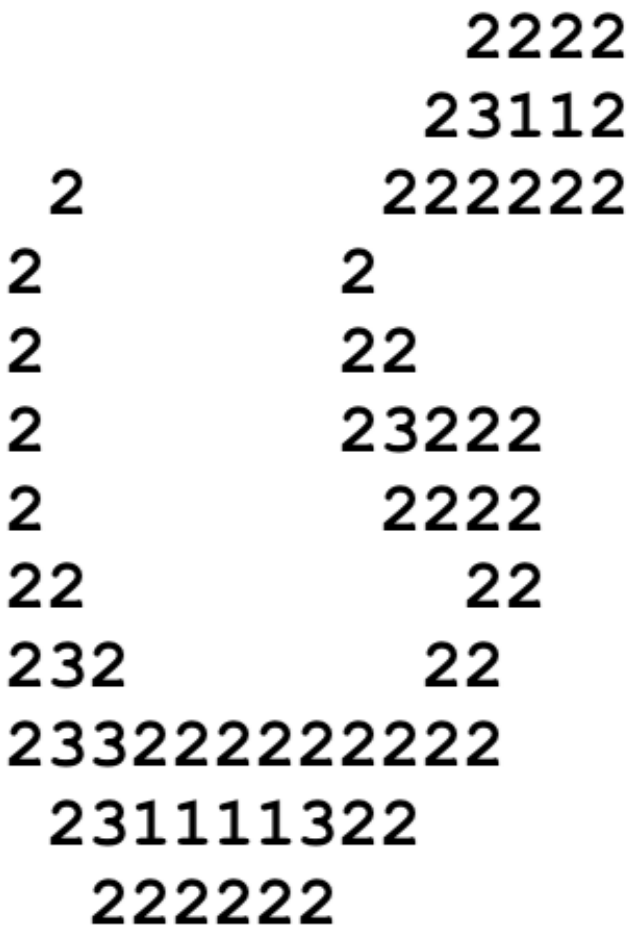

Figure 4: KMM Step 2

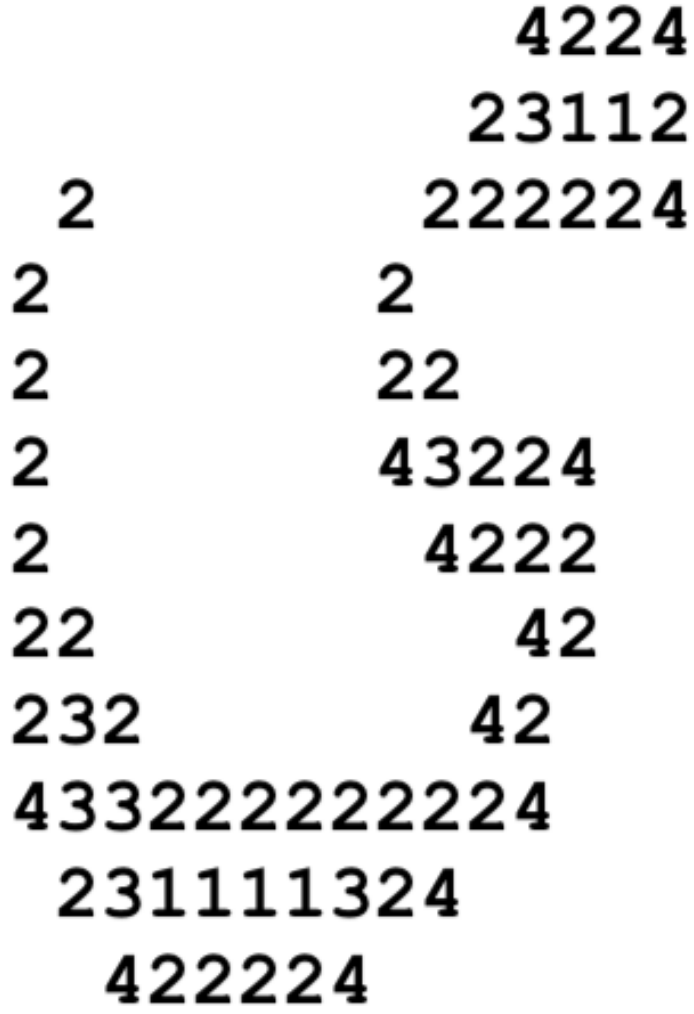

Figure 5: KMM Step 3

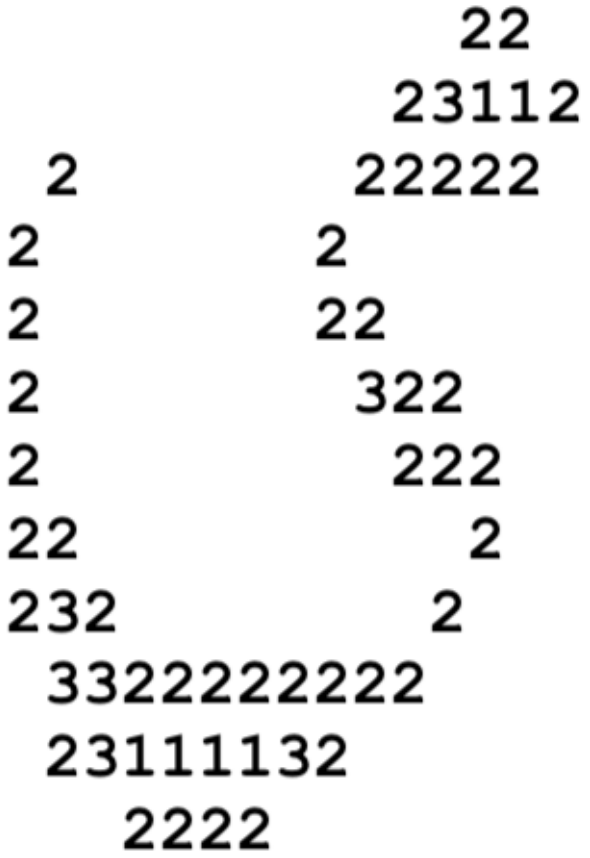

