Object Skeletonization: Comparing State of the Art Algorithms

Daksh Thapar
IIIT Delhi
daksh18137@iiitd.ac.in

Himanshu Raj
IIIT Delhi
himanshu18038@iiitd.ac.in

Naman Tyagi
IIIT Delhi
naman18055@iiitd.ac.in

Pankil Kalra
IIIT Delhi
pankil18061@iiitd.ac.in

ABSTRACT

Given an image, we identify and highlight the topological skeleton region of object(s) present in the image using thinning techniques based on Zhang Suen Algorithm, Guo Hall Algorithm, start of the art techniques like KMM Thinning Algorithm, K3M Thinning Algorithm and Modified K3M Thinning Algorithm. The objective is to then qualitatively and quantitatively compare their performances based on multiple factors like types of image geometry, Thinning Rate (TR), Thinning Speed (TS), object points, preservation of right angles, topology and presence of 1-pixel width.

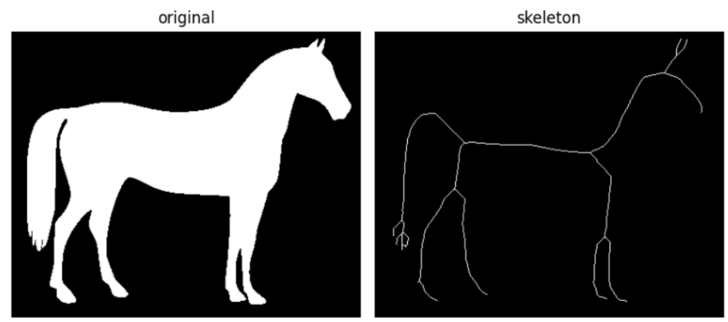

Figure 1: Skeletonization

1. INTRODUCTION

Skeletonization of a binary picture object may be **defined** as a connected medial graph along the limbs of the figure. This representation is particularly appropriate for compactly describing the shapes of thin figures such as hand-drawn characters, which have a fairly uniform width. A simple approach to skeletonization is provided by **thinning**, which involves the successive erosion of the outermost pixels of a figure until only the (connected) unit-width skeleton remains.

In its general form, the skeleton retains numbers on its medial lines indicating the distances of the nearest pixels on the outside of the figure; the fact that the combined information of skeleton and distance function values is sufficient to regenerate the entire shape (within the accuracy of 1 pixel) increases the power of the skeleton approach to shape analysis.

2. MOTIVATION

Skeletonization provides a compact yet effective representation of 2-D objects, which is useful in many low- and high-level image-related tasks including **OCR** (Optical Character Recognition), Object representation, Retrieval, Manipulation, Matching, **Registration**, Tracking, and **Compression**. It facilitates efficient characterization of topology, geometry, scale, and other related local properties of an object.

Despite that the notion of skeletonization is well defined in a continuous space, in the discrete world of image processing and CV, it is not, and, therefore, it is more often described using procedural approaches. Several computational approaches are available in the literature for extracting the skeleton of an object, some of which are widely different even at the level of their basic principles.

In this project, we present a comprehensive and concise survey of different skeletonization principles and algorithms and discuss their properties, challenges, and benefits. Different important aspects of skeletonization, namely, topology preservation, skeleton simplification and pruning, multiscale skeletonization, and parallelization are discussed.

3. METHODOLOGY

We are planning to implement different algorithms for Skeletonization purpose and will compare the results obtained from them. Algorithms we will be using are -

3.1 Zhang Suen Algorithm

The Zhang-Suen Thinning algorithm [5] is probably the most used thinning algorithm. Devised in 1984, the algorithm is what is called a 2-pass algorithm, meaning that for each iteration it performs two sets of checks to remove pixels from the image. The checks are devised so that the first set removes from the south east (bottom right) corner of the image, and the second set removes from the north west (top left) corner.

Algorithm: Assume black pixels are one and white pixels zero, and that the input image is a rectangular N by M array of ones and zeroes.The algorithm operates on all black pixels P1 that can have eight neighbours.

The neighbours are, in order, arranged as:

P9	P2	P3
P8	**P1**	P4
P7	P6	P5

Figure 2: Neighborhood Pixel of P1

Obviously the boundary pixels of the image cannot have the full eight neighbours.

