HAL
open science

Average Efficiency of Data Structures for Binary Image Processing

Claire Mathieu, Claude Puech, Hussein Yahia

To cite this version:

Claire Mathieu, Claude Puech, Hussein Yahia. Average Efficiency of Data Structures for Binary Image Processing. Information Processing Letters, 1987, 26 (2), pp.89-93. 10.1016/0020-0190(87)90043-3 . hal-00948103

HAL Id: hal-00948103

https://inria.hal.science/hal-00948103v1

Submitted on 17 Feb 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Information Processing Letters 26 (1987/88) 89–93
North-Holland

19 October 1987

AVERAGE EFFICIENCY OF DATA STRUCTURES FOR BINARY IMAGE PROCESSING

C. MATHIEU, Claude PUECH and Hossein YAHIA

Département de Mathématiques et d'Informatique, Laboratoire d'Informatique de l'Ecole Normale Supérieure, 45 Rue d'Ulm, 75230 Paris Cédex 05, France

Communicated by W.L. Van der Poel
Received 12 February 1987

Keywords: Image processing, quadtree, chain code, run-length encoding, average memory occupancy

1. How to store a binary image

Several structures exist for representing binary images: let us suppose that the image is drawn on a $2^h \times 2^h$ array made of black-or-white pixels.

Each connected region can be represented by a starting point and a chain code that describes its boundary in terms of a sequence of directions chosen from E (East), N (North), W (West) or S (South), each letter of the sequence corresponding to an elementary step in the associated direction. Thus, the boundary of image I (see Fig. 1) is encoded by $E^4S^4E^2NESWS^2W^6N^6$.

The image can also be stored as a linked list of the runs of pixels of the same colour met when reading the array row by row, each run being given by its global length.

But, the data structure we are most concerned with here is the quadtree representation of an image, which is based on a recursive subdivision of the space. Starting from a $2^h \times 2^h$ grid, we subdivide it into four $2^{h-1} \times 2^{h-1}$ grids ('quadrants'), each of which is in turn subdivided into four subquadrants, and so on until a quadrant of uniform colour is reached. Thus, we obtain a tree (of height $\leqslant$ h) where each interior node has outdegree four (see Fig. 2 for an example). In recent years, it has aroused much interest (Samet, in particular, has studied it at length—see [1,4]), and yields a model of random images that allows to compute its average performances.

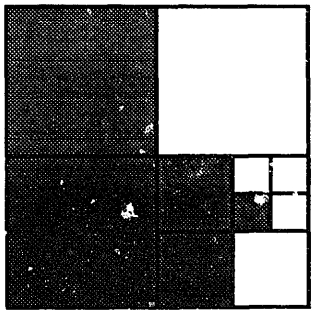

Fig. 1.

0020-0190/87/$3.50 © 1987, Elsevier Science Publishers B.V. (North-Holland)

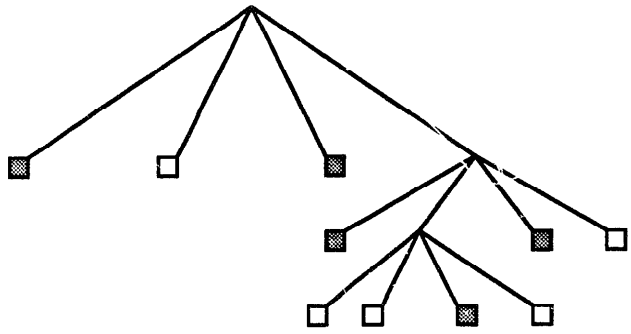

Fig. 2.

2. Average amount of storage required by each structure

Let Q_h be the set of all quadtrees of height $\leqslant h$. We define a probability distribution on Q_h in the following way: consider a sequence (β_i) of reals between zero and $\frac{1}{2}$ $(\beta_0 = \frac{1}{2})$. A random tree of Q_h is built by using a branching process, such that the quantity $2\beta_i$ is the probability for a node at level $h - i$ of being external (and so, at level h, all nodes are external with probability one, as they should). Once a node is known to be external, it is assumed to be white or black with equal probability. Thus, the probability of tree t of Q_h is given by the following formula:

$$\pi_h(t) = \begin{cases} \beta_h & \text{if } t = \circ, \\ \beta_h & \text{if } t = \bullet, \\ (1-2\beta)\pi_{h-1}(t_1)\pi_{h-1}(t_2)\pi_{h-1}(t_3)\pi_{h-1}(t_4) & \text{if } t = \underset{t_1\ t_2\ t_3\ t_4}{\bullet}. \end{cases}$$

In the case when, for arbitrary i, β_i is equal to a constant β, the following can be interpreted for the associated images: when β is near zero, the random images thus obtained look like clouds of points and, when β increases, the images are more and more structured.