Figure 6: KMM Step 4

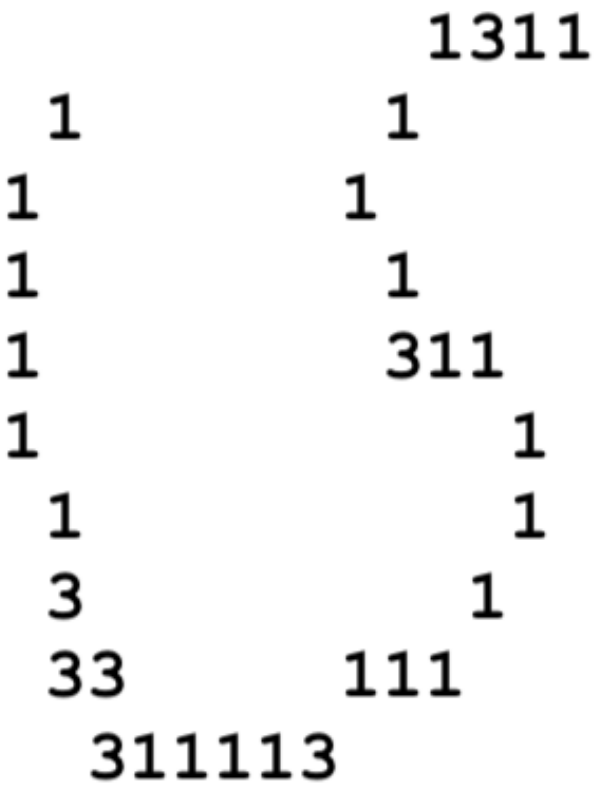

Figure 7: KMM Step 5

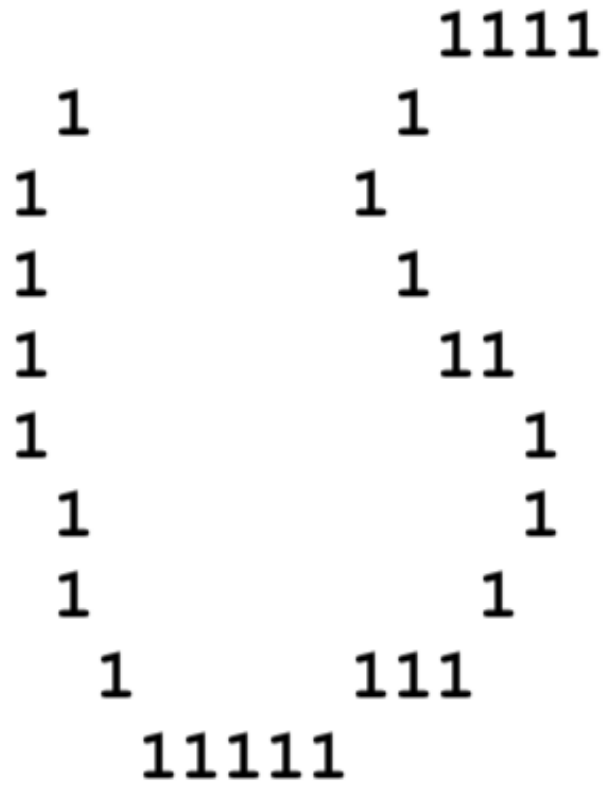

Figure 8: KMM Step 6