Project report

Skeletonization/thinning

Brief description:

Skeletonization also known as "thinning" is a preprocessing procedure widely used in computer vision, pattern recognition, image processing and computer graphics. It deals with the line drawing images whose characteristics have little to do with the thickness of the lines such as OPTICAL CHARACTER RECOGNITION.

Definition:

Skeletonization or thinning is a process by which a one-pixel width representation (or skeleton) of an object is obtained. In other words after pixels have been peeled off, the pattern should be still recognized .

Hence the skeleton obtained must have the following properties

- As thin as possible (possibly one pixel width)
- Connected
- Centered

I used a simple thinning algorithm with reference to HILDITCH'S algorithm.

Hilditch's algorithm consists of performing multiple passes on the pattern and on each pass, the algorithm

	P9	P 2	P 3
	P8	P1	P4
1	P 7	P6	P5

checks all the pixels and decides to change a pixel

P1 from black to white if it satisfies the following four conditions:

- 2 < = B(p1) < = 6
- A(p1)=1
- p2.p4.p8=0 or A(p2)!= 1
- p2.p4.p6=0 or A(p4)!= 1 Stop when nothing changes (no more pixels can be removed)

Where B(p1) = the number of non-zero neighbors of p1 and

A(p1) = number of 0,1 patterns in the sequence p2,p3,p4,p5,p6,p7,p8,p9,p2 (in clockwise sequence)

For example consider the following picture



Four Conditions explained

• Condition 1 ensures that it is a boundary pixel and no end point pixel and no isolated pixel is removed.

- Condition 2 ensures the connectivity.
- Condition 3 ensures that 2 pixel wide vertical lines do not get completely eroded.
- Condition 4 ensures that 2 pixel vertical lines do not get completely eroded.

When i have applied the algorithm as such to different fonts of Assamese script, the result is poor, the size of the characters almost remaining same.

So I made some modifications in this algorithm which proved helpful for me in thinning Assamese characters. Assamese script is similar to Hindi script. The unique feature of these two is "headline" property. Headline is a horizontal line which connects all the characters in the word. Almost Every word in Assamese or Hindi has this headline. I used this property during the modifications.

The modified algorithm is presented below.

The algorithm performs multiple passes on the pattern and on each pass, the algorithm checks all the pixels and decides to change a pixel p1 if it satisfies the following conditions

- 2 < = B(p1) < = 6
- A(p1)=1
- A(p4)!=1

The algorithm stops when no more pixels can be removed.

The condition 3 is removed because no vertical lines are present as such, these are connected by the head line so no problem aroused for me. The output of this contains thin vertical lines as needed but width of the horizontal lines remained same and the noise is more. Therefore the output obtained is again fed to the following algorithm

- 2 < = B(p1) < = 6
- A(p1)=1

Again it stops when no more pixels can be removed.

The output of this is to the expected level.

The output of 2 different font words :

Font 1:



Drawbacks:

On observation, I found following drawbacks in the output

- Character "na" is not properly thinned.
- Noise exists in the output.

References:

• Chapter 7: Skeletons Authored by Godfried Toussaint

• Optical Character Recognition of printed Tamil characters by Anubani Subramanian and Bhadri Kubendran. <u>http://www.ee.vt.edu/~anbumani/tamilocr/</u>

• B.Tech thesis of Mandeep Alumnus of IITK

• Project explanation of Daniela Azar http://cgm.cs.mcgill.ca/~godfried/teaching/projects97/azar/skeleton.html

• A simple and robust Thinning Algorithm C.Lee and P.S.Wang college of computer science, Northeastern University , Boston MA 02115 , USA

(This was done as a project in winter 2003 by Thanniru Ramakrishna under the guidance of Dr.SVRao Computer Science Department, IIT GUWAHATI.)