Choosing β_i equal to $1/2(i+1)$ amounts to having a random model of images such that, in the quadtree representation, the external node containing a fixed pixel has got equal probability of being of any size. This is equivalent to Samet's definition of a random image: a node is equally likely to appear in any position and level in the tree.

We now turn our attention to the amount of storage needed for each of the data structures.

2.1. Proposition. *The average number of runs in Run-Length-Encoding of two-dimensional binary images is equal to*:

Case $\beta_i = \beta$:

$$[4(1-2\beta)]^h \frac{(1-2\beta)}{2(1-4\beta)} + 2^{h-1}(1-6\beta)/(1-4\beta).$$

Case $\beta_i = 1/2(i+1)$:

$$h2^{h-1}/(h+1) + 4^h/(h+1).$$

General case:

$$2^h \sum_{i=0}^{h-1} 2^i(1-2\beta_h)\cdots(1-2\beta_{h-i+1})(3\beta_{h-i} - \tfrac{1}{2}) + 4^h(1-2\beta_h)\cdots(1-2\beta_1).$$

2.2. Proposition. *The average number of nodes in the quadtree representation of binary images is equal to:*
Case $\beta_i = \beta$:

$$1/(8\beta - 3) + [4(1 - 2\beta)]^h \frac{4(2\beta - 1)}{8\beta - 3}.$$

Case $\beta_i = 1/2(i + 1)$:

$$\tfrac{1}{3}(4^{h+1} - 1) - \frac{4}{h+1}\tfrac{1}{9}\left[(3h - 1)4^h + 1\right].$$

General case:

$$1 + \sum_{j=1}^{h} 4^j(1 - 2\beta_h)(1 - 2\beta_{h-1}) \cdots (1 - 2\beta_{h-j+1}).$$

See [2,5] for further results.

As for the chain code representation, since its length is proportional to the number of concerns of the boundary (number of changes of direction), its mean is easily seen to be equal to the average number of external nodes of the corresponding quadtrees, that is, $1 + (\frac{3}{4})^{N-1}$, where N is the total number of nodes of the quadtree.

The accuracy of the model of randomness may be tested through statistics on images, by computing, for example, the mean perimeter of the images and comparing it with its theoretical value. The perimeter algorithm itself may be found in [4].

2.3. Proposition. *The average perimeter* [per] *of a binary image is equal to:*
Case $\beta_i = 0$:

$$[\text{per}] = 2^{h+1}.$$

Case $\beta_i = \frac{1}{4}$:

$$[\text{per}] = \tfrac{1}{2}h + 2 - \tfrac{1}{3}\left(\tfrac{1}{4}\right)^h \sim \tfrac{1}{2}h \quad (h \to +\infty).$$

Case $\beta_i = \beta \neq 0, \frac{1}{4}$:

$$[\text{per}] = (6\beta - 1)/(4\beta - 1) + \frac{2\beta - 1}{4\beta - 1}[2(1 - 2\beta)]^h.$$

Case $\beta_i = 1/2(i + 1)$:

$$[\text{per}] = 2 + \frac{2^{h+1} - (h + 2)}{h + 1} \sim 2^{h+1}/(h + 1) \quad (h \to +\infty).$$

Proof. Let L(t) be the set of leaves of the tree t encoding the image. For each direction D taken in E, N, W, S, let $L^D_>(t)$ be the set of all black leaves of t whose neighbour in direction D is also black and bigger than themselves, and $L^D_=(t)$ be the set of all black leaves of t whose neighbour in direction D is also black and of equal size (i.e., at the same level as in t). $4/2^i$ is the perimeter of a leaf at level i. The perimeter of the

whole image is then given by the following formula:

$$\begin{aligned}\mathrm{per}(t) = &\sum_{n\in L(t)} \mathrm{per}(n) - 2\sum_{n\in L^E_=(t)} \tfrac{1}{4}\mathrm{per}(n) - \sum_{n\in L^E_>(t)} \tfrac{1}{4}\mathrm{per}(n)\\ &-2\sum_{n\in L^W_=(t)} \tfrac{1}{4}\mathrm{per}(n) - \sum_{n\in L^W_>(t)} \tfrac{1}{4}\mathrm{per}(n) - 2\sum_{n\in L^N_=(t)} \tfrac{1}{4}\mathrm{per}(n) - \sum_{n\in L^N_>(t)} \tfrac{1}{4}\mathrm{per}(n)\\ &-2\sum_{n\in L^S_=(t)} \tfrac{1}{4}\mathrm{per}(n) - \sum_{n\in L^S_>(t)} \tfrac{1}{4}\mathrm{per}(n).\end{aligned}$$