- Define **A(P1)** = the number of transitions from white to black, (0 -> 1) in the sequence P2, P3, P4, P5, P6, P7, P8, P9, P2. (Note the extra P2 at the end - it is circular).
- Define **B(P1)** = The number of black pixel neighbours of P1. (= sum(P2 .. P9))

Step 1 All pixels are tested and pixels satisfying all the following conditions (simultaneously) are just noted at this stage.
(0) The pixel is black and has eight neighbours
(1) 2 <= B(P1) <= 6
(2) A(P1) = 1
(3) At least one of P2 and P4 and P6 is white
(4) At least one of P4 and P6 and P8 is white
After iterating over the image and collecting all the pixels satisfying all step 1 conditions, all these condition satisfying pixels are set to white.

Step 2 All pixels are again tested and pixels satisfying all the following conditions are just noted at this stage.
(0) The pixel is black and has eight neighbours
(1) 2 <= B(P1) <= 6
(2) A(P1) = 1
(3) At least one of P2 and P4 and P8 is white
(4) At least one of P2 and P6 and P8 is white
After iterating over the image and collecting all the pixels satisfying all step 2 conditions, all these condition satisfying pixels are again set to white.

Iteration: If any pixels were set in this round of either step 1 or step 2 then all steps are repeated until no image pixels are so changed.

3.2 Guo Hall Algorithm

It is an iterative thinning algorithm [1] with 2 subcycles that gives a 1-pixel wide skeleton for image. It was devised in 1989 by Zicheng Guo and Richard W. Hall.

Algorithm:

```
While points are deleted do
    For all pixels p(i,j) do
        if (a) C(P1) = 1
           (b) 2  N(P1)  3
           (c) Apply one of the following:
              1. (P2 | P3 | !P5) & P4 = 0 in
              odd iterations
              2. (P6 | P7 | !P9) & P8 = 0 in
              even iterations
        then
           Delete pixel p(i,j)
        end if
    end for
end while
```

Where,

- C(P1) = !P2 & (P3 | P4) + !P4 & (P5 | P6) + !P6 & (P7 | P8) + !P8 & (P1 | P2)
- N1(P1) = (P9 | P2) + (P3 | P4) + (P5 | P6) + (P7 | P8)
- N2(P1) = (P2 | P3) + (P4 | P5) + (P6 | P7) + (P8 | P9)
- N(P1) = MIN[N1(P1), N2(P1)]

3.3 KMM Algorithm

It is a sequential and iterative thinning algorithm [2] which maintains connectivity and bias-reduced skeletons for skeletonization outputs. It was devised in 2001 by Khalid Saeed, Marek Tabedzki and Mariusz Rybnik.

Algorithm:

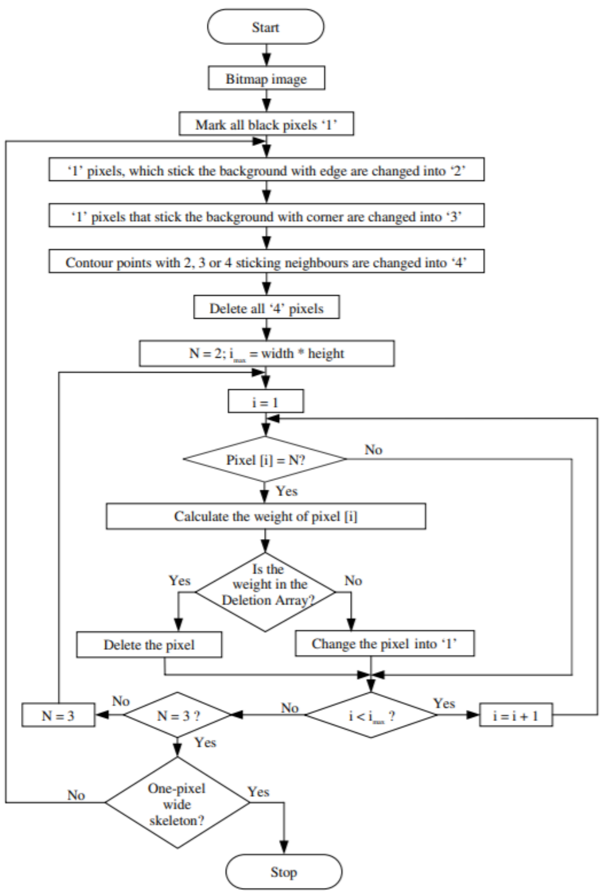

Figure 3: KMM algorithm

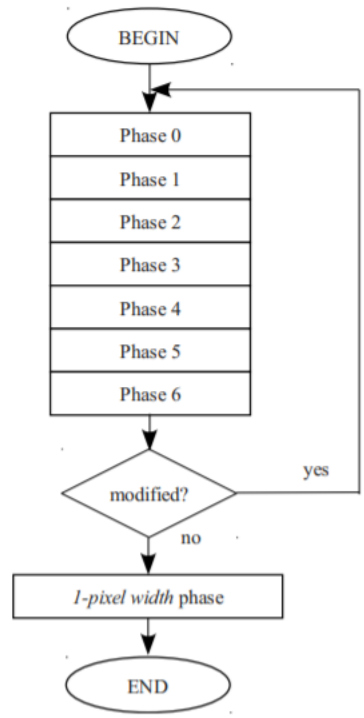

Figure 4: K3M algorithm

3.4 K3M Algorithm

K3M [3] is an improved and generalized KMM thinning algorithm and it was devised in 2016 by Khalid Saeed, Marek Tabedzki, Mariusz Rybnik and Marcin Adamski.

Algorithm:

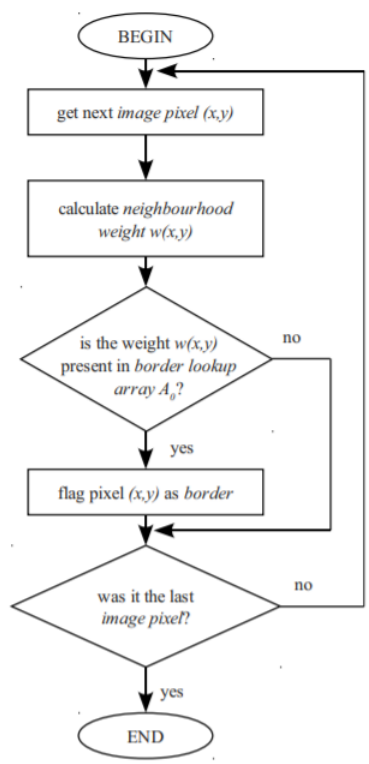

Figure 5: K3M 0th phase algorithm

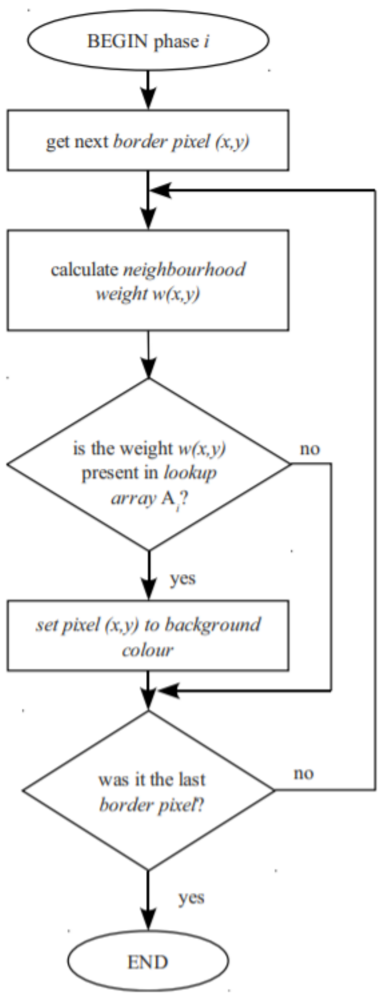

Figure 6: K3M Phase 1 to 5 algorithm

3.5 Modified K3M Algorithm

The K3M thinning algorithm is a general method for image data reduction by skeletonization. It had proved its feasibility in most cases as a reliable and robust solution in typical applications of thinning, particularly in preprocessing for optical character recognition. However, the algorithm had still some weak points. Since then K3M has been revised, addressing the best known drawbacks. Modified K3M [4] presents a modified version of the algorithm. The modifications solves the main drawback of K3M, namely, the results of thinning an image after rotation with various angles.

