Run-Length Coding

Run-Length Encoding

This methods exploits the fact that along any particular scan line there will usually be long runs of zeros or ones.

Two approaches are commonly used in run-length encoding.

-the start positions and length of runs of 1s for each row are used.

-lengths of runs, starting with the length of the 0 run.

1	1	1	0	0	0	1	1	0	0	0	1	1	1	1	0	1	1	0	1	1	1
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Start and length of 1 runs: (1,3) (7,2) (12,4) (17,2) (20,3)
(5,13) (19,4)
(1,3) (17,6)

Length of 0 and 1 runs: 0,3,3,2,3,4,1,2,1,3
4,13,1,4
0,3,13,6

Technical University of Cluj Napoca
Computer Science Department

Run-Length coding (2)

Computation of the geometric properties from run-length

Use the convention that r_{ik} is the k-th run of the i-th line and that the first run in each row is a run of zeros (thus all the even runs will correspond to ones in the image). There are m_i runs on the i-th line

Area

$$A = \sum_{i=1}^{n} \sum_{k=1}^{m_i/2} r_{i,2k}$$

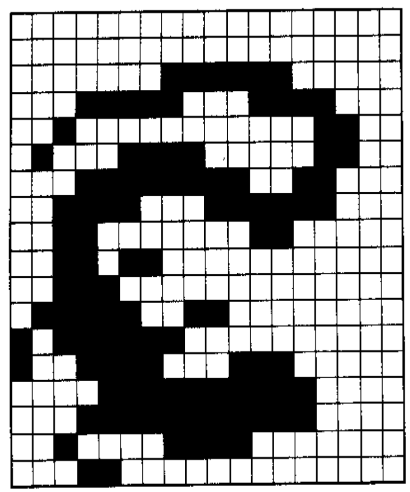
Position

First of all, we obtain the horizontal projection as follows:

$$h_i = \sum_{k=1}^{m_i/2} r_{i,2k}$$

From this we can easily compute the vertical position of the center of area using:

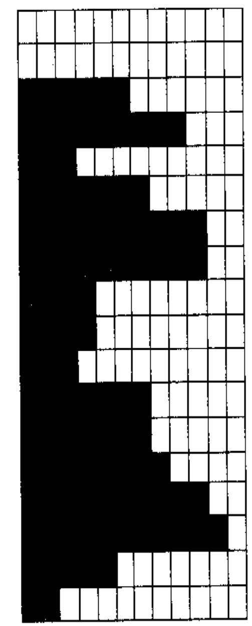
$$\bar{i}A = \sum_{i=1}^{n} i h_i$$



Run-length code

(0,0)
(0,0)
(8,6)
(4,5) (12,4)
(3,1) (15,2)
(2,1) (6,4) (15,2)
(4,8) (14,2)
(3,4) (10,6)
(3,2) (12,2)
(3,2) (6,2)
(3,3)
(2,5) (9,2)
(1,1) (3,6)
(1,1) (4,4) (11,3)
(5,10)
(4,11)
(3,1) (8,4)
(4,2)

Horizontal projection



Run-Length coding (3)

The vertical projection is obtained by adding up all picture cell values in one column of the image.

It is difficult to compute vertical projection directly from the run lengths. Consider instead the **first difference of the vertical projection**: $\bar{v}_j = v_j - v_{j-1}$, $\bar{v}_1 = v_1$

The first difference of the vertical projection can be obtained by projecting not the image data, but the **first horizontal differences of the image data**: $\bar{b}_{i,j} = b_{i,j} - b_{i,j-1}$

The first difference $\bar{b}_{i,j}$ has the advantage over $b_{i,j}$ itself that is nonzero only at the beginning of each run. It equals +1 where the data change from 0 to a 1, and –1 where the data change from a 1 to a 0.

Technical University of Cluj Napoca
Computer Science Department

Run-Length coding (4)

													0	
														$r_{i,k}$
							0	1	1	0			2	8,2,3
						0	1	1	1	0			3	7,3,3
					0	1	1	1	1	1	0		5	6,5,2
		0	1	0	0	1	1	1	1	1	0		6	3,1,2,5,2
	0	1	1	1	1	1	1	1	1	0			8	2,8,3
0	1	1	1	1	1	1	1	1	0				8	2,8,4
0	1	1	1	1	1	1	1	0					7	1,7,5
		0	1	1	1	0							3	3,3,7
													h_i	

Original image

Run-Length coding (5)

$\bar{b}_{i,j}$

													0
								+1	0	-1			2
							+1	0	0	-1			3
						+1	0	0	0	0	-1		5
			+1	-1		+1	0	0	0	0	-1		6
		+1	0	0	0	0	0	0	0	-1			8
	+1	0	0	0	0	0	0	0	-1				8
	+1	0	0	0	0	0	0	-1					7
			+1	0	0	-1							3
													h_i
0	+2	+1	+2	-1	0	+1	+1	0	-1	-3	-2	0	
0	2	3	5	4	4	5	6	6	5	2	0	0	v_j

Technical University of Cluj Napoca
Computer Science Department

Run-Length coding (6)

We can locate these places by computing the summed run-length code

$$\tilde{r}_{ik} = 1 + \sum_{p=1}^{k} r_{i,p}$$

recursively: $\tilde{r}_{i,0} = 1$ and $\tilde{r}_{i,j+1} = \tilde{r}_{i,j} + r_{i,j+1}$

Then, for a transition at $j = \tilde{r}_{ik}$ we add $(-1)^{k+1}$ to the accumulated total for the first difference of the vertical projection

Technical University of Cluj Napoca
Computer Science Department

Run-Length coding (7)

													0
							0	1	1	0			2
						0	1	1	1	0			3
					0	1	1	1	1	1	0		5
		0	1	0	0	1	1	1	1	1	0		6
	0	1	1	1	1	1	1	1	1	0			8
0	1	1	1	1	1	1	1	1	0				8
0	1	1	1	1	1	1	1	0					7
		0	1	1	1	0							3
													h_i
0	+2	+1	+2	-1	0	+1	+1	0	-1	-3	-2	0	
0	2	3	5	4	4	5	6	6	5	2	0	0	v_j

$r_{i,k}$	$\tilde{r}_{ik} = 1 + \sum_{p=1}^{k} r_{i,p}$
8,2,3	1,9,11,14
7,3,3	1,8,11,14
6,5,2	1,7,12,14
3,1,2,5,2	1,4,5,7,12,14
2,8,3	1,3,11,14
1,8,4	1,2,10,14
1,7,5	1,2,9,14
3,3,7	1,4,7,14

IMAGE PROCESSING

Run-Length coding (4)

- From this difference we can compute the vertical projection itself using the simple summation

$$v_j = \sum_{p=1}^{j} \bar{v}_p$$

- Or recursively: $v_0 = 0$ and $v_{j+1} = v_j + \bar{v}_{j+1}$
- Given the vertical projection, we can easily compute the horizontal projection of the center of area $A\bar{j} = \sum_{j=1}^{m} j v_j$

- The diagonal projection can be obtained in a way similar to that used to obtain the vertical projection.

Technical University of Cluj Napoca
Computer Science Department