Taking expectations, and allowing for the fact that all directions are equivalent, we have

$$\begin{aligned}[\mathrm{per}] = &\sum_{t\in Q^{[h]}} \pi^{[h]}(t)\Big(\sum_{n\in L(t)} \mathrm{per}(n)\Big) - 8\sum_{t\in Q_{[h]}} \pi^{[h]}(t)\Big(\sum_{n\in L^E_>(t)} \tfrac{1}{4}\mathrm{per}(n)\Big)\\ &-4\sum_{t\in Q^{[h]}} \pi^{[h]}(t)\Big(\sum_{n\in L^E_=(t)} \tfrac{1}{4}\mathrm{per}(n)\Big),\end{aligned}$$

and splitting according to the levels of the leaves yields

$$[\mathrm{per}] = \sum_{i=0}^{h} \frac{4}{2^i}\left(f_i - 2f^E_{>,i} - f^E_{=,i}\right),$$

where f_i (respectively $f^E_{>,i}$, $f^E_{=,i}$) is the average cardinality of $L(t)$ (respectively $L^E_>(t)$, $L^E_=(t)$) at level i. If $p_>(i)$ (respectively $p_=(i)$) denotes the probability that the eastern neighbour of a leaf at level i is bigger than (respectively is of the same size as) itself, we clearly have $f^E_> = \frac{1}{4}p_>(i)f_i$, and $f^E_= = \frac{1}{4}p_=(i)f_i$.

An easy calculation shows that $f_i = 2\times 4^i\beta_{h-i}(1-2\beta_{h-i+1})\cdots(1-2\beta_h)$. Computing $p_=(i)$ can be done by considering level $i-j$ of the youngest ancestor common to a leaf and to its eastern neighbour; searching for a leaf neighbour is done by going up the branch of the leaf and asking at each level whether the node is an eastern son of its father (in which case we have to go further up the branch) or whether it is a western son (in which case we have found an ancestor common to the leaf and its neighbour, and can now proceed downwards in a symmetric way). Thus, at each level examined, the probability that we have found the ancestor is $\frac{1}{2}$. The neighbour we search is at level i if and only if all the intermediate nodes in the downwards path are internal, until at level i an external node is reached. This justifies the following result:

$$p_=(i) = \sum_{j=0}^{i-1} \frac{1}{2^{i-j}}(1-2\beta_{h-j-1})\cdots(1-2\beta_{h-i+1})2\beta_{h-i}. \tag{1}$$

In a similar fashion we obtain

$$p_>(i) = \sum_{j=0}^{i-1} \frac{1}{2^{i-j}} \sum_{k=j+1}^{i-1} (1-2\beta_{h-j-1})\cdots(1-2\beta_{h-k+1})2\beta_{h-k}. \tag{2}$$

Putting all this together finally yields the desired result on the mean perimeter. □

References

[1] C.R. Dyer, A. Rosenfeld and H. Samet, Region representation: Boundary codes from quadtrees, Comm. ACM 23 (3) (1980) 171–179.

[2] C. Puech and H. Yahia, Quadtrees, octrees, hyperoctrees: A unified analytical approach to tree data structures used in graphics, geometric modeling and image processing, ACM Symp. on Computational Geometry, Baltimore, MD (1985) 272—280.

[3] H. Samet, Computing perimeters of images represented by quadtrees, IEEE Trans. Pattern Analysis & Machine Intelligence PAMI-3 (6) (1981) 683–687.
[4] H. Samet, The quadtree and related hierarchical data structures, ACM Comput. Surveys 16 (2) (1984) 187–260.
[5] H. Yahia, Analyse des Structures de Données Arborescentes Représentant des Images, Thèse de Troisième Cycle, Univ. d'Orsay, December 1986.