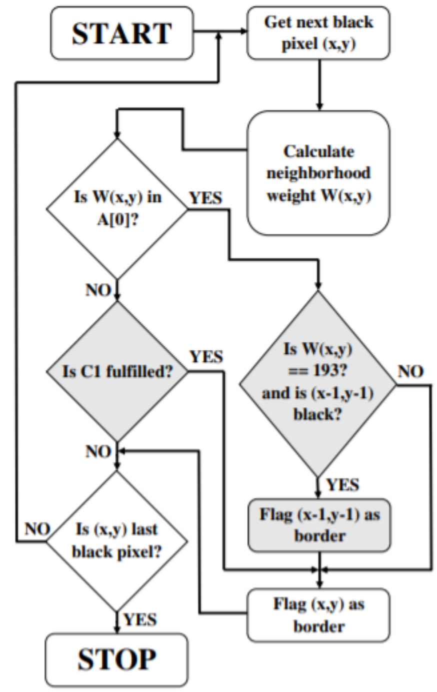

Figure 7: Modified K3M Phase 0 algorithm

This is an additional phase added in the modified K3M algorithm -

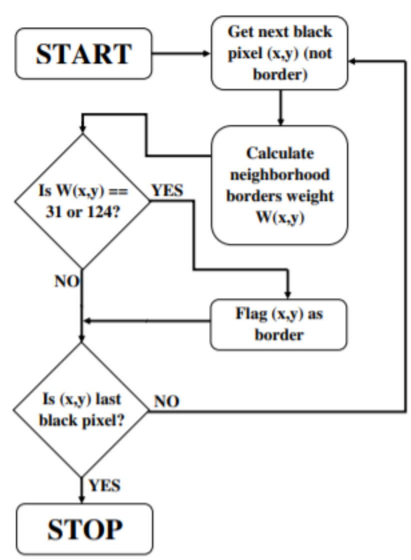

Figure 8: Modified K3M Phase 0-a algorithm

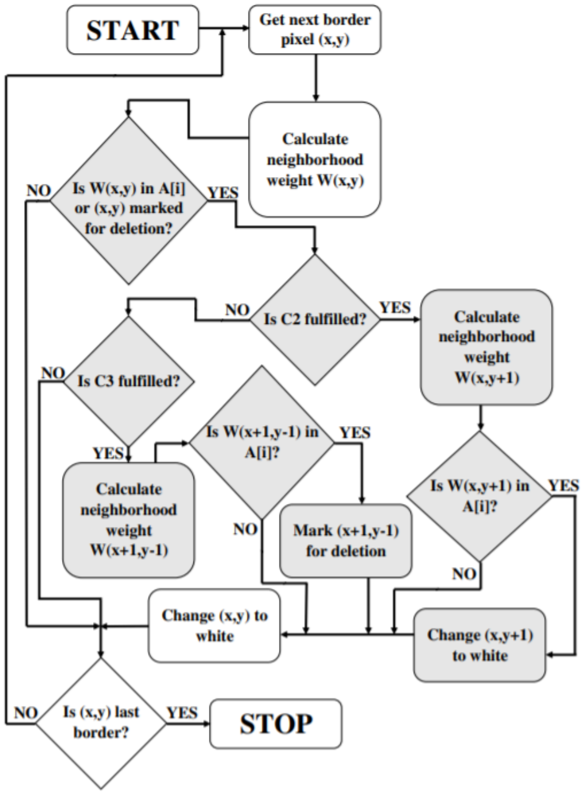

Figure 9: Modified K3M Phase 1 to 5 algorithm

4. RESULTS AND ANALYSIS

We applied all the algorithms on various binary images like human, dog, circle, arrow, few alphabets and words. Our result on the image containing alphabet A is attached below:

Figure 10: Original Image

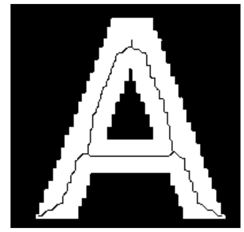

Figure 11: Result from Zhang Suen Algorithm

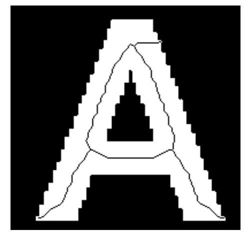

Figure 12: Result from Guo Hall Algorithm

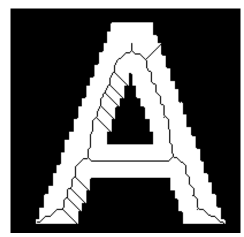

Figure 13: Result from KMM Algorithm

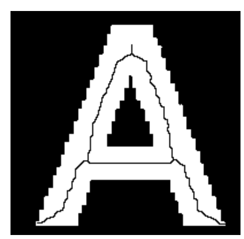

Figure 14: Result from K3M Algorithm

Figure 15: Result from Modified K3M Algorithm

Table 1: Evaluation Scores for different algorithms on different images

		Multiple Objects	Arrow	Circle	Human	Alphabet A	ti	Right Angles
ZS	OP	25043	5532	7845	8941	12770	281	41704
	SP	1705	168	1	414	476	61	1217
	TR	0.995	0.998	1.000	0.996	0.995	0.965	0.996
	TS	363.99	423.7	205.3	632.8	562.5	3367.8	616.6
	ET	64.11	12.66	38.20	13.47	21.86	0.065	65.65
GH	OP	25043	5532	7845	8941	12770	281	41704
	SP	1642	191	3	377	412	67	1177
	TR	0.996	0.998	0.999	0.997	0.997	0.965	0.996
	TS	369.3	367.1	206.2	646.3	677.4	3523.3	502.7
	ET	63.36	14.54	38.03	13.25	18.24	0.060	80.60
KMM	OP	25043	5532	7845	8941	12770	281	41704
	SP	1673	173	28	394	542	63	1240
	TR	0.997	0.998	0.999	0.997	0.996	0.967	0.996
	TS	934.4	1026.4	530.5	1545.3	1331.5	6767.0	1437.2
	ET	25.01	5.22	14.73	5.53	9.18	0.032	28.15
K3M	OP	25043	5532	7845	8941	12770	281	41704
	SP	2406	220	165	541	549	74	1244
	TR	0.992	0.997	0.998	0.993	0.993	0.951	0.996
	TS	5414.2	5593.2	3990.3	7265.1	6188.7	12588.2	5764.1
	ET	4.18	0.95	1.92	1.15	1.97	0.016	7.02
M.K3M	OP	25043	5532	7845	8941	12770	281	41704
	SP	1658	183	133	391	415	62	1219
	TR	0.996	0.998	0.998	0.997	0.996	0.966	0.997
	TS	324.8	447.3	207.4	663.4	450.0	3195.6	663.00
	ET	72.00	11.95	37.17	12.83	27.45	0.068	61.06

Observation from above Results -

On this image (**Figure 10**: Original Image of 'A'), most of the algorithms are giving good results for skeletonization. However we can see that results from **K3M, Modified K3M are much better** than Guo Hall, KMM and Chang Suen. To compare the algorithms in detail, we can take a few parameters like number of skeleton points, execution speed, etc.

We have run these algorithms on more examples and their detailed comparison on different parameters are shown in Table 1. The definition of different parameters are as follows -

- **Object Points** - OP refers to the number of Pixels in the Foreground Image.
- **Skeleton Points** - SP refers to the number of Skeleton Points in the Skeleton.
- **Thinning Rate** - TR refers to the degree of thinness of the image. If TR==1, then the image is perfectly thinned while TR==0 means the image is not thinned at all.
- **Thinning Speed** - TS refers to the number of pixels thinned per unit time.
- **Execution Time** - ET refers to the time taken by the algorithm to execute.

4.1 Observations and Analysis

Now based on the results obtained on different parameters on different images we have observed the following points -

- For **circular objects**, **Zhang Suen** is performing the best as we only get only one skeleton point (From Table 1) which is the **center of the circle**. Whereas the other algorithms like Guo Hall and KMM are producing small horizontal and vertical lines. And K3M and Modified K3M are producing big cross like structure as skeleton.

- In algorithms like Zhang Suen, Guo Hall and KMM, the right angle is not preserved in the objects. Whereas in the algorithms like **K3M and modified K3M the right angle is preserved**.

- The Thinning Speed (**TS**) is maximum for **K3M**, it is almost 10 times faster than Zhang Suen, Guo Hall, modified K3M and 5 times faster than KMM algorithm. Also, the Execution Time (**ET**) is the least for **K3M**, around 10-15 times lesser than Zhang

Suen, Guo Hall, modified K3M and 5 times lesser than KMM algorithm.

- The **topology is preserved** in **Zhang Suen, KMM, K3M and Modified K3M** algorithms. But in Guo Hall, the topology of the foreground object is not preserved.
- The Thinning Rate (**TR**) is consistently better in **Modified K3M** algorithm (0.997 from Table - 1) when compared to other algorithms, closely followed by KMM and Guo Hall.

-

5. CONCLUSION

For the observation we have used novel evaluation scores to compare the skeletonization process of various image geometries. Using these metrics, we are able to successfully compare the different algorithms and verify the success of state of the art algorithms to others. We have shown that the new algorithms are able to produce better thinning results and more efficient in terms of execution time.

6. REFERENCES

[1] Z. Guo and R. W. Hall. Parallel thinning with two-subiteration algorithms. *Communications of the ACM*, 32(3):359–373, 1989.

[2] K. Saeed, M. Rybnik, and M. Tabedzki. Implementation and advanced results on the non-interrupted skeletonization algorithm. In *International Conference on Computer Analysis of Images and Patterns*, pages 601–609. Springer, 2001.

[3] K. Saeed, M. Tabędzki, M. Rybnik, and M. Adamski. K3m: A universal algorithm for image skeletonization and a review of thinning techniques. 2010.

[4] M. Tabedzki, K. Saeed, and A. Szczepański. A modified k3m thinning algorithm. *International Journal of Applied Mathematics and Computer Science*, 26(2):439–450, 2016.

[5] T. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236–239, 1984.